Information Hidden in Gradients of Regression with Target Noise

Arash Jamshidi
University of Helsinki

Katsiaryna Haitsiukevich
University of Helsinki

Kai Puolamäki
University of Helsinki

Abstract

Second-order information—such as curvature or data covariance—is critical for optimisation, diagnostics, and robustness. However, in many modern settings, only the gradients are observable. We show that the gradients alone can reveal the Hessian, equalling the data covariance Σ for the linear regression. Our key insight is a simple variance calibration: injecting Gaussian noise so that the total target noise variance equals the batch size ensures that the empirical gradient covariance closely approximates the Hessian, even when evaluated far from the optimum. We provide non-asymptotic operator-norm guarantees under sub-Gaussian inputs. We also show that without such calibration, recovery can fail by an $\Omega(1)$ factor. The proposed method is practical (a "set target-noise variance to n" rule) and robust (variance $\mathcal{O}(n)$ suffices to recover Σ up to scale). Applications include preconditioning for faster optimisation, adversarial risk estimation, and gradient-only training, for example, in distributed systems. We support our theoretical results with experiments on synthetic and real data.

1 INTRODUCTION

Gradients are fundamental in modern machine learning. They drive optimisation algorithms such as stochastic gradient descent (SGD), yet their usefulness extends far beyond optimisation. Gradients also serve as statistical objects that reveal structural information about the loss and the underlying data distribution. For instance, Hessian–vector products can be approximated using gradients at nearby points (Pearlmutter (1994)), providing a practical way to estimate curvature, and gradient-based stopping rules have been proposed to improve generalisation (Jamshidi et al. (2025); Mahsereci et al. (2017); Forouzesh and Thiran (2021)). These examples illustrate that gradients encode rich information that can be extracted in addition to their role in optimisation.

In this work, we investigate whether gradients can reveal the Hessian of the loss function. We consider the setting where the learner does not observe raw data but only average gradients over batches of data. This situation naturally arises in distributed and federated learning, where sharing data is infeasible but gradients are communicated. Another use case is the efficient implementation of optimisation methods in machine learning, where the optimisation algorithm may have easy access to only gradient information. Formally, our focus is on linear regression,

$$y = x^\top w_0 + \varepsilon, \quad x = \varphi(z), \quad (x, y) \in \mathbb{R}^d \times \mathbb{R}, \quad (1)$$

such that z is the data point and φ is a possible non-linear transformation of z. We assume covariates $x \sim \mathcal{D}_x$ satisfying $\mathbb{E}[x] = 0$ and $\mathbb{E}[xx^\top] = \Sigma \preceq I$, and target noise $\varepsilon \sim \mathcal{N}(0, \widetilde{\sigma}^2)$. We assume access only to aggregate batch gradients of the squared loss, for any batch $B = \{(x_i, y_i)\}_{i=1}^n$. We observe k such independent batches. Our goal is to recover the Hessian Σ using only these gradients. Later, in the experiments, we show that the results hold for non-linear models.

Problem (Estimating Σ from gradients)**.** *Given $\epsilon > 0$, construct an estimator $\widehat{\Sigma}$ from batch gradients such that, with high probability,*

$$\|\widehat{\Sigma} - \Sigma\|_{\mathrm{op}} \leq \epsilon, \quad (2)$$

where $\|\cdot\|_{\mathrm{op}}$ denotes the operator norm.

This task is challenging because gradients are entangled with both the current parameter w and the target noise ε. To address this, we highlight two key insights:

Insight 1: Bias–Variance Calibration. For sufficiently large batch size n, if we add artificial noise

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

$\mathcal{N}(0, O(n))$ to the targets y, the gradient covariance reveals the structure of Σ:

$$\mathrm{Cov}(\nabla L^{(B)}(w)) = \mathcal{O}(1)\,\Sigma + \Delta(w), \tag{3}$$

such that the nuisance term $\|\Delta(w)\|_{\mathrm{op}} \leq \epsilon$ and with the leading term proportional to Σ.[1]

Insight 2: Noise does not corrupt mean gradients. Injecting $\mathcal{N}(0, O(n))$ noise into the targets does not significantly alter the mean gradient across batches. Specifically, by defining $\nabla L^*(w) := \Sigma(w - w_0)$ as the true (population-level) gradient, for $k = \Omega(d/\epsilon^2)$, the noisy and true gradients satisfy

$$\left\| \frac{1}{k} \sum_{j=1}^{k} \nabla L^{(j)}(w) - \nabla L^*(w) \right\|_2 \leq \epsilon \tag{4}$$

with high probability.

Thus, noise calibration *preserves the accuracy of the gradient signal* while *enabling accurate recovery of* Σ. These two insights are summarised in Fig. 1.

Algorithmic Idea. Guided by these insights, we inject Gaussian noise of variance $O(n)$ into the targets y and then use the empirical gradient covariance of the batches $S_g(w)$ as an estimator of Σ.

Our Contributions. This work makes four main contributions: **(1)** We formalize the problem of estimating the Hessian Σ from gradient information alone. The gradient covariance encodes Σ but is generally biased. We provide a non-asymptotic analysis showing that, by injecting Gaussian noise into the targets at variance $O(n)$ and using the empirical gradient covariance $S_g(w)$, one can consistently recover Σ for any sub-Gaussian distribution $\mathcal{D}_x$ under mild assumptions, with high probability. **(2)** We show that *noise injection is not only sufficient but also necessary*. Without calibration, the operator-norm distance between gradient covariances and Σ can remain $\Omega(1)$, highlighting that adding Gaussian noise at variance $O(n)$ with a sufficient batch size is essential for recovery. **(3)** We demonstrate practical implications in gradient-only settings (e.g., efficient implementation of optimisation algorithms with access only to the gradient information, occurring, for example, in distributed systems): (i) estimating adversarial risk when only gradient access is available, and (ii) using calibrated gradient covariance for preconditioning, thereby accelerating convergence. **(4)** We show the effectiveness of our method in estimating the Hessian, in both linear and non-linear (ReLU neural networks) models, using real-world and synthetic data. Our code is available at `https://github.com/edahelsinki/noisy_targets`.

[1] We use the usual definition of covariance of a random vector Y, $\mathrm{Cov}(Y) = \mathbb{E}[(Y - \mathbb{E}(Y))(Y - \mathbb{E}(Y))^\top]$.

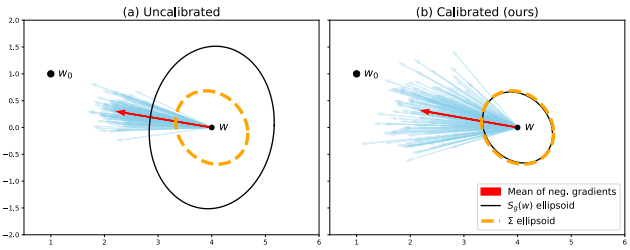

Figure 1: **Gradient covariance calibrated with target noise matches the Hessian (Insight 1) and preserves the mean gradient (Insight 2).** Blue thin lines: batch gradients; red thick line: their average; black solid ellipse: empirical covariance $S_g(w)$; orange dashed ellipse: data covariance Σ. *Left*: without calibration, $S_g(w)$ deviates from Σ. *Right*: after adding noise to the targets, the gradients spread out and $S_g(w)$ aligns closely with Σ, while the mean gradient remains accurate.

2 RELATED WORK

The addition of purposeful noise during model training can be considered from several angles.

Noise injection as a regulariser Noise injection is a common regularisation method, introduced via sampling, such as dropout (Srivastava et al., 2014) or SGD (Robbins and Monro, 1951; Bottou, 1991), or via additive perturbations (Dhifallah and Lu, 2021). It can be applied to model inputs (Dhifallah and Lu, 2021; Cohen et al., 2019), layer inputs (Orvieto et al., 2023), weights (Camuto et al., 2020; Wu et al., 2020a), or gradients (Zhu et al., 2018; Wen et al., 2020; Wu et al., 2020b), with gradient noise particularly effective for generalisation and large-batch training (Wu et al., 2020b; Zhu et al., 2018; Wen et al., 2020). Our approach—adding noise to targets—is closest to gradient perturbation, but introduces noise to the gradients implicitly and recovers the Hessian, enabling faster convergence when used as a preconditioner.

Hessian estimate by gradients using Fisher Information Second-order optimisation methods often use the Fisher Information matrix as a surrogate for the Hessian (Martens, 2020; Sen et al., 2024; Jhunjhunwala et al., 2024). Since it is difficult to compute directly, it is typically approximated by the empirical Fisher, i.e., the unnormalised uncentered gradient covariance (Rame et al., 2022). However, this approximation may not be reliable since its underlying assumptions rarely hold in practice (Kunstner et al., 2019). Additionally, the empirical Fisher information matrix uses point-wise gradients calculated from the model outputs, which can be computationally expen-

sive to obtain, especially if the size of the output is large. In contrast, our method uses the batch gradient of the loss, which is much more computationally efficient. We further demonstrate that the equality between the noise-free gradient covariance and the Hessian does not hold, even in linear regression.

Quasi-quadratic optimisation This class of optimisation methods estimates the second-order information from the gradients (Martens, 2010; Byrd et al., 2016; Goldfarb et al., 2020; Berahas et al., 2022). The main difference between our method and quasi-quadratic approaches is that quasi-quadratic methods focus on optimisation problems and building an efficient optimiser, while we use the statistical properties of the problem to extract information about the Hessian in general, with several possible applications, optimisation being one of them. Additionally, with our method, the weight updates can be performed with the mean of the noisy gradients to avoid calculating the forward and backward paths twice and more importantly, to reduce the information that is shared between nodes in case of distributed training.

Adversarial attacks and defences Adversarial attacks often involve adding carefully crafted noise to inputs to alter model predictions (Szegedy et al., 2013; Kurakin et al., 2018; Xu et al., 2023; Kong et al., 2025). In contrast, our noise injection is not learned and does not target prediction changes; instead, we preselect a suitable Gaussian noise level for targets to estimate the Hessian structure. On the other hand, federated learning trains models collaboratively by sharing parameter updates or gradients while keeping data local. Prior work shows that gradients can leak information about the data distribution, such as the target distribution (Wainakh et al., 2021; Kariyappa and Qureshi, 2023). A common defence is adding noise to gradients (Li et al., 2024; Wan et al., 2023). We show that carefully designed noise can also reveal the covariance structure of the inputs.

3 WARM-UP: EQUAL HESSIAN LOSSES

Before analysing the link between gradient covariance and the Hessian (or equivalently, the data covariance) in linear regression, we consider a simplified case. Let $L^{(i)}(w)$, $\nabla L^{(i)}(w)$, and $\nabla^2 L^{(i)}(w)$ denote the loss, gradient, and Hessian for batch i. Assume each $L^{(i)}$ is a quadratic function with shared Hessian $\nabla^2 L^{(i)} = \Sigma$ for all $i \in [k] = \{1, \dots, k\}$. In this setting, we have:

Example 3.1. *Let each batch $i \in [k]$ be associated with a general quadratic loss function of the form*

$$L^{(i)}(w) = a_i + b_i^\top w + w^\top \Sigma w/2, \tag{5}$$

where $a_i \in \mathbb{R}$, $b_i \in \mathbb{R}^d$, and $(a_i, b_i) \overset{i.i.d.}{\sim} \mathcal{D}$ for some unknown distribution $\mathcal{D}$. Suppose $\Sigma \in \mathbb{R}^{d\times d}$ is a fixed, symmetric, positive definite matrix. Define

$$w_*^{(i)} := \arg\min_{w\in\mathbb{R}^d} L^{(i)}(w), \quad \widehat{w} := \sum\nolimits_{i=1}^{k} w_*^{(i)}/k. \tag{6}$$

Then, for $w \in \mathbb{R}^d$, the gradient covariance is given by

$$S_g(w) = \Sigma \widehat{S}_{w_*} \Sigma, \tag{7}$$

where $\widehat{S}_{w_*} := \frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - \widehat{w})(w_*^{(i)} - \widehat{w})^\top$ is the empirical covariance of the optimal points $\{w_*^{(i)}\}_{i=1}^k$.

While Σ may differ between batches in practice (e.g., due to finite data effects), the central intuition in this chapter is that by controlling the distribution of $\widehat{S}_{w_*}$, we can alter the distribution of $S_g(w)$. In the case of linear regression, one way to achieve this is by manipulating the targets, for example, by adding noise.

4 GRADIENT COVARIANCE RECOVERS Σ WITH NOISE INJECTION

In this section, we show that the gradient covariance $S_g(w)$ approximates the Hessian Σ in linear regression when the target noise is $\mathcal{N}(0, n)$, with n the batch size, under both large- and small-batch regimes. Note that if the original targets already have target noise with known variance $\widetilde{\sigma}^2 < n$, it suffices to inject additional noise $\mathcal{N}(0, n - \widetilde{\sigma}^2)$ so that the total noise variance is n. We also discuss this case, as well as when $\widetilde{\sigma}^2$ is unknown, later, in Section 6.

In both regimes, we assume $n = \Omega(\|w - w_0\|_2^2)$ and $k = \Omega(d/\epsilon^2)$, which ensures the batch gradients are sub-exponential $SE_d(\mathcal{O}(1), \mathcal{O}(1))$ (see Appendix B.9). Consequently, the average of k batches concentrates around the population gradient:

$$\left\| \frac{1}{k}\sum\nolimits_{j=1}^{k} \nabla L^{(j)}(w) - \Sigma(w - w_0) \right\|_2 \lesssim \epsilon, \quad \epsilon < 1, \tag{8}$$

by standard concentration bounds (Wainwright, 2019). Thus, while recovering Σ, we also preserve accurate gradient estimation required for optimisation.

4.1 Large Batch Size

We prove our main result (gradient covariance estimates Hessian with target noise $\sigma^2 = n$) when the batch size is large.

Theorem 4.1. *Let $j \in [k]$ index a batch of size n, and suppose the inputs $x_i^{(j)} \overset{i.i.d.}{\sim} \mathcal{D}_x$ such that $\mathcal{D}_x \in SG_d(1)$*

(sub-Gaussian) such that $\mathbb{E}[x] = 0$ *and* $\mathbb{E}[xx^\top] = \Sigma \preceq I$*, the noise terms* $\varepsilon_i^{(j)} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2 = n)$*, and the targets are given by* $y_i^{(j)} = (x_i^{(j)})^\top w_0 + \varepsilon_i^{(j)}$*, for* $i \in [n]$*. Define the empirical least-squares loss over batch* j *as*

$$L^{(j)}(w) = \frac{1}{2n}\|X^{(j)}w - y^{(j)}\|_2^2, \tag{9}$$

and let $\nabla L^{(j)}(w)$ *denote the gradient of the loss with respect to* w*. Then, by defining the average gradient*

$$\nabla L := \frac{1}{k}\sum\nolimits_{\ell=1}^{k} \nabla L^{(\ell)}(w), \tag{10}$$

the empirical batch gradient covariance is given by

$$S_g(w) = \frac{1}{k}\sum_{j=1}^{k}\left(\nabla L^{(j)}(w) - \nabla L\right)\left(\nabla L^{(j)}(w) - \nabla L\right)^\top.$$

Then, by defining $c := \lambda_{min}(\Sigma)$ *as a constant, there exists dimension independent absolute constants* C_1, C_2, C_3*, and* C_4 *such that if batch size* n *and number of batches* k *satisfy the following:*

$$n \geq C_2 d^2(d + \log(k/\delta))/\epsilon^2 \tag{11}$$

$$d \geq C_4[\log(2k) + \log(1/\delta)] \tag{12}$$

$$k \geq C_1(d + \log(1/\delta))/\epsilon^2, \tag{13}$$

then, with probability at least $1 - 4\delta$*, we have*

$$\|S_g(w) - \Sigma\|_{\text{op}} \leq \epsilon \tag{14}$$

for all $\{w : ||w - w_0|| \leq C_3\sqrt{d}\}$*.*

The proof builds on Example 3.1, using the fact that for large batch size the Hessians of different batches are, with high probability, close in operator norm. The argument then follows from standard concentration and covering techniques.

We note two points about Theorem 4.1 and its proof:

(i) Although Theorem 4.1 is stated assuming equal batch size n, it is only for notational simplicity. The proof extends to unequal sizes $n_1, \ldots, n_k$, provided

$$\min(n_1, n_2, \ldots, n_k) \;\geq\; C_2 d^2(d + \log(k/\delta))/\epsilon^2. \tag{15}$$

(ii) A novel aspect of our analysis is that for sufficiently large batch size n, we guarantee

$$\|S_g(w) - \Sigma\|_{\text{op}} \lesssim \epsilon \tag{16}$$

uniformly for all w near w_0, using only $\Theta(d/\epsilon^2)$ batches—the sharp complexity already needed for concentration at a single point. Typically, uniform guarantees require far more batches; our improvement comes from exploiting correlations in large batches within a refined covering argument, avoiding extra logarithmic or polynomial factors in k.

