Journal of Machine Learning Research 27 (2026) 1-43 Submitted 9/25; Published 2/26

Pointwise Confidence Estimation in the Non-linear ℓ^2-regularized Least Squares

Ilja Kuzborskij ILJAK@GOOGLE.COM
Google DeepMind

Yasin Abbasi Yadkori YASIN.ABBASI@GMAIL.COM

Editor: Kevin Jamieson

Abstract

We consider a high-probability non-asymptotic confidence estimation in the ℓ^2-regularized non-linear least-squares setting with fixed design. In particular, we study confidence estimation for local minimizers of the regularized training loss. We show a *pointwise* confidence bound, meaning that it holds for the prediction on any given fixed test input x. Importantly, the proposed confidence bound scales with similarity of the test input to the training data in the implicit feature space of the predictor (for instance, becoming very large when the test input lies far outside of the training data). This desirable last feature is captured by the weighted norm involving the inverse-Hessian matrix of the objective function, which is a generalized version of its counterpart in the linear setting, $x^\top \mathrm{Cov}^{-1} x$. Our generalized result can be regarded as a non-asymptotic counterpart of the classical confidence interval based on asymptotic normality of the MLE estimator. We propose an efficient method for computing the weighted norm, which only mildly exceeds the cost of a gradient computation of the loss function. Finally, we complement our analysis with empirical evidence showing that the proposed confidence bound provides better coverage/width trade-off compared to a confidence estimation by bootstrapping, which is a gold-standard method in many applications involving non-linear predictors such as neural networks.

Keywords: uncertainty quantification, non-linear least-squares

1. Introduction

One of the fundamental problems in statistics and machine learning is uncertainty quantification for some unknown ground truth function, such as constructing a high-probability confidence interval for a *fixed* function $f(x;\theta^\star)$, that is, parameterized by a vector $\theta^\star$ and evaluated at a *fixed* point x, when all we have is a collection of noisy observations $f(x_i;\theta^\star)+\mathrm{noise}_i$ at *fixed* training inputs $x_1,\dots,x_n$. Given an uncertainty estimate, the practitioner might decide to collect additional data in regions of high uncertainty or abstain from using the model if the model is deemed too unreliable.

The construction of confidence intervals (or, more generally, confidence sets) is a long-lived topic of interest at the heart of statistics (Van der Vaart, 2000; Shao, 2003; Wasserman, 2004). At the same time, confidence intervals fall in contrast with the de facto approach in machine learning literature, where one obtains a *single* estimate of $f(x;\theta^\star)$ based on the loss minimization (Shalev-Shwartz and Ben-David, 2014). While it is easy to estimate the out-of-sample performance of such standard learning approaches under the i.i.d. assumption

(say, using a held-out sample), the standard approach often falls short of identifying the limits of our knowledge about $f(x;\theta^\star)$ for an *arbitrary* input point x. This problem is particularly relevant when the input point x is far from the training data.

A notable success is a linear regression setting where the ground truth is given by $f(x;\theta^\star) = x^\top\theta^\star$, and where the solution to a (regularized) linear least-squares problem $\widehat{\theta}$ can be used to state a *high-probability* confidence interval of a form $|x^\top\widehat{\theta} - x^\top\theta^\star|^2 \lesssim x^\top\Sigma^{-1}x/n$ where Σ is a covariance matrix. Here an important term is the *weighted norm* $x^\top\Sigma^{-1}x$ that captures how well the test input x is represented in the span of training inputs, and so if x is very dissimilar from the training data, the norm will be large. Similar confidence intervals can also be stated in the non-linear case by lifting derivations to the formalism of kernel methods (Rasmussen and Williams, 2006; Srinivas et al., 2010; Yadkori, 2012; Chowdhury and Gopalan, 2017).

In this paper we focus on uncertainty estimation of *non-linear* ground truth functions which can be modelled by highly non-linear predictors, such as overparameterized neural networks. While a large body of recent literature has explored the practical adaptation of ideas from statistics to neural networks, such as bootstrap and ensemble methods (Dwaracherla et al., 2023), these tend to require strong assumptions on the distribution of the test input point x—usually that it is drawn i.i.d. from the same distribution as the training set. Another limitation of these approaches is computational: to control their variance, a large number of dataset replications and re-training steps are required, which is impractical in the context of such predictors.