4.2 Small Batch Size

Section 4.1 showed recovery of Σ from $S_g(w)$ with noisy targets when batches are large. We now argue it also holds for smaller batches, for any w, provided $k = \widetilde{\Omega}(d/\epsilon^2)$.[2] We begin with the following lemma:

Lemma 4.2. *Let* $j \in [k]$ *index a batch of size* n*, and suppose the inputs* $x_i^{(j)} \overset{i.i.d.}{\sim} \mathcal{D}_x$*, such that* $\mathcal{D}_x \in SG(1)$ *with* $\mathbb{E}[x] = 0$*,* $\Sigma = \mathbb{E}[xx^\top] \preceq I$*, the noise terms* $\varepsilon_i^{(j)} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$*, and the targets are given by* $y_i^{(j)} = (x_i^{(j)})^\top w_0 + \varepsilon_i^{(j)}$*, for* $i \in [n]$*. Define the empirical least-squares loss over batch* j *as* $L^{(j)}(w)$ *(as in Equation (9)) and let* $\nabla L^{(j)}(w)$ *denote the gradient of the loss with respect to* w*. Then we have:*

$$\text{Cov}(\nabla L^{(j)}(w)) = \frac{\sigma^2}{n}\Sigma + \frac{\text{noise}}{n}, \tag{17}$$

$$s.t.\ \|\text{noise}\|_{\text{op}} \leq C\|w - w_0\|_2^2, \tag{18}$$

for some absolute constant C*.*

From Lemma 4.2, for any fixed w, if $\sigma^2 = n$ and the batch size satisfies $n = \Omega(\|w - w_0\|_2^2/\epsilon)$, then

$$\|\text{Cov}(\nabla L^{(j)}(w)) - \Sigma\|_{\text{op}} \leq \epsilon, \tag{19}$$

as the operator norm of the nuisance term scales linearly with batch size n, and we proved $\|\text{noise}\|_{\text{op}} \leq C\|w - w_0\|_2^2$. So, the intuition behind our method is that by increasing the batch size and σ^2 at the same time, we preserve Σ while forcing the nuisance terms to become smaller. Moreover, because batch gradients are i.i.d. sub-exponential random vectors, $SE_d(\mathcal{O}(1), \mathcal{O}(1))$, using any covariance estimator $\widehat{S}_g(w)$ suitable for sub-exponential vectors (e.g., with truncation (Ke et al., 2019)) and $k = \widetilde{\Omega}(d/\epsilon^2)$ batches, we obtain

$$\|\widehat{S}_g(w) - \Sigma\|_{\text{op}} \lesssim \epsilon \tag{20}$$

with high probability. We do not elaborate further, as this is a direct application of the standard covariance estimation for sub-exponential vectors. Note that by using only these properties, a uniform convergence similar to Section 4.1 requires more batches, by the union bound.

5 RECOVERY OF Σ FAILS WITHOUT CALIBRATION

While the analysis in Section 4 demonstrates that when the target noise variance matches the batch size (i.e., $\sigma^2 = n$), the empirical gradient covariance $S_g(w)$

[2] See Appendix A.1 for the definition of $\widetilde{\Omega}$ and other notations.

closely approximates the population covariance Σ, it remains unclear from the non-asymptotic perspective whether adding artificial noise to the target variable y is actually necessary.

In this section, we answer this question affirmatively by analysing the exact (population-level) covariance of the gradients. We show that if the data points are i.i.d. draws from a multivariate Gaussian distribution, $x_i^{(j)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$, for all $i \in [n]$, $j \in [k]$, with $\Sigma \succ 0$, then the choice $\sigma^2 = \mathcal{O}(n)$ is necessary to recover Σ from $S_g(w)$.

Lemma 5.1. *Under the setting of Lemma 4.2, if we have $x_i^{(j)} \overset{i.i.d.}{\sim} \mathcal{N}(0, \Sigma)$, then we obtain:*

$$\begin{aligned}\operatorname{Cov}(\nabla L^{(j)}(w)) = (\sigma^2/n)\,\Sigma \\ + \Sigma(w - w_0)(w - w_0)^\top \Sigma/n \\ + ((w - w_0)^\top \Sigma (w - w_0))\Sigma/n. \quad (21)\end{aligned}$$

Hence, by Lemma 5.1, we show that recovery of Σ fails without noise injection:

Corollary 5.1.1. *Under the setting of Lemma 5.1, for any constants noise variance σ^2 and batch size n, there exists $w \in \mathbb{R}^d$ such that for any $j \in [k]$ we have*

$$\|\operatorname{Cov}(\nabla L^{(j)}(w)) - \Sigma\|_{\text{op}} / \|\Sigma\|_{\text{op}} \geq 0.99. \quad (22)$$

To further elaborate on the importance of Lemma 5.1, we show that the worst-case behaviour could not simply be derived using the general result of Lemma 4.2:

Lemma 5.2. *In dimension $d = 1$, there exists a (non-Gaussian) distribution $\mathcal{D}_x$ with $\mathbb{E}[x] = 0$ and $\mathbb{E}[x^2] = \Sigma = 1$, $w_0 \in \mathbb{R}$, such that for any $w \in \mathbb{R}$, batch size n, we have:*

$$\operatorname{Cov}(\nabla L^{(j)}(w)) = (\sigma^2/n)\,\Sigma. \quad (23)$$

Hence, as long as $\sigma^2 = n$, we have $\operatorname{Cov}(\nabla L^{(j)}(w)) = \Sigma$, for all $w \in \mathbb{R}^d$.

6 GRADIENT-ONLY APPLICATIONS

In many modern machine learning scenarios, a training algorithm receives only gradients while raw data remains hidden. Such situations also arise in federated or distributed learning (McMahan et al. (2017); Kairouz et al. (2021)), as well as in privacy-sensitive applications and bandwidth-limited systems. In such settings, the statistical properties of the gradients become the key source of information.

Our analysis enables estimation of the population covariance Σ from gradients alone, making it broadly applicable in gradient-only regimes. We highlight two applications: (i) designing preconditioners for faster convergence, since Σ coincides with the Hessian of the loss; and (ii) estimating adversarial risk from gradients, enabling robustness assessment without data access. Algorithm 1 summarises these ideas.

Algorithm 1 Optimisation with Noisy Gradients. At each step t, $\|w_t - w_0\|_\Sigma^2$ can be estimated by $\nabla L(w_t)^\top (S_g(w_t))^{-1} \nabla L(w_t)$ (Lemma 6.4) and $E^{\|\cdot\|}(w_t, w_0, \beta)$ by Lemma 6.3.

Require: Starting parameter $w_1 \in \mathbb{R}^d$, # epochs T, # batches k, batch size n, inherent target noise estimate $\widetilde{\sigma^2}$, target values $y^{(j)} \in \mathbb{R}^n$ and gradient function $\nabla L^{(j)}(w, y) \in \mathbb{R}^d$ for every batch $j \in [k]$, optimisation update OPTIMISER : $(w_t, g_t) \mapsto w_{t+1}$ where $w_t \in \mathbb{R}^d$ is the current parameter and $g_t \in \mathbb{R}^d$ is the preconditioned gradient.

$y^{(j)} \leftarrow y^{(j)} + \mathcal{N}\left(\overline{0}, (n - \widetilde{\sigma^2}) I_{n \times n}\right)$ for all $j \in [k]$
for $t = 1$ to T **do** ▷ Number of epochs
  $\mathbf{G}_t = \{\nabla L^{(j)}(w, y^{(j)})\}_{j \in [k]}$ ▷ Noisy gradients
  $\nabla L(w_t) \leftarrow \text{Mean}(\mathbf{G}_t)$ ▷ Gradient mean
  $S_g(w_t) \leftarrow \text{Cov}(\mathbf{G}_t)$ ▷ Gradient covariance
  $w_{t+1} \leftarrow \text{OPTIMISER}\left(w_t, (S_g(w_t))^{-1} \nabla L(w_t)\right)$
  ▷ See Sect. C.1 for efficient matrix inversion
end for

We assume targets follow $y = x^\top w_0 + \varepsilon$ such that $\varepsilon \sim \mathcal{N}(0, \widetilde{\sigma}^2)$ and $\widetilde{\sigma}^2 < n$. To align gradient covariance $S_g(w)$ with Σ, we add independent noise $\mathcal{N}(0,\, n - \widetilde{\sigma}^2)$, so the total variance is equal to n. In practice, $\widetilde{\sigma}^2$ may be unknown. If $\widetilde{\sigma}^2 < n$, adding noise $\mathcal{N}(0, n)$ yields

$$S_g(w) \approx c\Sigma, \quad c \in [1, 2], \quad (24)$$

recovering Σ up to scale (by Lemma 4.2), sufficing for most applications, where only the relative geometry of Σ matters. We also assumed $\mathbb{E}[x] = 0$ and no bias term. In practice, this can be corrected by centring x and y in each batch using their empirical means.

6.1 Preconditioning and Faster Convergence

In this section, we show that we can accelerate the convergence rate of gradient descent for the problem of linear regression by estimating the Hessian Σ using $S_g(w)$. The standard convergence result of gradient descent with constant step size is that to get error η we need $\approx \Omega(\kappa \log 1/\eta)$ steps (Nesterov (2013)), where $\kappa = \lambda_{\max}/\lambda_{\min}$, is the condition number. The above convergence rate is particularly slow when the condition number κ is large, but can be fixed using a good preconditioner, e.g., $(S_g(w))^{-1}$, such that $\|(S_g(w))^{-1} - \Sigma^{-1}\|_{\text{op}} \leq \epsilon$.

Theorem 6.1. *It is possible to significantly accelerate*

the convergence rate of gradient descent in linear regression, using noisy gradient covariance $S_g(w)$ if for each w such that $\|w - w_*\|_2 < \|w_1 - w_*\|_2$, we have $\|(S_g(w))^{-1} - \Sigma^{-1}\|_{\text{op}} \leq \epsilon$, where w_1 is the starting parameter. To achieve an error of η, one can reduce the number of required steps from

$$\Omega(\kappa \log(1/\eta)) \quad \textit{to} \quad \Omega\left(\frac{\log(1/\eta)}{\log(1/\epsilon)}\right) \tag{25}$$

where $\kappa = \lambda_{\max}/\lambda_{\min}$ is the condition number of Hessian (data covariance) $\Sigma \preceq I$. This demonstrates a method to achieve a condition-number-independent convergence rate.

6.2 Estimating Adversarial Risk

In this section, we show that the test-time adversarial risk in linear regression can be estimated from gradients alone, provided sufficient noise is added to the targets. This result also offers a principled way to perform early stopping using only gradient information (Scetbon and Dohmatob, 2023; Javanmard and Soltanolkotabi, 2022; Xing et al., 2021).

Lemma 6.2 (Informal)**.** *It is possible to estimate the adversarial risk $E^{\|\cdot\|}(w, w_0, \beta)$ using only noisy gradient information, for every w in a neighborhood of w_0.*

To formalise this, we begin with the following definition, adapted from Scetbon and Dohmatob (2023); Madry et al. (2017); Xing et al. (2021).

Definition 6.2.1 (Adversarial Risk)**.** *Let $w \in \mathbb{R}^d$ and $\beta \geq 0$. The adversarial risk of w at radius β is defined as*

$$E^{\|\cdot\|}(w, w_0, \beta) := \mathbb{E}_x \left[\sup_{\|\delta\|_2 \leq \beta} \left((x+\delta)^\top w - x^\top w_0\right)^2 \right],$$

where $x \sim \mathcal{N}(0, \Sigma)$ is a test-time input sampled from the data distribution. When $\beta = 0$, we call $E^{\|\cdot\|}(w, w_0, 0)$ the Ordinary Risk.

The intuition behind Definition 6.2.1 is that each training example x_i comes from $\mathcal{N}(0, \Sigma)$, and a test input x may be perturbed by a worst-case noise vector δ with $\|\delta\| \leq \beta$. This models the standard adversarial setting, where inputs are corrupted within a norm ball to maximise error. The resulting risk measure can also serve as a conservative criterion for early stopping.

In linear regression with Gaussian data, the adversarial risk admits a closed-form expression:

Lemma 6.3 ((Scetbon and Dohmatob, 2023) Closed–Form Expression for Adversarial Risk)**.** *Let $c_0 := \sqrt{2/\pi}$. Then, for any $w \in \mathbb{R}^d$ and $\beta \geq 0$, the adversarial risk is given by*

$$E^{\|\cdot\|}(w, w_0, \beta) = \|w - w_0\|_\Sigma^2 + \beta^2 \|w\|_2^2 + 2c_0\beta \|w - w_0\|_\Sigma \|w\|_2.$$

At first glance, Lemma 6.3 seems to require knowing both w_0 and Σ. We show that this dependence can be avoided: adversarial risk can be estimated using only w and noisy gradients.

Lemma 6.4 (Quadratic Form Approximation of $\|w - w_0\|_\Sigma^2$)**.** *Let $E^{\|\cdot\|}(w, w_0, 0) = \|w - w_0\|_\Sigma^2 = \mathcal{O}(1)$. Suppose we are given gradient and gradient covariance $\nabla L(w), S_g(w)$, such that*

$$\|\nabla L(w) - \Sigma(w - w_0)\|_2 \leq \mathcal{O}(\epsilon), \tag{26}$$

$$\|\left(S_g(w)\right)^{-1} - \Sigma^{-1}\|_{\text{op}} \leq \mathcal{O}(\epsilon) \tag{27}$$

for some $\epsilon < 1$. Then, for some quadratic function based only on gradients, we can provide an estimate of $\|w - w_0\|_\Sigma^2$ with additive error $\mathcal{O}(\epsilon)$; that is,

$$\left| \nabla L(w)^\top \left(S_g(w)\right)^{-1} \nabla L(w) - \|w - w_0\|_\Sigma^2 \right| \leq \mathcal{O}(\epsilon).$$

With an approximation of $\|w - w_0\|_\Sigma^2$, it is now easy to approximate $E^{\|\cdot\|}(w, w_0, \beta)$ (using Lemma 6.3), and hence establishing Lemma 6.2. Accurate estimation of $E^{\|\cdot\|}(w, w_0, \beta)$ enables principled early stopping of optimisation algorithms using only gradient information.

7 EXPERIMENTS

This section presents evaluations of the proposed method, along with its sensitivity analysis. We verify our theoretical results first on generated data from a Gaussian distribution and then extend the analysis to four real-world datasets. We test (a) linear regression models for the estimates of the data covariance (which is equal to the Hessian for linear models) and (b) multi-layer perceptrons (MLP) with ReLU activations for Hessian approximation. Both models are evaluated in the presence of added target noise and without it. We demonstrate the application of the Hessian estimate for adversarial risk calculation and as a preconditioner.

In this section, the quality of the Hessian approximations is evaluated by a relative operator norm r of the difference between $S_g(w)$ and Σ. The norm r is calculated as follows:

$$r = ||S_g(w) - \Sigma||_{\text{op}} / ||\Sigma||_{\text{op}} \tag{28}$$

7.1 Method analysis with generated data

We first evaluate the proposed target noise injection on data drawn from a Gaussian distribution with a dense

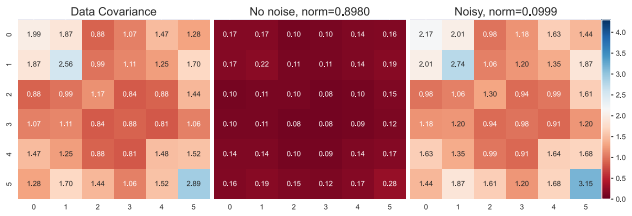

Figure 2: **Comparison of true data covariance and its estimates for the data generated from a Gaussian distribution.** Data covariance (*left*), its estimates by gradient covariance matrices with clean targets (*middle*), and with target noise of variance equal to batch size, $n = 256$ (*right*). We compare the results using the relative operator norm r (Eq. 28).

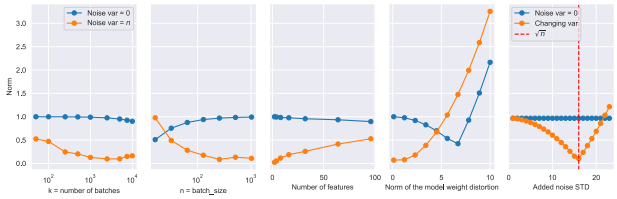

Figure 3: **Sensitivity of Hessian estimates.** Dependency of the estimates on (a) number of batches, (b) batch size, (c) number of features, (d) proximity of model weights to optimum, and (e) standard deviation of added target noise. We report norm r (Eq. 28). The estimate of the covariance matrix improves with a larger number of batches and larger batch sizes, and remains stable in the vicinity of the true model weights. The optimal added noise variance is equal to the batch size $n = 256$. Here, the model is randomly initialised except in plot (d), the optimal weights are distorted by a random vector with norm c.

covariance matrix. Fig. 2 compares the true data covariance with estimates from clean and noisy targets. While gradient covariance with clean targets partially captures the structure, it fails to reflect magnitude and complex dependencies. In contrast, adding noise to targets yields gradient covariances that closely match the true covariance (norm $r = 0.0999$). For this experiment, we used batch size $n = 256$, noise variance equal to n, and $k = 3125$ batches.

We further investigated the sensitivity of the data covariance estimates to the number of available batches, batch size, the number of features in the input, and the standard deviation of the added target noise. For this experiment, we generated 800,000 samples from a Gaussian distribution with a diagonal covariance matrix and a standard deviation of 2. Fig. 3a, b show that the estimates benefit from a larger number of batches k, given a sufficient batch size. We find that batch sizes above 64 work well; however, increasing the batch size n is beneficial. We set $n = 256$. As shown in Fig. 3c, the dataset with a bigger number of input features would require more data to maintain the same quality of the estimates. The covariance estimate is robust to distortions in model parameters (Fig. 3d). In this evaluation, the optimal values of model parameters are distorted by the addition of a random vector $\mathbf{v}$ with a norm equal to c, where c changes from 0.1 to 10. Our covariance matrix estimate becomes more accurate as the parameter values approach the optimum. Lastly, Fig. 3e shows that the optimal standard deviation (STD) for the added noise is equal to $\sqrt{n} = 16$. Further details can be found in Appendix D.1– D.3.

7.2 Linear regression model

We further tested the estimate of the Hessian using noise injection to the targets on real-world datasets with linear regression models. We demonstrate the robustness of the proposed Hessian estimate during model training. In the experiment, the model parameters (weights) were updated after each batch with the noise-free gradients (gradients obtained from clean targets) using the Adam optimiser (Kingma (2014)) with a batch size of 64.

In this experiment, we utilised the following four publicly available datasets: wave energy converters (Neshat et al. (2019)), bike sharing (Fanaee-T and Gama (2014)), California housing (Pace and Barry (1997)), and wine quality (Cortez et al. (2009)). To maintain consistency with theoretical sections, all input features were centred around zero by subtracting their mean. Similarly, we centred the targets; alternatively, a bias term can be included in the model to account for this. More details on data preprocessing and training can be found in Appendix D.4.

The characteristics of the tested datasets and the results by linear regression models are summarised in Table 1. We reported test MSE for the trained models and the relative operator norm r for three estimates of the Hessian: 1. the gradient covariance of clean gradients ('no noise'); 2. the estimate from 1. multiplied by the batch size n ('$\times n$'); 3. the proposed method for estimating the Hessian from the gradients with injected target noise with variance equal to n ('noise = n').

The presented results demonstrate the efficacy of the proposed approach compared to the alternatives. Simple rescaling of the noise-free gradient covariance by multiplication ('$\times n$') yields an informative estimate, as seen for the bike and wine datasets; however, the estimate using the noisy gradient is always superior. In

Table 1: **Hessian estimation for public datasets with clean and noisy gradient covariance matrices.** Mean (and standard deviation) of test MSE and the relative operator norm r are obtained with 10 random seeds.

DATA	FEAT.	SIZE	TEST MSE		NORM r, linear			NORM r, MLP	
			linear	MLP	no noise	$\times n$	noise $= n$	no noise	noise $= n$
Wave	48	288k	1.7e-10	6.9e-8	1.0 (0.0)	1.0 (6e-8)	**0.02** (0.01)	1.0 (4e-8)	**0.02 (0.01)**
Bike	50	17k	0.448	0.227	0.99 (4e-4)	0.65 (0.02)	**0.29** (0.05)	1.0 (2e-4)	**0.18 (0.04)**
Housing	8	20k	0.322	0.286	0.99 (9e-4)	2.05 (0.30)	**0.23** (0.24)	0.99 (0.01)	**0.51 (0.35)**
Wine	10	5k	0.474	0.421	0.99 (2e-3)	0.42 (0.13)	**0.17** (0.10)	0.99 (1e-3)	**0.22 (0.11)**

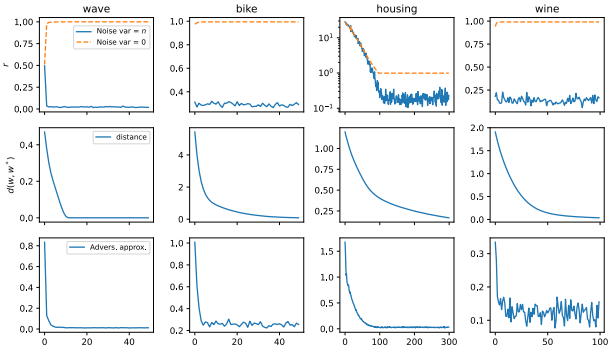

Figure 4: **Hessian estimate quality during model training.** *Top*: the relative operator norm r (Eq. 28 for the four studied datasets. *Middle*: the corresponding distance between the analytical solution and the weight on the current epoch. *Bottom*: Absolute error for the quadratic form approximation (Lemma 6.4) used in adversarial risk calculation (Lemma 6.3). The results correspond to the runs in Table 1 and are averaged across 10 random seeds.

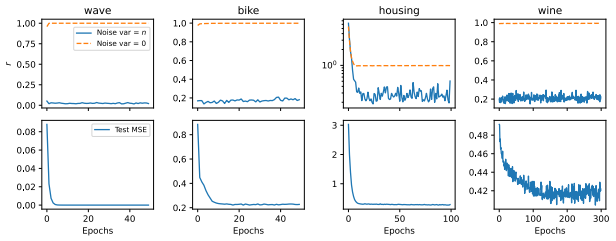

Figure 5: **Hessian estimate during MLP training with gradient covariance matrices.** *Top*: the relative operator norm for the four studied datasets with noise-free and noisy gradients. *Bottom*: Test MSE metric as the measure of convergence. The results correspond to the runs in Table 1 with an MLP.

the experiment, the model weights are updated after evaluating each batch, introducing a slight mismatch between the mean gradients used for the covariance calculation. However, the estimate is robust to it, and these effects can be neglected in practice.

Furthermore, Fig. 4 shows that the proposed Hessian estimate is valid in a sufficiently large neighbourhood of the optimal solution; before the optimisation procedure fully converged (see the evolution of the norm r during training, top row, and the corresponding distance between the current model weights and the analytical solution, middle row). This allows the utilisation of the obtained Hessian estimate as a preconditioner. Additionally, the estimated Hessian enables the quantification of adversarial risk by accurately approximating the quadratic form $\|w - w_0\|^2_\Sigma$ as derived in Lemma 6.4 (see Fig. 4, bottom row).

7.3 Extension to non-linear models

The results of the previous section can be extended to non-linear models for Hessian approximation. In the same experimental setup as in Section 7.2, we evaluated the Hessian estimates using a covariance matrix of clean and noisy gradients. Similar to the case of the linear model, the obtained Hessian estimates are stable in a neighbourhood of the optimal solution. They are more accurate compared to the covariance of noise-free gradients (see Table 1 and Fig. 5). For all datasets, we trained an MLP with 64 hidden neurons and an output layer to avoid overfitting. Further details are provided in Appendix D.4.

We also note that for an MLP with ReLU nonlinearities optimised with MSE loss, the blocks of the Hessian are target-independent (see Fig. 9 in Appendix D.5). Additionally, Appendix D.5.2 shows that noisy gradient covariance provides a reasonable estimate of the Hessian for MLPs of varying sizes on generated data.

7.4 Preconditioning

In this experiment, we utilised noisy gradients both for estimating the Hessian and for parameter updates during training. We also compared it to the case when clean gradients are used for updates. We refer to this setting as "oracle" since two sets of targets (clean and noisy) may not be available. We trained a linear model on datasets from Section 7.2 with gradient descent (GD) and Adam as optimisers as described in Algorithm 1. Fig. 6 shows that 1. the results for op-

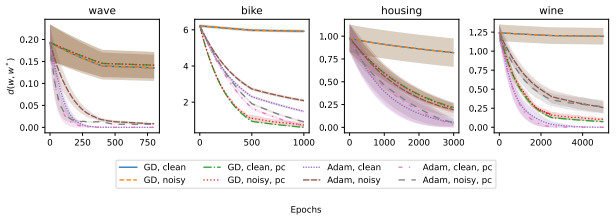

Figure 6: **Estimated Hessian as a preconditioner.** Optimisation with noisy mean gradient is close to the clean gradient (oracle). The inverse of the Hessian estimate as a covariance of noisy gradients is an effective preconditioner (pc) for both GD and Adam.

timisation with noisy mean gradients are close to the ones with a clean mean gradient (oracle) and 2. the estimated Hessian serves as an effective preconditioner for both optimisers, accelerating convergence in both clean and noisy gradient cases. More results are in Appendix D.6.

8 CONCLUSION AND FUTURE WORK

We showed that injecting Gaussian noise of variance $O(n)$ into the targets makes the empirical gradient covariance a reliable estimator of the Hessian Σ, with non-asymptotic operator-norm guarantees. The method preserves gradient accuracy and enables applications such as preconditioning, adversarial risk estimation, and gradient-only optimisation without access to raw data. We also showed extensions to non-linear models experimentally.

For future work, we highlight two directions. First, while our experiments show that target noise improves Hessian estimation in neural networks, its theoretical properties remain open. Second, while adding noise with large variance helps recover the Hessian, it also increases the variance of the gradients, thereby requiring more data as the noise level grows. This raises the question of whether there exists a way to perform post hoc variance reduction to largely undo the additional variance.

Acknowledgments

We thank the Research Council of Finland for funding, grants 364226 (Virtual Laboratory for Molecular Level Atmospheric Transformations (VILMA) Centre of Excellence) and 345704 (Finnish Center for Artificial Intelligence (FCAI)). We also thank the Helsinki Institute for Information Technology HIIT for funding. We thank the Finnish Computing Competence Infrastructure (FCCI) and CSC – IT Center for Science, Finland – for computational resources. The authors acknowledge the research environment provided by ELLIS Institute Finland.

References

Berahas, A. S., Jahani, M., Richtárik, P., and Takáč, M. (2022). Quasi-Newton methods for machine learning: forget the past, just sample. *Optimization Methods and Software*, 37(5):1668–1704.

Bottou, L. (1991). Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12.

Byrd, R. H., Hansen, S. L., Nocedal, J., and Singer, Y. (2016). A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031.

Camuto, A., Willetts, M., Simsekli, U., Roberts, S. J., and Holmes, C. C. (2020). Explicit regularisation in gaussian noise injections. *Advances in Neural Information Processing Systems*, 33:16603–16614.

Cohen, J., Rosenfeld, E., and Kolter, Z. (2019). Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4):547–553.

Dhifallah, O. and Lu, Y. (2021). On the inherent regularization effects of noise injection during training. In *International Conference on Machine Learning*, pages 2665–2675. PMLR.

Fanaee-T, H. and Gama, J. (2014). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2):113–127.

Forouzesh, M. and Thiran, P. (2021). Disparity between batches as a signal for early stopping. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer.

Goldfarb, D., Ren, Y., and Bahamou, A. (2020). Practical quasi-Newton methods for training deep neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 2386–2396.

Honorio, J. and Jaakkola, T. (2014). Tight bounds for the expected risk of linear classifiers and pac-bayes

finite-sample guarantees. In *International Conference on Artificial Intelligence and Statistics*, pages 384–392. PMLR.

Isserlis, L. (1918). On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, 12(1/2):134–139.

Jamshidi, A., Seppäläinen, L., Haitsiukevich, K., Luu, H. P. H., Björklund, A., and Puolamäki, K. (2025). GRADSTOP: Early stopping of gradient descent via posterior sampling. In *28th European Conference on Artificial Intelligence (ECAI)*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pages 2057–2064. IOS Press.

Javanmard, A. and Soltanolkotabi, M. (2022). Precise statistical analysis of classification accuracies for adversarial training. *The Annals of Statistics*, 50(4):2127–2156.

Jhunjhunwala, D., Wang, S., and Joshi, G. (2024). Fedfisher: Leveraging fisher information for one-shot federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1612–1620. PMLR.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210.

Kariyappa, S. and Qureshi, M. K. (2023). Exploit: Extracting private labels in split learning. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 165–175. IEEE.

Ke, Y., Minsker, S., Ren, Z., Sun, Q., and Zhou, W.-X. (2019). User-friendly covariance estimation for heavy-tailed distributions. *Statistical Science*, 34(3):454–471.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kong, W., Wang, X., Gao, R., Li, C., Zhang, Y., Yang, X., Wang, Y., and Tang, J. (2025). Adversarial semantic and label perturbation attack for pedestrian attribute recognition. *arXiv preprint arXiv:2505.23313*.

Kunstner, F., Hennig, P., and Balles, L. (2019). Limitations of the empirical fisher approximation for natural gradient descent. *Advances in Neural Information Processing Systems*, 32.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2018). Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*, pages 99–112. Chapman and Hall/CRC.

Li, Z., Chen, H., Gao, Y., Ni, Z., Xue, H., and Shao, H. (2024). Staged noise perturbation for privacy-preserving federated learning. *IEEE Transactions on Sustainable Computing*, 9(6):936–947.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mahsereci, M., Balles, L., Lassner, C., and Hennig, P. (2017). Early stopping without a validation set. *arXiv preprint arXiv:1703.09580*.

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

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

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

Neshat, M., Wagner, M., and Alexander, B. (2019). Wave energy converters. UCI Machine Learning Repository. DOI: `https://doi.org/10.24432/C5QS4V`.

Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Orvieto, A., Raj, A., Kersting, H., and Bach, F. (2023). Explicit regularization in overparametrized models via noise injection. In *International Conference on Artificial Intelligence and Statistics*, pages 7265–7287. PMLR.

Pace, R. K. and Barry, R. (1997). Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297.

Pearlmutter, B. A. (1994). Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160.

Rame, A., Dancette, C., and Cord, M. (2022). Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Scetbon, M. and Dohmatob, E. (2023). Robust linear regression: Gradient-descent, early-stopping, and beyond. In *International Conference on Artificial Intelligence and Statistics*, pages 11583–11607. PMLR.

Sen, M., Qin, A. K., Chen, Y.-W., Raman, B., et al. (2024). Sofim: Stochastic optimization using regularized fisher information matrix. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Speicher, R. (2023). High-dimensional analysis: Random matrices and machine learning. *Lecture Notes, Department of Mathematics, Saarland University*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Wainakh, A., Ventola, F., Müßig, T., Keim, J., Cordero, C. G., Zimmer, E., Grube, T., Kersting, K., and Mühlhäuser, M. (2021). User-level label leakage from gradients in federated learning. *arXiv preprint arXiv:2105.09369*.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press.

Wan, G., Du, H., Yuan, X., Yang, J., Chen, M., and Xu, J. (2023). Enhancing privacy preservation in federated learning via learning rate perturbation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4772–4781.

Wen, Y., Luk, K., Gazeau, M., Zhang, G., Chan, H., and Ba, J. (2020). An empirical study of stochastic gradient descent with structured covariance noise. In *International Conference on Artificial Intelligence and Statistics*, volume 108, pages 3621–3631. PMLR.

Wu, D., Xia, S.-T., and Wang, Y. (2020a). Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969.

Wu, J., Hu, W., Xiong, H., Huan, J., Braverman, V., and Zhu, Z. (2020b). On the noisy gradient descent that generalizes as SGD. In *International Conference on Machine Learning*, volume 119, pages 10367–10376. PMLR.

Xing, Y., Zhang, R., and Cheng, G. (2021). Adversarially robust estimate and risk analysis in linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 514–522. PMLR.

Xu, Y., Dai, P., Li, Z., Wang, H., and Cao, X. (2023). The best protection is attack: Fooling scene text recognition with minimal pixels. *IEEE Transactions on Information Forensics and Security*, 18:1580–1595.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. (2018). The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. *arXiv preprint arXiv:1803.00195*.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such case we will ask you to upload it during the author target period and include it in camera ready (if accepted).

In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**] **Throughout the text and in Appendix A.1**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**] **Throughout the text and in Appendix**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**] **Submitted as a zip supplement**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**]
   (b) Complete proofs of all theoretical results. [**Yes**] **Proofs are provided in the Appendix B.**
   (c) Clear explanations of any assumptions. [**Yes**] **In the corresponding sections.**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**] **The code with the corresponding instructions is uploaded as a zip supplement.**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**] **Details are provided in Appendix D.1 – D.4.**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**] **Section 7**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**] **We used an internal cluster with Xeon E5 2680, Xeon Gold 6148, and Xeon Gold 6248 CPUs (for the majority of the experiments) and NVIDIA V100 GPUs.**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**] **The datasets are cited in Section 7.**
   (b) The license information of the assets, if applicable. [**Yes**] **We used publicly available datasets in this work.**
   (c) New assets either in the supplemental material or as a URL, if applicable. [**Yes**] **The code is submitted as a zip supplement.**
   (d) Information about consent from data providers/curators. [**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Not Applicable**]

A PRELIMINARIES

A.1 Notation

We consider a linear regression model where the response is generated as $y = x^\top w_0 + \varepsilon$, with $x \in \mathbb{R}^d$ denoting the input vector, $w_0 \in \mathbb{R}^d$ the true parameter, and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ independent noise. We assume $\mathbb{E}[x] = 0$ and denote the population covariance by $\Sigma = \mathbb{E}[xx^\top]$, which is assumed to be positive definite. While our theoretical results hold under sub-Gaussian assumptions, we also consider the special case where $x \sim \mathcal{N}(0, \Sigma)$.

The dataset is partitioned into k batches, each containing n i.i.d. samples. For batch $j \in [k]$, the design matrix is $X^{(j)} \in \mathbb{R}^{n\times d}$ and the response vector is $y^{(j)} \in \mathbb{R}^n$. The empirical least-squares loss is $L^{(j)}(w) = \frac{1}{2n}\|X^{(j)}w - y^{(j)}\|^2$, with gradient $\nabla L^{(j)}(w)$, Hessian $\nabla^2 L^{(j)}(w)$, and batch minimizer $w_*^{(j)} = \arg\min_w L^{(j)}(w)$. The empirical gradient covariance is

$$S_g(w) = \frac{1}{k}\sum_{j=1}^{k}\left(\nabla L^{(j)}(w) - \frac{1}{k}\sum_{\ell=1}^{k}\nabla L^{(\ell)}(w)\right)\left(\nabla L^{(j)}(w) - \frac{1}{k}\sum_{\ell=1}^{k}\nabla L^{(\ell)}(w)\right)^\top.$$

The empirical covariance for batch j is $H^{(j)} = \frac{1}{n}(X^{(j)})^\top X^{(j)}$, and the population covariance is Σ.

For any vector $x \in \mathbb{R}^d$, we denote by $x_{[i]} \in \mathbb{R}$ the ith coordinate of x, for every $i \in [d]$. Similarly, for any matrix $A \in \mathbb{R}^{d\times d}$, we denote by $A_{[i,j]} \in \mathbb{R}$ the entry in the ith row and jth column of A, for every $i, j \in [d]$.

We use $\|\cdot\|_2$ for the Euclidean norm, $\|\cdot\|_{\mathrm{op}}$ for the operator norm (largest singular value), and $\|\cdot\|_F$ for the Frobenius norm. For symmetric matrices, $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the smallest and largest eigenvalues. We write $\mathbb{E}[\cdot]$ and $\mathbb{P}[\cdot]$ for expectation and probability, respectively.

We denote by $\mathrm{SG}(\sigma^2)$ the class of sub-Gaussian random variables with parameter σ^2, and by $\mathrm{SG}_d(\sigma^2)$ the class of sub-Gaussian random vectors in $\mathbb{R}^d$ with parameter σ^2. Similarly, we denote by $\mathrm{SE}(\nu^2, \alpha)$ the class of sub-exponential random variables with parameters (ν^2, α), and by $\mathrm{SE}_d(\nu^2, \alpha)$ the class of sub-exponential random vectors in $\mathbb{R}^d$ with parameters (ν^2, α). Constants such as $\epsilon \in (0, 1)$, $\delta \in (0, 1)$, and $c > 0$ appear in concentration inequalities and sample complexity results.

We use $\mathcal{O}(\cdot)$ and $\Omega(\cdot)$ to denote upper and lower bounds up to absolute constants. The notations $\widetilde{\mathcal{O}}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ are used similarly, but with logarithmic factors hidden. The symbol $\gtrsim$ denotes informal lower bounds up to absolute constants, i.e., $a \gtrsim b$ means $a \geq Cb$ for some constant $C > 0$; the reverse relation $a \lesssim b$ denotes $a \leq Cb$.

A.2 Useful Prior Results

Unless stated otherwise, most of the results in this section are taken from Vershynin (2018) and Wainwright (2019).

Definition A.0.1 (ε-net)**.** *Let (T, d) be a metric space. Consider a set $K \subset T$ and a number $\varepsilon > 0$. A subset $N_\epsilon \subset K$ is called an ε-net of K if every point in K is within distance ε of some point of N_ϵ, i.e.*

$$\forall x \in K\ \exists x_0 \in N_\epsilon : d(x, x_0) \leq \varepsilon. \tag{29}$$

Definition A.0.2 (Covering number)**.** *The smallest cardinality of an ε-net of K is called the covering number of K and is denoted $\mathcal{N}(K, \varepsilon)$.*

Definition A.0.3 (Sub-Gaussian random vector)**.** *A vector $\mathbf{x} \in \mathbb{R}^d$ with $\mathbb{E}[x] = \mu$ is vector sub-Gaussian ($\mathbf{x} \in SG_d(\sigma^2)$) with parameter σ if for all $v \in \mathbb{R}^d$ such that $||\mathbf{v}|| = 1$ we have*

$$\mathbb{E}\left[\exp\left(\mathbf{v}^\top(\mathbf{x} - \mu)\right)\right] \leq \exp\left(\lambda^2\sigma^2/2\right) \tag{30}$$

Definition A.0.4 (Sub-Exponential random variable)**.** *Centered random variable $X \in SE(\nu^2, \alpha)$ with parameters $\nu, \alpha > 0$ if:*

$$\mathbb{E}\left[\exp(\lambda X)\right] \leq \exp(\lambda^2\nu^2/2), \forall\lambda : |\lambda| < \frac{1}{\alpha}. \tag{31}$$

Definition A.0.5 (Sub-Exponential random vector). *A vector $\mathbf{x} \in \mathbb{R}^d$ with $\mathbb{E}[x] = \mu$ is vector sub-Exponential ($\mathbf{x} \in SE_d(\nu^2, \alpha)$) with parameter $\nu^2, \alpha > 0$ if for all $v \in \mathbb{R}^d$ such that $||\mathbf{v}|| = 1$ we have*

$$\mathbb{E}\left[\exp(\lambda(x-\mu)\right] \leq \exp(\lambda^2\nu^2/2), \forall\lambda : |\lambda| < \frac{1}{\alpha}. \tag{32}$$

Theorem A.1 (Sub-Gaussian Squared). *Honorio and Jaakkola (2014) For a sub-gaussian random variable $X \in SG(\sigma^2)$ with $\mathbb{E}[X] = 0$, $X^2 - \mathbb{E}[X^2]$ is sub-exponential, i.e., $X^2 - \mathbb{E}[X] \in SE(32\sigma^4, 4\sigma^2)$.*

Theorem A.2 (Tail bound for Sub-Exponential Random Variables). *Let $X \in SE(\nu^2, \alpha)$. Then:*

$$\mathbb{P}(|X-\mu| \geq t) \leq \begin{cases} 2e^{-t^2/(2\nu^2)}, & 0 < t \leq \frac{\nu^2}{\alpha} \quad \textit{(Sub-Gaussian behavior)} \\ 2e^{-t/(2\alpha)}, & t > \frac{\nu^2}{\alpha} \end{cases} \tag{33}$$

It can be equivalently stated as:

$$\mathbb{P}(|X-\mu| \geq t) \leq e^{-\frac{1}{2}\min\left\{\frac{t^2}{\nu^2}, \frac{t}{\alpha}\right\}} \tag{34}$$

Theorem A.3 (Composition property of Sub-Exponential random variables). *Let $X_1, \ldots, X_n$ be independent random variables such that $\mathbb{E}X_i = \mu_i$ and $X_i \in SE(\nu_i^2, \alpha_i)$. Then*

$$\sum_{i=1}^n (X_i - \mu_i) \in SE\left(\sum_{i=1}^n \nu_i^2, \max_i \alpha_i\right)$$

Lemma A.4 (Gaussian is sub-Gaussian). *If $X \sim \mathcal{N}(0, \Sigma)$, then X is a sub-Gaussian random vector with parameter $\|\Sigma\|_{op}$ ($X \in SG_d(\|\Sigma\|_{\text{op}})$.*

Theorem A.5 (Covering numbers of the Euclidean ball). *The covering numbers of the unit Euclidean ball B_2^d satisfy the following for any $\varepsilon > 0$:*

$$\left(\frac{1}{\varepsilon}\right)^d \leq \mathcal{N}(B_2^d, \varepsilon) \leq \left(\frac{2}{\varepsilon} + 1\right)^d. \tag{4.17}$$

The same upper bound is true for the unit Euclidean sphere S^{d-1}.

Theorem A.6 (Gaussian norm Concentration). *Speicher (2023) Consider a d-dimensional standard Gaussian random vector $x \sim \mathcal{N}(0, I_d)$. Then, for $0 \leq \varepsilon \leq \sqrt{d}$,*

$$\mathbb{P}\left\{\left|\|x\| - \sqrt{d}\right| \geq \varepsilon\right\} \leq 2\exp\left(-\frac{\varepsilon^2}{16}\right) \tag{35}$$

Theorem A.7 (Weyl's Theorem: Spectral Stability). *For any symmetric matrices $A, B \in \mathbb{R}^{d\times d}$ we have:*

$$\left|\lambda_i(B) - \lambda_i(A)\right| \leq \|A - B\|_{\text{op}}, \qquad i = 1, \ldots, d. \tag{36}$$

Lemma A.8 (Gaussian Sample Covariance Concentration). *Let $\mathbf{x}_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \Sigma)$ for $i = 1, \ldots, n$, where $\mathbf{x}_i \in \mathbb{R}^d$. Defining the empirical covariance matrix $\hat{\Sigma} \triangleq \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top$, as long as $n \geq \frac{20}{\epsilon^2}(d + \log(1/\delta))$, we have with probability at least $1 - \delta$,*

$$\frac{\|\hat{\Sigma} - \Sigma\|_{op}}{\|\Sigma\|_{op}} \leq \epsilon \tag{37}$$

Theorem A.9 (Sub-Gaussian Sample Covariance Concentration). *Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be an i.i.d. sequence of σ^2 sub-Gaussian random vectors such that $\text{Cov}[\mathbf{x}_1] = \Sigma$, and let $\hat{\Sigma} := \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top$ be the empirical covariance matrix. Then there exists a universal constant $C_1 > 0$ such that, for $\delta \in (0, 1)$, with probability at least $1 - \delta$,*

$$\frac{\|\hat{\Sigma} - \Sigma\|_{op}}{\sigma^2} \leq C_1 \max\left\{\sqrt{\frac{d + \log(2/\delta)}{n}}, \frac{d + \log(2/\delta)}{n}\right\}. \tag{38}$$

Therefore, for $\epsilon < 1$, there exists a universal constant $C > 0$, such that as long as we have $n \geq C\frac{\sigma^2}{\epsilon^2}(d + \log(1/\delta))$, then

$$\|\hat{\Sigma} - \Sigma\|_{op}/\sigma^2 \leq \epsilon. \tag{39}$$

Theorem A.10 (Isserlis's theorem)**.** *Isserlis (1918) Let $(X_1, \ldots, X_n)$ be a zero-mean multivariate normal random vector. Then for even n, the expectation of the product is given by*

$$\mathbb{E}[X_1 X_2 \cdots X_n] = \sum_{p \in P_n^2} \prod_{\{i,j\} \in p} \mathbb{E}[X_i X_j] = \sum_{p \in P_n^2} \prod_{\{i,j\} \in p} \mathrm{Cov}(X_i, X_j),$$

where the sum is over all pairings p of the set $\{1, \ldots, n\}$, i.e., all distinct ways of partitioning $\{1, \ldots, n\}$ into unordered pairs $\{i, j\}$, and the product is over the pairs in p.

B PROOFS

B.1 Proof of Example 3.1

Proof. Using Taylor's theorem, each loss function can be expressed as

$$L^{(i)}(w) = L^{(i)}(w_*^{(i)}) + \frac{1}{2}(w - w_*^{(i)})^\top \Sigma (w - w_*^{(i)}),$$

since $\nabla L^{(i)}(w_*^{(i)}) = 0$ and $\nabla^2 L^{(i)} = \Sigma$. Differentiating, we find

$$\nabla L^{(i)}(w) = \Sigma(w - w_*^{(i)}).$$

Averaging over i gives

$$\frac{1}{k}\sum_{i=1}^{k} \nabla L^{(i)}(w) = \Sigma\left(w - \widehat{w}\right).$$

The gradient covariance matrix is defined as

$$\begin{aligned}
S_g(w) &= \frac{1}{k}\sum_{i=1}^{k}\left(\nabla L^{(i)}(w) - \frac{1}{k}\sum_{j=1}^{k}\nabla L^{(j)}(w)\right)\left(\nabla L^{(i)}(w) - \frac{1}{k}\sum_{j=1}^{k}\nabla L^{(j)}(w)\right)^\top \\
&= \frac{1}{k}\sum_{i=1}^{k}\left(\Sigma(w - w_*^{(i)}) - \Sigma(w - \widehat{w})\right)\left(\Sigma(w - w_*^{(i)}) - \Sigma(w - \widehat{w})\right)^\top \\
&= \frac{1}{k}\sum_{i=1}^{k}\left(\Sigma(\widehat{w} - w_*^{(i)})\right)\left(\Sigma(\widehat{w} - w_*^{(i)})\right)^\top \\
&= \Sigma\left(\frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - \widehat{w})(w_*^{(i)} - \widehat{w})^\top\right)\Sigma \\
&= \Sigma \widehat{S}_{w_*} \Sigma.
\end{aligned}$$

□

B.2 Proof of Theorem 4.1

Proof. First, note that we have k batches, each containing n data points. We will derive the values of k and n later.

For each batch j we define $H^{(j)} = \frac{1}{n}\sum_{i=1}^{n} x_i^{(j)}(x_i^{(j)})^\top$. Note that based on Theorem A.9 for $n \geq \frac{Cd^2}{c^4\epsilon^2}(d + \log(k/\delta))$ (where C is an absolute constant), for every batch we have with probability $\geq 1 - \frac{\delta}{k}$ that $||H^{(j)} - \Sigma||_{\mathrm{op}} \leq \frac{c^2\epsilon}{d}$. Hence, by union bounding over the k batches, with probability at least $1 - \delta$ we have

$$||H^{(j)} - \Sigma||_{\mathrm{op}} \leq \frac{c^2\epsilon}{d} \quad \text{for all } j \in [k]. \tag{40}$$

We call this event $\mathcal{E}_1$ and through the rest of the proof, we are in the regime that $\mathcal{E}_1$ has occurred.

For simplicity we define $E^{(j)} := H^{(j)} - \Sigma$ and $\widehat{w} := \frac{1}{k}\sum_{i=1}^{k} w_*^{(i)}$. Now we write

$$S_g(w) = \frac{1}{k}\sum_{i=1}^{k}(\nabla L^{(i)}(w) - \frac{1}{k}\sum_{j=1}^{k}\nabla L^{(j)}(w))(\nabla L^{(i)}(w) - \frac{1}{k}\sum_{j=1}^{k}\nabla L^{(j)}(w))^\top \tag{41}$$

$$= \frac{1}{k}\sum_{i=1}^{k}\left[H^{(i)}(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}H^{(j)}(w - w_*^{(j)})\right] \tag{42}$$

$$\left[H^{(i)}(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}H^{(j)}(w - w_*^{(j)})\right]^\top \tag{43}$$

$$= \frac{1}{k}\sum_{i=1}^{k}\left[(\Sigma + E^{(i)})(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}(\Sigma + E^{(j)})(w - w_*^{(j)})\right] \tag{44}$$

$$\left[(\Sigma + E^{(i)})(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}(\Sigma + E^{(j)})(w - w_*^{(j)})\right]^\top \tag{45}$$

$$= \frac{1}{k}\sum_{i=1}^{k}\left[\Sigma(\widehat{w} - w_*^{(i)}) + E^{(i)}(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}E^{(j)}(w - w_*^{(j)})\right] \tag{46}$$

$$\left[\Sigma(\widehat{w} - w_*^{(i)}) + E^{(i)}(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}E^{(j)}(w - w_*^{(j)})\right]^\top \tag{47}$$

$$= \frac{1}{k}\sum_{i=1}^{k}\left[\Sigma(\widehat{w} - w_0 + w_0 - w_*^{(i)}) + E^{(i)}(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}E^{(j)}(w - w_*^{(j)})\right] \tag{48}$$

$$\left[\Sigma(\widehat{w} - w_0 + w_0 - w_*^{(i)}) + E^{(i)}(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}E^{(j)}(w - w_*^{(j)})\right]^\top \tag{49}$$

Hence, we can write:

$$S_g(w) = \frac{1}{k}\sum_{i=1}^{k}\left[\Sigma(\widehat{w} - w_0) + \Sigma(w_0 - w_*^{(i)}) + E^{(i)}(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}E^{(j)}(w - w_*^{(j)})\right] \tag{50}$$

$$\left[\Sigma(\widehat{w} - w_0) + \Sigma(w_0 - w_*^{(i)}) + E^{(i)}(w - w_*^{(i)}) - \frac{1}{k}\sum_{j=1}^{k}E^{(j)}(w - w_*^{(j)})\right]^\top \tag{51}$$

By expanding, we have 16 terms:

$$S_g(w) = \underbrace{\Sigma(\widehat{w} - w_0)(\widehat{w} - w_0)^\top\Sigma}_{A} + \underbrace{\frac{1}{k}\sum_{i=1}^{k}\Sigma(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top\Sigma}_{B} \tag{52}$$

$$+ \underbrace{\frac{1}{k}\sum_{i=1}^{k}\Sigma(\widehat{w} - w_0)(w_0 - w_*^{(i)})^\top\Sigma}_{C} + \underbrace{\frac{1}{k}\sum_{i=1}^{k}\Sigma(w_0 - w_*^{(i)})(\widehat{w} - w_0)^\top\Sigma}_{D} + ... \tag{53}$$

Before analysing the terms, we need some auxiliary lemmas.

Lemma B.1. *Under event $\mathcal{E}_1$, for all batches $j \in [k]$, we have*

$$||(H^{(j)})^{-1} - \Sigma^{-1}|| \leq \frac{2\epsilon}{d}. \tag{54}$$

Proof. Note that for all $j \in [k]$ we have $(H^{(j)})^{-1}$ is close to Σ. Indeed for any $j \in [k]$ we have:

$$(H^{(j)})^{-1} - \Sigma^{-1} = \Sigma^{-1}(\Sigma - H^{(j)})(H^{(j)})^{-1}, \tag{55}$$

hence we have

$$||(H^{(j)})^{-1} - \Sigma^{-1}||_{\text{op}} \leq ||\Sigma^{-1}||_{\text{op}}||\Sigma - H^{(j)}||_{\text{op}}||(H^{(j)})^{-1}||_{\text{op}} \tag{56}$$

$$\leq \frac{1}{c}\frac{c^2\epsilon}{d}||(H^{(j)})^{-1}||_{\text{op}}. \tag{57}$$

Now note that as $||H^{(j)} - \Sigma||_{\text{op}} \leq \frac{c^2\epsilon}{d}$, by Weyl's theorem (Theorem A.7) we have

$$|\lambda_d(H^{(j)}) - \lambda_d(\Sigma)| \leq \frac{c^2\epsilon}{d}. \tag{58}$$

Hence, we have $||(H^{(j)})^{-1}||_{\text{op}} \leq \frac{1}{\lambda_d(H^{(j)}) - (c^2\epsilon/d)} = \frac{1}{c - (c^2\epsilon/d)} \leq 2/c$. Then we can conclude:

$$||(H^{(j)})^{-1} - \Sigma^{-1}|| \leq \frac{1}{c}\frac{c^2\epsilon}{d}\frac{2}{c} = \frac{2\epsilon}{d}. \tag{59}$$

Lemma B.2. *Under event $\mathcal{E}_1$, with probability at least $1 - \delta$, event $\mathcal{E}_2$ as defined by*

$$||w_*^{(i)} - w_0||_2 \leq \sqrt{8d/c} \quad \textit{for all} \quad i \in [k]. \tag{60}$$

happens, if we have

$$d \geq 64[\log(2k) + \log(1/\delta)]. \tag{61}$$

Proof. We need to show that with high probability the distances between all $w_*^{(i)}$s and w_0 are bounded (i.e., they are on the order of $\approx \sqrt{d}$). First note that

$$w_*^{(i)} - w_0 \sim \mathcal{N}(0, (H^{(i)})^{-1}) \Leftrightarrow w_*^{(i)} - w_0 = (H^{(i)})^{-\frac{1}{2}} Z^{(i)} \quad \text{s.t.} \quad Z^{(i)} \sim \mathcal{N}(0, I). \tag{62}$$

Hence, we can write:

$$||w_*^{(i)} - w_0||_2 \leq ||(H^{(i)})^{-\frac{1}{2}}||_{\text{op}}||Z^{(i)}||_2 \leq \sqrt{2/c}||Z^{(i)}||_2. \tag{63}$$

By Theorem A.6 we have $\mathbb{P}\left\{\left|\|Z^{(i)}\| - \sqrt{d}\right| \geq \sqrt{d}/2\right\} \leq 2\exp\left(-\frac{d}{64}\right)$, hence by union bound over all batches, we have

$$\mathbb{P}\left\{\exists i \in [k] : \left|\|Z^{(i)}\| - \sqrt{d}\right| \geq \sqrt{d}/2\right\} \leq 2k\exp\left(-\frac{d}{64}\right). \tag{64}$$

Now, if we consider the failure probability for this event as $\delta > 0$, we have:

$$2k\exp\left(-\frac{d}{64}\right) \leq \delta \Leftrightarrow d \geq 64[\log(2k) + \log(1/\delta)] \tag{65}$$

So the final result is that, with probability at least $1-\delta$ we have

$$||w_*^{(i)} - w_0||_2 \leq \sqrt{2/c}\frac{3}{2}\sqrt{d} \leq \sqrt{8d/c} \text{ for all } i \in [k]. \tag{66}$$

Another Lemma that we need for the rest of the proof is that under the event that $\mathcal{E}_1$ is true, $||\widehat{w} - w_0||_2$ is small.

Lemma B.3. *Under event $\mathcal{E}_1$, if $k \geq d/\epsilon^2$, with probability at least $1-\delta$, event $\mathcal{E}_3$ as defined by*

$$||(\frac{1}{k}\sum_{j=1}^{k} w_*^{(i)}) - w_0||_2 \leq \frac{2\epsilon}{\sqrt{c}}. \tag{67}$$

happens.

Proof. Note that for all $i \in [k]$ we have $w_*^{(i)} \sim \mathcal{N}(w_0, (H^{(i)})^{-1})$. Hence because of independence of $w_*^{(i)}$s we have

$$(\frac{1}{k}\sum_{j=1}^{k} w_*^{(i)}) - w_0 \sim \mathcal{N}(0, \frac{1}{k^2}\sum_{i=1}^{k}(H^{(i)})^{-1}). \tag{68}$$

By the triangle inequality, we have

$$||\frac{1}{k^2}\sum_{i=1}^{k}(H^{(i)})^{-1}||_{\text{op}} \leq 2/ck. \tag{69}$$

Hence, using a similar argument as the previous lemma, by defining $Z \sim \mathcal{N}(0, I)$, we have

$$(\frac{1}{k}\sum_{j=1}^{k} w_*^{(i)}) - w_0 = (\frac{1}{k}\sum_{i=1}^{k}(H^{(i)})^{-1})^{1/2} Z \tag{70}$$

Now similar to proof of Lemma B.2 we can prove that for large enough d, with probability at least $1-\delta$ we have

$$||(\frac{1}{k}\sum_{j=1}^{k} w_*^{(i)}) - w_0||_2 \leq \sqrt{\frac{4d}{ck}}. \tag{71}$$

And because we will set $k \geq d/\epsilon^2$, we have

$$||(\frac{1}{k}\sum_{j=1}^{k} w_*^{(i)}) - w_0||_2 \leq \frac{2\epsilon}{\sqrt{c}}. \tag{72}$$

Lemma B.4. *Under event $\mathcal{E}_1$ and $\mathcal{E}_2$, for all $i \in [k]$ we have*

$$\|w_*^{(i)} - w_0\|_2 \leq 3\sqrt{d} \tag{73}$$

Proof. By triangle inequality it's easy to see that for all i we have $||w - w_*^{(i)}||_2 \leq 4\sqrt{d}$ as by assumption we have $||w - w_0||_2 \leq \sqrt{d}$ and we proved in Lemma B.2 that for all $i \in [k]$ we have $\|w_*^{(i)} - w_0\|_2 \leq 3\sqrt{d}$.

Proof of Theorem 4.1 (continued). We now continue the proof. We condition on happening of events $\mathcal{E}_1$, $\mathcal{E}_2$, and

$\mathcal{E}_3$.

We derive each term in order to prove $B \approx \Sigma$ and for all other terms X we have $||X||_{\text{op}} \lesssim \epsilon$:

Given the above Lemmas and using the fact that for any two matrices A, B we have $||AB||_{\text{op}} \leq ||A||_{\text{op}}||B||_{\text{op}}$ It is easy to see that for all terms X except B, we have $||X||_{\text{op}} \leq C\epsilon$ for some constant C. The only thing that we should calculate is the term B:

$$B = \frac{1}{k}\sum_{i=1}^{k} \Sigma(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top \Sigma = \Sigma\left(\frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top\right)\Sigma. \tag{74}$$

By adding and subtracting terms, we have

$$B = \Sigma\left(\frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top - \Sigma^{-1} + \Sigma^{-1}\right)\Sigma \tag{75}$$

$$= \Sigma + \Sigma\left[\frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top - \Sigma^{-1}\right]\Sigma \tag{76}$$

$$= \Sigma + \Sigma\underbrace{\left[\left(\frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top - (H^{(i)})^{-1}\right) + \left(\frac{1}{k}\sum_{i=1}^{k}(H^{(i)})^{-1} - \Sigma^{-1}\right)\right]}_{\text{We need to show this term has } ||\cdot||_{\text{op}} \lesssim \epsilon}\Sigma \tag{77}$$

It's easy to see $||\frac{1}{k}\sum_{i=1}^{k}(H^{(i)})^{-1} - \Sigma^{-1}||_{\text{op}} \lesssim \epsilon/d$ by triangle inequality, as each of $(H^{(i)})^{-1}$s are ϵ/d close to Σ^{-1}. Indeed, by using Lemma B.1 we can see that:

$$||\frac{1}{k}\sum_{i=1}^{k}(H^{(i)})^{-1} - \Sigma^{-1}||_{\text{op}} \leq \frac{1}{k}\sum_{i=1}^{k}||(H^{(i)})^{-1} - \Sigma^{-1}||_{\text{op}} \leq \frac{2\epsilon}{d} \tag{78}$$

So it's enough to show that with high probability $||\frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top - (H^{(i)})^{-1}||_{\text{op}} \lesssim \epsilon$.

Note that each $w_*^{(i)}$ is sampled from a distribution with mean w_0 and covariance matrix $(H^{(i)})^{-1}$. Based on Lemma B.1 we know all $(H^{(i)})^{-1}$s are close to Σ^{-1}, so we expect $\frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top$ to be close to Σ^{-1}. If the covariance matrices were in fact equal, we could use the standard covariance concentration bounds (Lemmas A.8 and A.9), but here, because they are not equal, we have to bound them some other way.

Before proceeding with the proof, we state a Lemma that we need for the rest of the proof. The proof of this lemma can be found in Vershynin (2018).

Lemma B.5. *(Computing the operator norm on a net Vershynin (2018)). Let A be an $d \times d$ symmetric matrix and $\varepsilon \in [0, 1)$. Then, for any ε-net N_ϵ of the sphere S^{n-1}, we have*

$$\|A\|_{\text{op}} \leq \frac{1}{1-2\varepsilon} \cdot \sup_{x \in N_\epsilon} |x^\top A x| \tag{79}$$

Proof of Theorem 4.1 (continued). To proceed with the proof, we use a covering argument. We consider $N_{1/4}$ a 1/4-net of sphere S^{n-1}. By setting $A = \frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top - (H^{(i)})^{-1}$ we have:

$$\|A\|_{\text{op}} \leq \max_{x \in N_{1/4}} 2|x^\top A x|, \text{ s.t. } A := \frac{1}{k}\sum_{i=1}^{k}(w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top - (H^{(i)})^{-1} \tag{80}$$

For any $x \in N_{1/4}$ we have:

$$x^\top A x = x^\top \left[\frac{1}{k} \sum_{i=1}^{k} (w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top - (H^{(i)})^{-1} \right] x \tag{81}$$

$$= \frac{1}{k} \sum_{i=1}^{k} (x^\top (w_*^{(i)} - w_0))^2 - x^\top (H^{(i)})^{-1} x, \tag{82}$$

such that for each $i \in [k]$ we have:

$$\mathbb{E}\left[(x^\top (w_*^{(i)} - w_0))^2 - x^\top (H^{(i)})^{-1} x \right] = x^\top \mathbb{E}\left[(w_*^{(i)} - w_0)(w_*^{(i)} - w_0) \right]^\top x - x^\top (H^{(i)})^{-1} x = 0. \tag{83}$$

We can see that by Lemma A.4 for each $i \in [k]$, we have $x^\top (w_*^{(i)} - w_0) \in SG_d(\|x\|_2^2 \|(H^{(i)})^{-1}\|_{\text{op}})$, and by Lemma B.1 and the fact that $\|x\|_2^2 \leq 1$ we can conclude $x^\top (w_*^{(i)} - w_0) \in SG(2/c)$ (e.g., $x^\top (w_*^{(i)} - w_0)$ is a sub-Gaussian vector with parameter $2/c$).

Now note that by Lemma A.1, for every $i \in [k]$, $(x^\top (w_*^{(i)} - w_0))^2 - x^\top (H^{(i)})^{-1} x \in SE(128/c^2, 4/c)$ and they are independent. By Lemma A.3 we have:

$$\frac{1}{k} \sum_{i=1}^{k} (x^\top (w_*^{(i)} - w_0))^2 - x^\top (H^{(i)})^{-1} x \in SE(128/c^2 k, 4/ck). \tag{84}$$

Note that by Theorem A.5, we have $N_{1/4} \leq 9^d$. We can then use the standard sub-exponential concentration bound (Lemma A.2) and union bound over all the $x \in N_{1/4}$:

$$\mathbb{P}\left[||\frac{1}{k} \sum_{i=1}^{k} (w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top - (H^{(i)})^{-1}||_{\text{op}} \geq \epsilon \right] \leq \tag{85}$$

$$\mathbb{P}\left[\max_{x \in N_{1/4}} | \, x^\top \left(\frac{1}{k} \sum_{i=1}^{k} (w_*^{(i)} - w_0)(w_*^{(i)} - w_0)^\top - (H^{(i)})^{-1} \right) x \, | \geq \epsilon/2 \right] \leq \tag{86}$$

$$9^d \mathbb{P}\left[SE(128/c^2 k, 4/ck) \geq \epsilon/2 \right] \leq \tag{87}$$

$$\exp\left(-\frac{1}{2} \min\left\{ \frac{c^2 k \epsilon^2}{512}, \frac{ck\epsilon}{8} \right\} + 3d \right) \leq \tag{88}$$

$$\exp\left(-\frac{c^2 k \epsilon^2}{1024} + 3d \right), \tag{89}$$

where the first inequality is by using the net argument, the second inequality is by union bounding and the fact that random variables are sub-exponential, the third inequality is by probability bound of sub-exponential random variables, and the last inequality is by simple algebraic manipulations. We want this event (which we call $\mathcal{E}_4$) to happen with probability at most δ. Hence, we have:

$$\exp\left(-\frac{c^2 k \epsilon^2}{1024} + 3d \right) \leq \delta \Leftrightarrow k \geq \frac{1024(\log(1/\delta) + 3d)}{c^2 \epsilon^2}. \tag{90}$$

Hence, ignoring absolute constants, it is enough to have a large number of batches k, i.e.,

$$k \gtrsim \frac{d + \log(1/\delta)}{\epsilon^2}. \tag{91}$$

For all of our results to fail, at least one of the events $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$, or $\mathcal{E}_4$ must fail. Recall that the validity of $\mathcal{E}_2$ and $\mathcal{E}_3$ depends on the success of $\mathcal{E}_1$, while the validity of $\mathcal{E}_4$ depends on the joint success of $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$. For any event $\mathcal{E}_i$, we denote its complement by $\bar{\mathcal{E}}_i$. Thus, because we bound a finite number of matrices to have operator norm $\leq C\epsilon$ for some constant C, for some absolute constant C' we obtain:

$$\mathbb{P}\left[\|S_g(w) - \Sigma\|_{\text{op}} > C'\epsilon \right] \leq \mathbb{P}\left[\bar{\mathcal{E}}_1 \vee \bar{\mathcal{E}}_2 \vee \bar{\mathcal{E}}_3 \vee \bar{\mathcal{E}}_4 \right] \tag{92}$$

$$\leq \mathbb{P}[\bar{\mathcal{E}}_1] + \mathbb{P}[\mathcal{E}_1]\mathbb{P}[\bar{\mathcal{E}}_2 \mid \mathcal{E}_1] + \mathbb{P}[\mathcal{E}_1]\mathbb{P}[\bar{\mathcal{E}}_3 \mid \mathcal{E}_1] + \mathbb{P}[\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3]\mathbb{P}[\bar{\mathcal{E}}_4 \mid \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3] \tag{93}$$

$$\leq \mathbb{P}[\bar{\mathcal{E}}_1] + \mathbb{P}[\bar{\mathcal{E}}_2 \mid \mathcal{E}_1] + \mathbb{P}[\bar{\mathcal{E}}_3 \mid \mathcal{E}_1] + \mathbb{P}[\bar{\mathcal{E}}_4 \mid \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3] \tag{94}$$

$$\leq 4\delta \tag{95}$$

□

B.3 Proof of Lemma 4.2

Proof. We compute $\mathrm{Cov}(\nabla L^{(j)}(w))$ for the j'th batch. As the distribution for all batches is equal, for simplicity of the notation, we remove the batch index j in this proof. We define $H = \frac{1}{n}X^\top X = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top$. First note that

$$\nabla L(w) = \frac{1}{n}X^\top(Xw - y) = \frac{1}{n}X^\top(Xw - Xw_0 - \varepsilon) \tag{96}$$

$$= \frac{1}{n}X^\top Xw - \frac{1}{n}X^\top Xw_0 - \frac{1}{n}X^\top\varepsilon = H(w - w_0) - \frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i \tag{97}$$

For simplicity we call $\frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i = \zeta$. Hence we have

$$\mathbb{E}[\nabla L(w)] = \mathbb{E}[H](w - w_0) - \sum_{i=1}^n \mathbb{E}[\varepsilon_i]\,\mathbb{E}[x_i] = \Sigma(w - w_0) \tag{98}$$

where the second expectation is zero because ε_i and x_i are independent. Then we have:

$$\mathrm{Cov}(\nabla L(w)) = \mathbb{E}[(\nabla L(w) - \mathbb{E}[\nabla L(w)])(\nabla L(w) - \mathbb{E}[\nabla L(w)])^\top] \tag{99}$$

$$= \mathbb{E}[((H - \Sigma)(w - w_0) - \zeta)((H - \Sigma)(w - w_0) - \zeta)^\top] \tag{100}$$

which is equal to

$$\mathrm{Cov}(\nabla L(w)) = \mathbb{E}[\zeta\zeta^\top - ((H - \Sigma)(w - w_0))\zeta^\top - \zeta((H - \Sigma)(w - w_0))^\top \tag{101}$$

$$+(H - \Sigma)(w - w_0)(w - w_0)^\top(H - \Sigma)] \tag{102}$$

There are four terms, and we derive them individually.

1- $\mathbb{E}[\zeta\zeta^\top]$:

$$\mathbb{E}[\zeta\zeta^\top] = \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n \mathbb{E}[\varepsilon_i\varepsilon_j x_i x_j^\top] \tag{103}$$

$$= \frac{1}{n^2}\sum_{i\neq j} \mathbb{E}\,\varepsilon_i\,\mathbb{E}\,\varepsilon_j\,\mathbb{E}\,x_i\,\mathbb{E}\,x_j^\top + \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}[\varepsilon_i\varepsilon_i]\,\mathbb{E}[x_i x_i^\top] \tag{104}$$

$$= 0 + \frac{1}{n^2}\sum_{i=1}^n \sigma^2\,\mathbb{E}[x_i x_i^\top] = \frac{\sigma^2}{n}\Sigma. \tag{105}$$

2- $\mathbb{E}[((H - \Sigma)(w - w_0))\zeta^\top]$:

$$\mathbb{E}[((H - \Sigma)(w - w_0))\zeta^\top] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[H(w - w_0)\varepsilon_i x_i^\top] \tag{106}$$

$$-\frac{1}{n}\sum_{i=1}^n \mathbb{E}[\Sigma(w - w_0)\varepsilon_i x_i^\top] \tag{107}$$

$$= \frac{1}{n}\sum_{i=1}^n \mathbb{E}_X[\mathbb{E}_\varepsilon[H(w - w_0)\varepsilon_i x_i^\top | X]] \tag{108}$$

$$-\frac{1}{n}\sum_{i=1}^n \mathbb{E}_X[\mathbb{E}_\varepsilon[\Sigma(w - w_0)\varepsilon_i x_i^\top | X]] = 0. \tag{109}$$

Both expectation terms are zero because the inner expectations are zero.

3- $\mathbb{E}[\zeta((H-\Sigma)(w-w_0))^\top]$: Similarly to case 2, we can write

$$\mathbb{E}[\zeta((H-\Sigma)(w-w_0))^\top] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\varepsilon_i x_i (w-w_0)^\top H] \tag{110}$$

$$-\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\varepsilon_i x_i (w-w_0)^\top \Sigma] \tag{111}$$

$$=\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_X[\mathbb{E}_\varepsilon[\varepsilon_i x_i (w-w_0)^\top H|X]] \tag{112}$$

$$-\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_X[\mathbb{E}_\varepsilon[\varepsilon_i x_i (w-w_0)^\top \Sigma|X]] = 0. \tag{113}$$

4- $\mathbb{E}[(H-\Sigma)(w-w_0)(w-w_0)^\top(H-\Sigma)]$: for simplicity we call $u = w - w_0$. We use u instead of $w - w_0$ for clear notation. We have:

$$\mathbb{E}[(H-\Sigma)uu^\top(H-\Sigma)] = \mathbb{E}[\frac{1}{n}(\sum_{i=1}^{n} x_i x_i^\top - \Sigma)uu^\top(\sum_{i=1}^{n} x_i x_i^\top - \Sigma)\frac{1}{n}] \tag{114}$$

$$=\frac{1}{n^2}\sum_{i\neq j}(\mathbb{E}(x_i x_i^\top) - \Sigma)uu^\top(\mathbb{E}(x_j x_j^\top) - \Sigma) \tag{115}$$

$$+\frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}[(x_i x_i^\top - \Sigma)uu^\top(x_i x_i^\top - \Sigma)] \tag{116}$$

$$=\frac{1}{n}\mathbb{E}[(x_1 x_1^\top - \Sigma)uu^\top(x_1 x_1^\top - \Sigma)]. \tag{117}$$

We drop the subscript 1 from now on in this proof and will use a subscript for the index. We denote the j'th index of x with $x_{[j]} \in \mathbb{R}$ in the rest of the proof. We continue to derive $\mathbb{E}[(xx^\top - \Sigma)uu^\top(xx^\top - \Sigma)]$.

$$\mathbb{E}[(xx^\top - \Sigma)uu^\top(xx^\top - \Sigma)] = \Sigma uu^\top\Sigma - \mathbb{E}[xx^\top]uu^\top\Sigma - \Sigma uu^\top\mathbb{E}[xx^\top] + \mathbb{E}[xx^\top uu^\top xx^\top] \tag{118}$$

$$= \mathbb{E}[xx^\top uu^\top xx^\top] - \Sigma uu^\top\Sigma. \tag{119}$$

Note that because we assumed $\Sigma \preceq I$ we have:

$$\|\Sigma uu^\top\Sigma\|_{\text{op}} \leq \|\Sigma\|_{\text{op}}^2\|u\|_2^2 = \|\Sigma\|_{\text{op}}^2\|w-w_0\|_2^2 \leq \|w-w_0\|_2^2. \tag{120}$$

Now we proceed to bound the operator norm of $\mathbb{E}[xx^\top uu^\top xx^\top] = \mathbb{E}[(x^\top u)^2 xx^\top]$. Note that this matrix is symmetric and positive semi-definite. Now observe that for any $v \in \mathbb{R}^d$ such that $\|v\|_2 = 1$ we have:

$$v^\top\mathbb{E}[(x^\top u)^2 xx^\top]v = \mathbb{E}[(x^\top u)^2 v^\top xx^\top v] = \|u\|_2^2\mathbb{E}[(x^\top u/\|u\|_2)^2(x^\top v)^2] \leq \|u\|_2^2\sqrt{\mathbb{E}[(x^\top u/\|u\|_2)^4]}\sqrt{\mathbb{E}[(x^\top v)^4]}, \tag{121}$$

where the inequality is by Cauchy-Schwarz. Now note that because $x \in \text{SG}_d(1)$ by our assumption, both $x^\top u/\|u\|_2$ and $x^\top v$ are SG(1) by definition. Hence by using the bound on m-th moment of sub-Gaussian random variables with $m = 4$ (Vershynin (2018)), we have:

$$\|u\|_2^2\sqrt{\mathbb{E}[(x^\top u/\|u\|_2)^4]}\sqrt{\mathbb{E}[(x^\top v)^4]} \leq C'\|u\|_2^2 = C'\|w-w_0\|_2^2 \tag{122}$$

for some absolute constant C' and by using the definition of $u = w - w_0$. Hence, by the triangle inequality, we have:

$$\|\,\mathbb{E}[xx^\top uu^\top xx^\top] - \Sigma uu^\top\Sigma\|_{\text{op}} \leq \|\,\mathbb{E}[xx^\top uu^\top xx^\top]\|_{\text{op}} + \|\Sigma uu^\top\Sigma\|_{\text{op}} \leq (C'+1)\|w-w_0\|_2^2 = C\|w-w_0\|_2^2, \tag{123}$$

for $C = C' + 1$. Now by setting noise $= \mathbb{E}[xx^\top uu^\top xx^\top] - \Sigma uu^\top \Sigma$ we have:

$$\mathrm{Cov}(\nabla L^{(j)}(w)) = \frac{\sigma^2}{n}\Sigma + \frac{\text{noise}}{n}, \tag{124}$$

such that

$$\|\text{noise}\|_{\text{op}} \le C\|w - w_0\|_2^2, \tag{125}$$

for some absolute constant C. □

B.4 Proof of Lemma 5.1

Proof. The first part of the proof is identical to the first part of the proof of Lemma 4.2, in which we have:

$$\mathrm{Cov}(\nabla L^{(j)}(w)) = \frac{\sigma^2}{n}\,\Sigma \;+\; \frac{1}{n}\left[\mathbb{E}[xx^\top (w-w_0)(w-w_0)^\top xx^\top] - \Sigma(w-w_0)(w-w_0)^\top\Sigma\right] \tag{126}$$

The only difference is that in this case $x \sim \mathcal{N}(0, \Sigma)$, hence we can derive $\mathbb{E}[xx^\top (w-w_0)(w-w_0)^\top xx^\top] = \mathbb{E}[(x^\top (w-w_0))^2 xx^\top]$ explicitly. For simplicity, as before, we define $u := w - w_0$.

We call the matrix $\mathbb{E}[(x^\top u)^2 xx^\top] := A$. Now we proceed to compute the value of $A_{[i,j]}$ for $i, j \in [d]$.

$$A_{[i,j]} = \mathbb{E}[(\sum_{k=1}^{d} x_{[k]}u_{[k]})^2 x_{[i]}x_{[j]}] = \sum_{k=1}^{d}\sum_{p=1}^{d} u_{[k]}u_{[p]}\,\mathbb{E}[x_{[k]}x_{[p]}x_{[i]}x_{[j]}]. \tag{127}$$

To derive the fourth moment, we use Isserlis's theorem (Theorem A.10). Note that $x_{[k]}x_{[p]}x_{[i]}x_{[j]}$ are jointly Gaussian, even if some of the indices get repeated.

$$\begin{aligned}\mathbb{E}[x_{[k]}x_{[p]}x_{[i]}x_{[j]}] &= \mathbb{E}[x_{[k]}x_{[p]}]\,\mathbb{E}[x_{[i]}x_{[j]}] + \mathbb{E}[x_{[k]}x_{[i]}]\,\mathbb{E}[x_{[p]}x_{[j]}] + \mathbb{E}[x_{[k]}x_{[j]}]\,\mathbb{E}[x_{[p]}x_{[i]}] &(128)\\ &= \Sigma_{[k,p]}\Sigma_{[i,j]} + \Sigma_{[k,i]}\Sigma_{[p,j]} + \Sigma_{[k,j]}\Sigma_{[p,i]}. &(129)\end{aligned}$$

Hence, we have:

$$\begin{aligned}A_{[i,j]} &= \sum_{k=1}^{d}\sum_{p=1}^{d} u_{[k]}u_{[p]}(\Sigma_{[k,p]}\Sigma_{[i,j]} + \Sigma_{[k,i]}\Sigma_{[p,j]} + \Sigma_{[k,j]}\Sigma_{[p,i]}) &(130)\\ &= \sum_{k=1}^{d}\sum_{p=1}^{d} u_{[k]}u_{[p]}\Sigma_{[k,p]}\Sigma_{[i,j]} + \sum_{k=1}^{d}\sum_{p=1}^{d} u_{[k]}u_{[p]}\Sigma_{[k,i]}\Sigma_{[p,j]} + \sum_{k=1}^{d}\sum_{p=1}^{d} u_{[k]}u_{[p]}\Sigma_{[k,j]}\Sigma_{[p,i]} &(131)\\ &= (u^\top\Sigma u)\Sigma_{[i,j]} + 2(\sum_{k=1}^{d}\Sigma_{[i,k]}u_{[k]})(\sum_{k=1}^{d}\Sigma_{[j,k]}u_{[k]}). &(132)\end{aligned}$$

where in the last line we used the symmetric property of Σ Now note that $(\sum_{k=1}^{d}\Sigma_{[i,k]}u_{[k]})(\sum_{k=1}^{d}\Sigma_{[j,k]}u_{[k]}) = (\Sigma uu^\top\Sigma)_{[i,j]}$ and $(u^\top\Sigma u)\Sigma_{[i,j]} = ((u^\top\Sigma u)\Sigma)_{[i,j]}$.

By combining all parts and the fact that $u = w - w_0$ we have:

$$\mathrm{Cov}(\nabla L^{(j)}(w)) = \frac{\sigma^2}{n}\,\Sigma \;+\; \frac{1}{n}\left[\Sigma\,(w-w_0)(w-w_0)^\intercal\Sigma \;+\; \big((w-w_0)^\intercal\Sigma(w-w_0)\big)\Sigma\right]. \tag{133}$$

□

B.5 Proof of Corollary 5.1.1

Proof. For any $w \in \mathbb{R}^d$ we have:

$$\begin{aligned}\|\mathrm{Cov}(\nabla L^{(j)}(w)) - \Sigma\|_{\text{op}} &= \|\frac{\sigma^2}{n}\,\Sigma \;+\; \frac{1}{n}\left[\Sigma\,(w-w_0)(w-w_0)^\intercal\Sigma \;+\; \big((w-w_0)^\intercal\Sigma(w-w_0)\big)\Sigma\right] - \Sigma\|_{\text{op}} &(134)\\ &= \|(\frac{\sigma^2}{n} - 1 + \frac{1}{n}(w-w_0)^\top\Sigma(w-w_0))\Sigma + \frac{1}{n}\,\Sigma\,(w-w_0)(w-w_0)^\intercal\Sigma\|_{\text{op}}. &(135)\end{aligned}$$

Now note that as Σ is positive definite, for any $C > 0$ we can find at least one w (we call it w_C) far from w_0 such that it ensures

$$\frac{\sigma^2}{n} - 1 + \frac{1}{n}(w_C - w_0)^\top \Sigma (w_C - w_0) > C. \tag{136}$$

Now note that we can see that for any $w \in \mathbb{R}^d, v \in \mathbb{R}^d : v \neq 0$ we have

$$v^\top \Sigma\,(w - w_0)(w - w_0)^\intercal \Sigma v = (v^\top \Sigma\,(w - w_0))^2 \geq 0, \tag{137}$$

meaning $\Sigma\,(w - w_0)(w - w_0)^\intercal \Sigma$ is positive semidefinite. Hence for any $C > 0$ we have:

$$\begin{aligned}
\|\mathrm{Cov}(\nabla L^{(j)}(w_C)) - \Sigma\|_{\mathrm{op}} &= \|(\frac{\sigma^2}{n} - 1 + \frac{1}{n}(w_C - w_0)^\top \Sigma (w_C - w_0))\Sigma + \frac{1}{n}\Sigma\,(w_C - w_0)(w_C - w_0)^\intercal \Sigma\|_{\mathrm{op}} && (138)\\
&\geq \|(\frac{\sigma^2}{n} - 1 + \frac{1}{n}(w_C - w_0)^\top \Sigma (w_C - w_0))\Sigma\|_{\mathrm{op}} && (139)\\
&\geq C\|\Sigma\|_{\mathrm{op}}. && (140)
\end{aligned}$$

Letting $C \to \infty$ completes the proof. □

B.6 Proof of Lemma 5.2

Proof. Suppose $w_0 = 1, n = 1, \sigma^2 = 1$ and suppose $\mathcal{D}_x$ is the Rademacher random variable, i.e., $\mathbb{P}(x = 1) = \mathbb{P}(x = -1) = 1/2$. Hence based on the proof of Lemma 4.2 we have:

$$\mathrm{Cov}(\nabla L^{(j)}(w)) = \Sigma + (w - 1)^2 \mathbb{E}[x^4] - (w - 1)^2 \Sigma^2. \tag{141}$$

Now note that the variance of the Rademacher random variable is 1, so we have:

$$\mathrm{Cov}(\nabla L^{(j)}(w)) = \Sigma + (w - 1)^2[\mathbb{E}[x^4] - 1] = \Sigma + (w - 1)^2[1 - 1] = \Sigma, \tag{142}$$

where in the last step we used the simple fact that $\mathbb{E}[x^4] = \mathbb{E}[1] = 1$. □

B.7 Proof of Theorem 6.1

Proof. Before proving the result note that using a similar argument as Lemma A.7, we can infer closeness of $(S_g(w))^{-1}$ and Σ^{-1} by closeness of $S_g(w)$ and Σ, given sufficient regularity conditions ($\lambda_d(\Sigma)$ being a constant and $\epsilon < \lambda_d(\Sigma)/2$). Hence, $(S_g(w))^{-1}$ and Σ^{-1} being close is a natural assumption given the results of Section 4.

Suppose w_* is the empirical optimal parameter. For simplicity, we use gradient descent with a step size of 1. Note that:

$$\begin{aligned}
w_{t+1} - w_* &= w_t - (S_g(w_t))^{-1}\Sigma(w_t - w_*) - w_* && (143)\\
&= w_t - (\Sigma^{-1} + (S_g(w_t))^{-1} - \Sigma^{-1})\Sigma(w_t - w_*) - w_* && (144)\\
&= ((S_g(w_t))^{-1} - \Sigma^{-1})\Sigma(w_t - w_*). && (145)
\end{aligned}$$

Hence, we have

$$\|w_{t+1} - w_*\|_2 = \|((S_g(w_t))^{-1} - \Sigma^{-1})\Sigma(w_t - w_*)\|_2 \leq \epsilon\|w_t - w_*\|_2. \tag{146}$$

Where the inequality is by simply using $\|AB\|_{\mathrm{op}} \leq \|A\|_{\mathrm{op}}\|B\|_{\mathrm{op}}$ and the fact that $\Sigma \preceq I$. Then, by induction, we have

$$\|w_{t+1} - w_*\|_2 \leq \epsilon^t\|w_1 - w_*\|_2, \tag{147}$$

and the convergence rate will follow from this. □

B.8 Proof of Lemma 6.4

Proof. The proof relies on the fact that

$$\|w - w_0\|_\Sigma^2 = (w - w_0)^\top \Sigma \Sigma^{-1} \Sigma (w - w_0) \tag{148}$$

$$= (\Sigma(w - w_0))^\top \Sigma^{-1} (\Sigma(w - w_0)) \tag{149}$$

The rest of the proof follows using the approximations. □

B.9 Sub-Exponential Property of Batch Gradients

Note that, as before, for any batch $j \in [k]$ we have:

$$\nabla L^{(j)}(w) = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top (w - w_0) - \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i x_i. \tag{150}$$

$$= \frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_i x_i^\top (w - w_0)/\sqrt{n} - \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \varepsilon_i x_i/\sqrt{n}. \tag{151}$$

where x_i is the ith data point of batch j and $\varepsilon_i \sim \mathcal{N}(0, n)$. Now for any $v \in \mathbb{R}^d$ such that $\|v\|_2 = 1$ we have

$$v^\top \nabla L^{(j)}(w) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (v^\top x_i)(x_i^\top (w - w_0)/\sqrt{n}) - \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (\varepsilon_i/\sqrt{n})(v^\top x_i). \tag{152}$$

Now note that in both large batch size and small batch size regimes, we have $n = \Omega(\|w - w_0\|_2^2)$. Hence we have that $\sqrt{n} = \Omega(\|w - w_0\|_2)$ and by the fact that $x_i \sim \mathrm{SG}_d(1)$, each of the $(\cdot)$ terms are $\mathrm{SG}(1)$, hence by usual properties of sub-Gaussian and sub-exponential random variables Vershynin (2018); Wainwright (2019) (e.g., product of two dependent sub-Gaussian is sub-exponential, sum of independent sub-exponentials is sub-exponential, and sum of dependent sub-exponentials is sub-exponential) we can see that the whole expression will be $\mathrm{SE}(\mathcal{O}(1), \mathcal{O}(1))$.

C PRACTICAL CONSIDERATIONS

C.1 Gradient Covariance Inversion

As discussed in the paper, many applications, such as preconditioning, require computing the action of the inverse empirical gradient covariance on a vector, $(S_g(w))^{-1}v$. This can be done efficiently by solving the linear system

$$S_g(w)u = v, \tag{153}$$

via a least-squares method, rather than explicitly forming the inverse matrix.

C.2 Scaling

While we assumed $x \in \mathrm{SG}_d(1)$, which implies $\Sigma \preceq I$ by standard properties of sub-Gaussian vectors, all arguments extend to the more general case $x \in \mathrm{SG}_d(\alpha)$, yielding $\Sigma \preceq \alpha I$ for some $\alpha > 1$. For instance, in Lemma 4.2 the noise term satisfies

$$\|\text{noise}/n\|_{\text{op}} \leq C\alpha^2 \|w - w_0\|_2^2/n, \tag{154}$$

so achieving

$$\|\text{noise}/n\|_{\text{op}} \leq \epsilon \|\Sigma\|_{\text{op}} \tag{155}$$

requires

$$n = \Omega\Big(\frac{\alpha \|w - w_0\|_2^2}{\epsilon}\Big). \tag{156}$$

The lesson here is that the error term is not scale-independent: its magnitude depends on the sub-Gaussian parameter α, which in turn affects the required batch size.

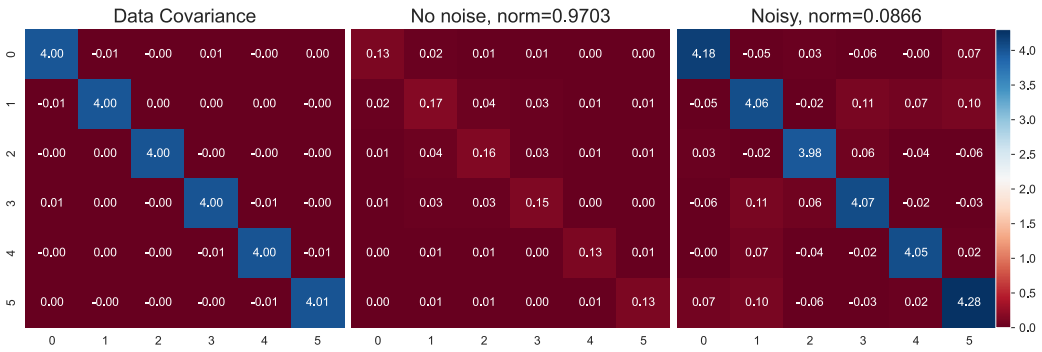

Figure 7: **Comparison of true data covariance and its estimates for the data generated from a Gaussian distribution with a diagonal covariance matrix.** Data covariance (left), its estimates by gradient covariance matrices obtained with no noise in the targets (middle), and with target noise of variance equal to batch size, $n = 256$ (right). We compare the results using an operator norm in Eq. 28.

C.3 Another Estimator to Consider

Another possible estimator is obtained by adding noise $\mathcal{N}(0, \alpha)$ to the targets and using $nS_g(w)/\alpha$ as the estimator. If the targets already contain inherent noise $\mathcal{N}(0, \widetilde{\sigma}^2)$, then by Lemma 4.2 one can see that letting $\alpha \to \infty$ makes the estimator converge to Σ. However, this comes at a cost: as $\alpha \to \infty$, the information about the mean gradient is effectively destroyed. In particular, each batch gradient becomes

$$\nabla L^{(j)}(w) = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^\top (w - w_0) - \frac{1}{n}\sum_{i=1}^{n} \mathcal{N}(0, \widetilde{\sigma}^2 + \alpha)_i \, x_i, \tag{157}$$

where the fluctuations in the second term are unrelated to the true gradient $\Sigma(w - w_0)$, dominate the first term. This raises two points:

1. If we have access to two sets of gradients for each batch, one with clean targets and the other with noisy targets, we can use the former to approximate the gradient and the latter to approximate the Hessian.

2. If we only have access to a single gradient per batch, is there a way to remove the effect of the added noise when estimating the mean gradient? In other words, can we perform post hoc variance reduction to largely undo the additional variance?

D ADDITIONAL EXPERIMENTS

D.1 Data generation from Gaussian distribution

In the experiments with data generated from a Gaussian distribution, we first sample a matrix of features from a multivariate Gaussian distribution and then pass the obtained features through a linear regression model to calculate the target values y. We add a Gaussian noise with a standard deviation of 0.1 to the target values to mimic the inherent measurement noise in real-world datasets. In the experiments, we refer to these targets as 'clean,' while the 'noisy' targets have an additional noise with variance equal to the batch size. In Section 7.1 and Appendix D.2, the number of generated features is set to 6.

D.2 Covariance estimate with a diagonal matrix

Here, we compare the covariance matrix estimates in the case of data generated from a Gaussian distribution with a diagonal covariance matrix, where the variance is 4 in both the clean target and noisy target cases (see Fig. 7). Similar to Section 7.1, we used batch size (and the noise variance) $n = 256$ with $k = 3125$.

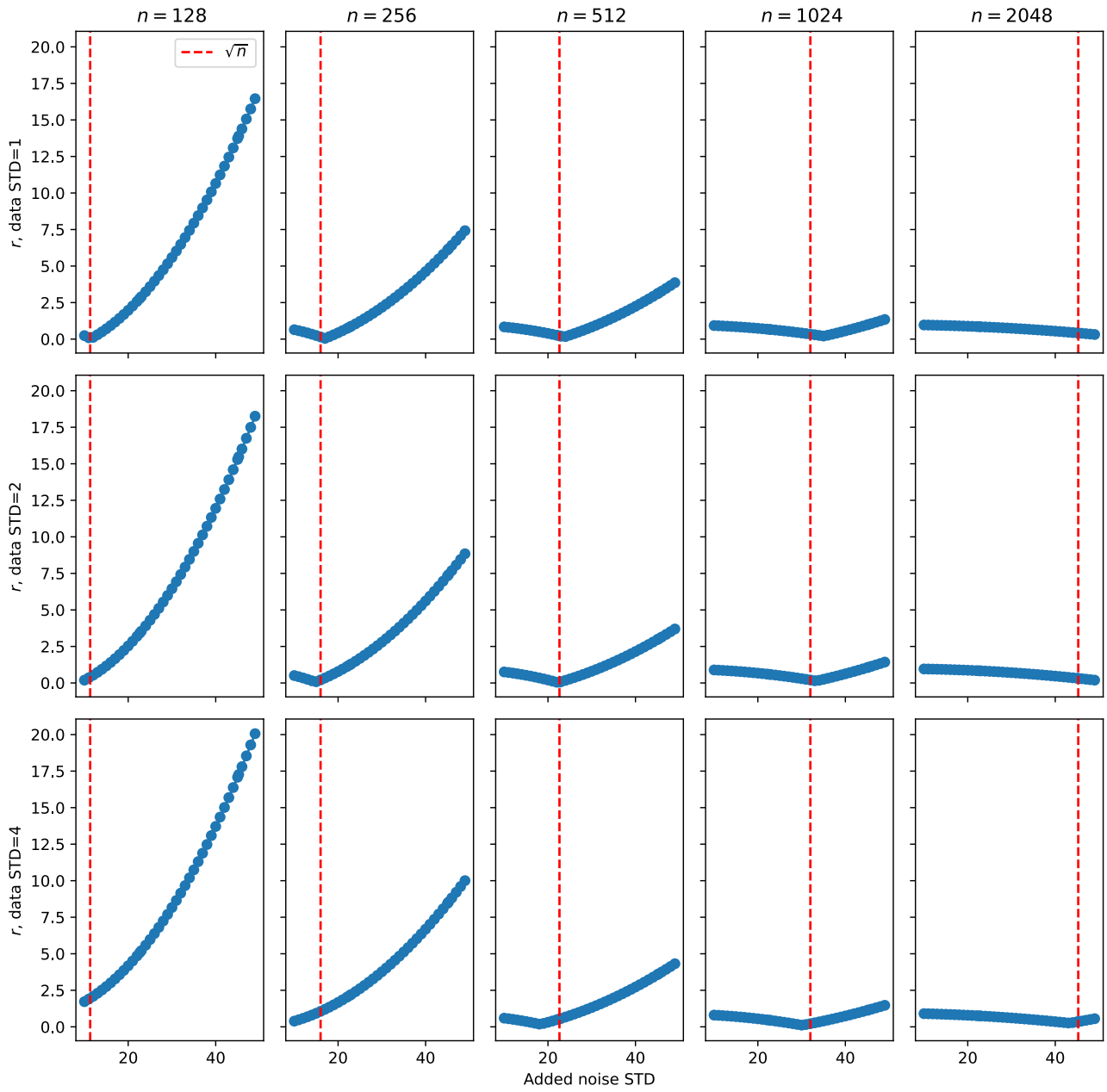

Figure 8: **Effects of the selected batch size and added noise on the estimate of the data covariance for the datasets coming from a Gaussian distribution with a diagonal covariance matrix with various standard deviations.** Rows: the results for the datasets with standard deviation equal to 1 (top), 2 (middle) and 4 (bottom). Columns: the results for batch sizes n equal to 128, 256, 512, 1024, and 2048 (from left to right). For all plots, the estimates of the covariance matrix are calculated with the standard deviation of noise added to the targets that varies from 10 to 50. The target noise standard deviation that is equal to $\sqrt{n}$ is marked with a vertical red dotted line. For all three datasets (rows), the sufficiently large batch size and the added target noise of $\sqrt{n}$ provide good estimates of the true data covariance.

D.3 Selection of batch size

The proposed method assumes the selection of a batch size as a hyperparameter. In this evaluation, we demonstrate the effects of batch size values and the corresponding added target noise on three generated datasets. We generated three datasets of 100K samples from a Gaussian distribution with a mean of zero and a diagonal covariance matrix with standard deviations of 1, 2, and 4. In this experiment, the generated data has two features. For each dataset, we calculated the relative operator norm r for the data covariance estimates with the gradient covariance matrix testing batch sizes n equal to 128, 256, 512, 1024, and 2048. For each batch size, the standard deviation of noise added to the targets was changed from 10 to 50. The results of the runs are summarised in Fig. 8. We notice that the datasets with a larger standard deviation benefit more from larger batch sizes n. See Appendix C.2 for the discussion. We also notice that the results with larger batch sizes are substantially more robust to changes in the added noise level in the targets. The noise variance, equal to the batch size, provides a reasonable estimate; however, for larger batch sizes, other values also provide good results.

D.4 Data preprocessing and training details for public datasets

In this section, we provide experimental details for the runs on four public datasets.

The wave energy converters dataset stores the values for the whole region in Watts as units. We divided the target variable by 10^6 to work with megawatts instead of watts. Additionally, we rescaled the position columns by division by 100 and the power columns by 10^4. For the California housing dataset, the 'Population' feature column was divided by 1000. For the bike sharing dataset, the target variable was transformed to a logarithmic scale. Both wine quality and bike sharing datasets exhibit correlated and redundant input features. To improve the model stability, we removed the 'density' feature from the wine quality dataset and features 'dteday', 'atemp', 'windspeed', and 'workingday' from the bike sharing dataset in the pre-processing step. For the wine dataset we divided 'total_sulfur_dioxide' and 'free_sulfur_dioxide' by 100. We represented categorical features with one-hot encodings. For each dataset, 10% of the data was reserved as a test set, the training set is used for training, and a validation set is provided for hyperparameter selection. We repeated the experiments with 10 random seeds, namely [0, 10, 20, 30, 40, 50, 60, 70, 80, 90].

Linear regression We trained all models with the Adam optimiser (Kingma, 2014). We used the following number of epochs and learning rates: for the bike sharing data, 50 epochs with a learning rate of 0.0025, for the California housing data, 300 epochs with a learning rate of 0.0001, and for the wine quality data, 100 epochs with a learning rate of 0.001. For wave energy converters, we trained the model for 25 epochs with a learning rate of 10^{-5} and then for another 25 epochs with a learning rate of 10^{-6} and then 10^{-7}. The batch size of 64 was used for all datasets considered in the experiment.

Multi-layer perceptron We trained all models with the Adam optimiser (Kingma, 2014), similar as in the case of linear models. We used the following number of epochs and learning rates: for the bike sharing data, 50 epochs with a learning rate of 0.0005, for the California housing data, 100 epochs with a learning rate of 0.0001, and for the wine quality data, 300 epochs with a learning rate of 0.001. For wave energy converters, we trained the model for 25 epochs with a learning rate of 10^{-5} and then for another 25 epochs with a learning rate of 10^{-6} and then 10^{-7}. The batch size of 64 was used for all datasets.

D.5 Hessians of MLPs with ReLU

D.5.1 Structure of the Hessian matrix

The diagonal blocks of the Hessian for an MLP network with ReLU nonlinearities are independent of the targets in the case of mean-squared error loss functions. The significance of this observation lies in the fact that, even when gradients are noisy, our Hessian estimates remain accurate for the diagonal blocks, as the Hessians of these blocks are independent of the target. The diagonal blocks of the Hessian are the sub-matrices consisting of the second derivatives with respect to each group of variables individually, where the groups of variables correspond to layer parameters. We empirically demonstrate the independence of the blocks on the targets using generated data.

Data generation We generated a dataset of 10000 samples by drawing two features from a Gaussian distribution with zero mean and standard deviation equal to 5. To obtain the targets, we pass the generated features through

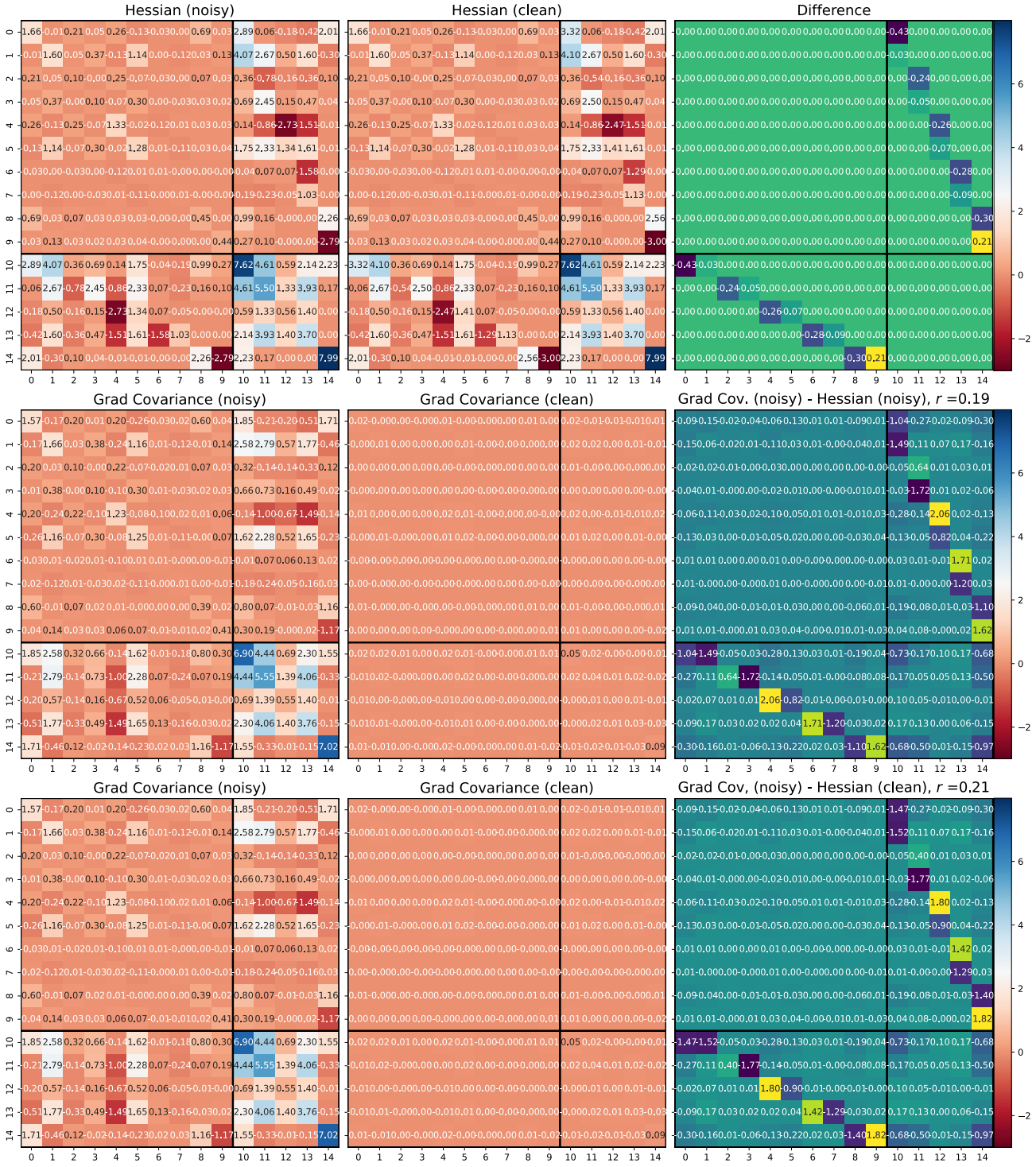

Figure 9: **Comparisons of Hessians and gradient covariance matrices computed with clean and noisy targets.** *Top*: True Hessians calculated with noisy and clean targets and their difference. *Middle*: Covariance matrices of noisy and clean gradients and the difference with the Hessian with noisy targets. *Bottom*: Covariance matrices of noisy and clean gradients and the difference with the Hessian with clean targets. The lines show diagonal blocks of Hessians corresponding to the individual layers of size ten and five.

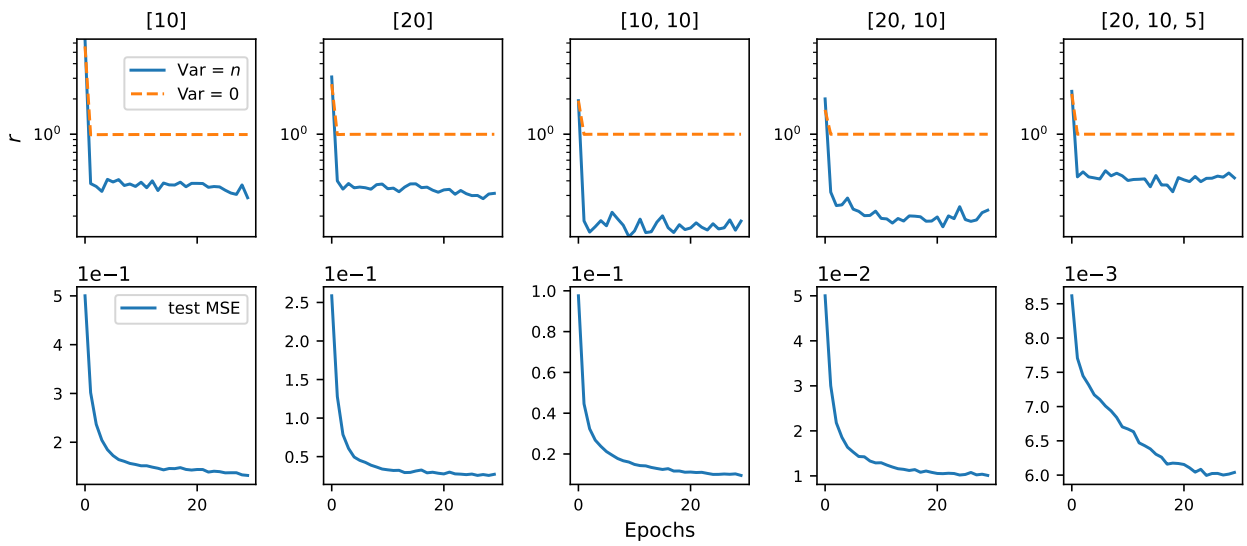

Figure 10: **Comparisons of Hessians estimated by the gradient covariance for MLPs.** The size of the network is indicated above. *Top*: The relative operator norm r for clean and noisy gradients. *Bottom*: Mean absolute error on the hold-out test set as a measure for convergence. All presented results are averaged across 10 random seeds.

a hidden layer of five neurons and an output layer, with no bias term in either the hidden or output layers.

In this experiment, we run the comparison with a batch size equal to 64 (and the noise variance equal to 8 correspondingly). In the evaluation, we used an MLP of the same size as in the data generation, with the true weights distorted by a random vector with the norm equal to 0.1. Fig. 9 demonstrates Hessian matrices obtained with clean and noisy gradients, the gradient covariance matrices of clean and noisy gradients and their difference with the true Hessians.

D.5.2 Hessian approximation on generated data with different network sizes

We further tested the quality of the estimate of the Hessians by the covariance matrix of noisy gradients during model training with MLPs of varying sizes.

In the experiment, we generated data by drawing 10K samples with 10 features from a Gaussian distribution with a zero mean and a standard deviation of five, and then passing them through an MLP (similar to the previous section). The resulting values were perturbed by adding Gaussian noise with a standard deviation equal to 0.1 to obtain the final targets. We tested the approximation of the true Hessian using the covariance matrices of the gradients obtained from the calculated targets and the targets with additional noise added, with variance equal to the batch size. We used a batch size of 64 with the Adam optimiser and a learning rate of 0.01. For each dataset, 10% of the data is reserved as a test set.

We tested the MLPs that have the hidden layers of size (a)[10], (b) [20], (c) [10, 10], (d) [20, 10], (e) [20, 10, 5] and an output layer. The initial weights for the model are set to the true parameters used for data generation, but they are distorted by a random vector with a norm of three. The results are summarised in Fig. 10. The obtained Hessian approximations are informative in a neighbourhood of the correct solution, and the results hold for the models of varying size.

D.6 Gradient covariance estimate as a pre-conditioner

In this section, we provide training details on the runs with the gradient covariance estimate as a preconditioner and provide an additional result with stochastic parameter updates performed after every 50 batches.

D.6.1 Training details

In this experiment, we run all models with a warm-up stage and then the main training stage. The warmup stage is performed with Adam optimiser, updates with noisy mean gradients and no preconditioning for all configurations tested in the main stage. We updated the model parameters after every 50 batches in the warmup stage. The main stage is presented in the results and was performed with gradient descent (GD) and Adam optimisers, the updates with clean and noisy gradients and preconditioning. The results in the main text correspond to the setting when model weights are updated once per epoch (full batch training). Additionally, we showed that preconditioning and weight updates with the noisy gradients are possible with a stochastic version of the algorithms in the main stage as well.

Full batch training We used the following number of warmup epochs, warmup learning rates, number of epochs and learning rates in the main stage: for the bike sharing data, 10 epochs with a learning rate of 0.01 (warmup), 500 epochs with a learning rate of 0.003 and 500 epochs with 0.0007 (main), for the California housing data, 10 epochs with a learning rate of 0.01 (warmup), 3000 epochs with a learning rate of 5e-4 (main), for the wine quality data, 100 epochs with a learning rate of 0.01 (warmup), 2500 epochs with a learning rate of 8e-4 and 2500 epochs with a learning rate of 2e-4 (main), and for wave energy converters, 10 epochs with a learning rate of 0.001 (warmup), 400 epochs with a learning rate of 7e-4 and 400 epochs with a learning rate of 7e-5 (main). The batch size of 64 is used for all datasets considered in the experiment.

Updates after every 50 batches (stochastic) The parameters for the warmup stage are the same as in the full batch training. For the main training stage, we used the following number of epochs and learning rates: for the bike sharing data, 500 epochs with a learning rate of 5e-4 and 500 epochs with 7e-5, for the California housing data, 3000 epochs with a learning rate of 5e-4, for the wine quality data, 2500 epochs with a learning rate of 2e-4 and 2500 epochs with a learning rate of 5e-5, and for wave energy converters, 400 epochs with a learning rate of 5e-5 and 400 epochs with a learning rate of 8e-6. The batch size of 64 is used for all datasets.

D.6.2 Stochastic optimisation results

In this section, we present additional results for the preconditioning with a Hessian estimate obtained from noisy gradients. We tested full batch training (covered in the main text) and the stochastic version of the optimisation algorithms. In the stochastic version, the batch gradients are averaged across every 50 batches to perform the update, while the preconditioner is updated once per epoch. The results are presented in Fig. 11.

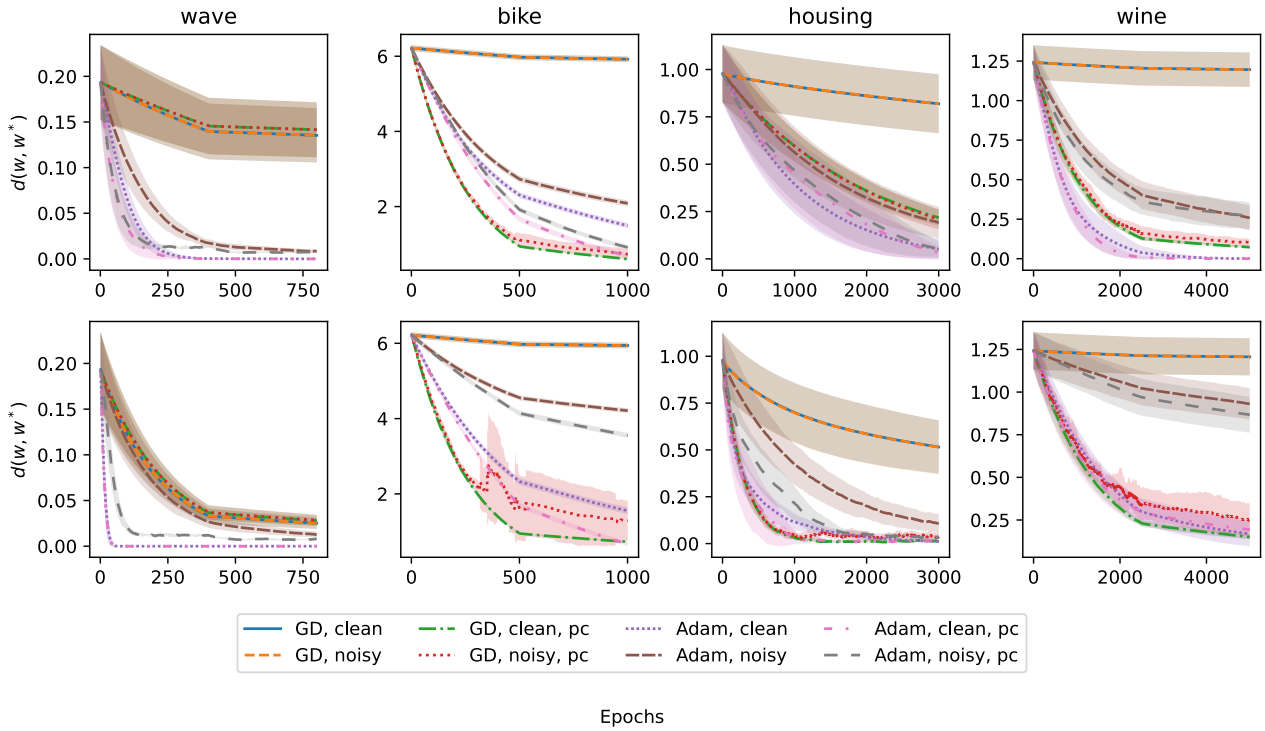

Figure 11: **Comparisons of optimisation algorithms with clean and noisy gradient updates and the gradient covariance as a preconditioner.** *Top*: full batch updates (the same as in the main text). *Bottom*: stochastic updates with the mean gradients computed from 50 batches.