In contrast to these approaches, in this paper we are motivated by the presence of the weighted norm beyond the linear case. Intuitively, one would expect that in the non-linear case the weighted norm should bear a similar meaning as in the linear case, but now capturing whether the test point resides in the span of some implicit 'features' learned by the algorithm. Moreover, we ask whether one can attain high-probability non-asymptotic confidence intervals that would be shaped by this kind of weighted norm. To this end, the concept of weighted norms in this context is not new. A fundamental result in parametric estimation going back to works of R. Fisher is the asymptotic normality of the Maximum Likelihood Estimator (MLE) $\widehat{\theta}$, namely $\sqrt{n}\,(\widehat{\theta} - \theta^\star) \xrightarrow{d} \mathcal{N}(0, \mathcal{I}(\theta^\star)^{-1})$, where the ground truth is parameterized by a fixed vector $\theta^\star$ and where $\mathcal{I}(\theta^\star)$ is a *Fisher information matrix* (Van der Vaart, 2000). This result is used to give various *asymptotic* confidence sets for $\theta^\star$, such as Wald-type elliptic sets and a so-called linearized Laplace approximation in the context of Bayesian literature (Mackay, 1992; Khan et al., 2019; Antorán et al., 2022) (see Appendix A for a detailed discussion of related work). Constructing confidence intervals on $f(x;\theta^\star)$ based on asymptotic normality would indeed involve a generalized weighted norm defined with respect to the Fisher information matrix. However, such methods occasionally suffer from coverage issues due to their asymptotic nature and a lack of high-probability guarantee, especially when the sample is small. In this paper we explore this problem from the viewpoint of concentration inequalities by focusing on the *point variance* of the estimator:

$$|f(x;\widehat{\theta}) - f(x;\theta^\star)| \le \underbrace{|f(x;\widehat{\theta}) - \mathbb{E}[f(x;\widehat{\theta})]|}_{\text{Point variance}} + \underbrace{|\,\mathbb{E}[f(x;\widehat{\theta})] - f(x;\theta^\star)|}_{\text{Bias}}\,. \tag{1}$$

Note that the point variance is the only *random* component of the confidence interval. Here we do not consider a *deterministic bias* of the estimator (a practice shared by standard

methods such as bootstrapping), controlling which is typically a hard problem on its own requiring assumptions about the regularity of the ground truth function (Györfi et al., 2006).

Our contributions In this work we show a high-probability non-asymptotic confidence bound that holds generally in the ℓ^2-regularized non-linear least-squares setting with fixed design (Theorem 2). The only requirement that we impose on the solution $\widehat{\theta}$ is to be a stationary point of this (differentiable) problem with an invertible Hessian. Note that this is a very general setting where f might be a highly non-linear predictor, such as a neural network. In particular, we show that with high probability

$$|f(x;\widehat{\theta}) - \mathbb{E}[f(x;\widehat{\theta})]| = \mathcal{O}_P\left(\frac{\|\nabla_\theta f(x;\widehat{\theta})\|_{M(\widehat{\theta})}}{\sqrt{n}}\right) \qquad \text{as} \qquad n \to \infty$$

where $M(\theta)$ is a particular Positive Semi-Definite (PSD) empirical weighting matrix that asymptotically converges to the inverse of the Fisher information matrix. In Section 3 we study this bound in various settings and show that it recovers known results for the linear setting, confidence bounds with weighted norms, and that it is asymptotically correct, meaning that it matches the shape of the confidence interval based on asymptotic normality.

Next, we propose an efficient algorithm for computation of the weighted norm, which exploits automatic differentiation techniques to attain the worst-case computational complexity of order $\tilde{\mathcal{O}}(pn)$ in contrast to a naive computation of order $\mathcal{O}(p^3 + pn)$ where p is the number of parameters and n is the sample size. Meanwhile, the memory complexity is of the order $\mathcal{O}(p)$ instead of $\mathcal{O}(p^2)$. In the interpolation regime, the computational cost reduces to $\tilde{\mathcal{O}}(\max(p, C_L))$, where C_L is the cost of evaluating the loss function.

Finally, we evaluate our bound experimentally in Section 5 and show that it is able to attain a good coverage / width trade-off on a non-linear problem even in the presence of regions of the input space without observations. In comparison, we show that bootstrapping, as a representative often used go-to method, exhibits under-coverage.

2. Preliminaries

Throughout, $\|\cdot\|$ is understood as the Euclidean norm for vectors and the Frobenius norm for matrices. For some vector $x \in \mathbb{R}^p$ and a PSD matrix $A \in \mathbb{R}^{p\times p}$, the weighted Euclidean norm is defined as $\|x\|_A^2 = x^\top A x$. We will use a shorthand notation for gradient evaluation at a point, $\nabla_\theta f(\dots, \theta', \dots) = (\mathrm{d}f/\mathrm{d}\theta)|_{\theta=\theta'}$, for example, $\nabla_\theta f(\dots, \widehat{\theta}, \dots)$, $\nabla_x f(\dots, x_i, \dots)$.

We will also work with unbounded random variables which are often captured through the following tail conditions (see also Boucheron et al. (2013, Section 2.4)).

Definition 1 (Sub-gamma random variable) *A random variable is called sub-gamma on the right tail with variance factor v^2 and scale $c > 0$ if*

$$\ln \mathbb{E}[e^{\lambda(X-\mathbb{E}[X])}] \le \frac{\lambda^2(v^2/2)}{1-c\lambda} \qquad \textit{for every} \quad 0 < \lambda < 1/c\ .$$

Similarly, X is called sub-gamma on the left tail if $-X$ is sub-gamma on the right tail with parameters (v^2, c). Finally, X is called sub-gamma if conditions on both tails hold simultaneously.

One intuitive consequence of a sub-gamma condition is that it implies tail bounds $\mathbb{P}\{X - \mathbb{E}[X] > \sqrt{2v^2t} + ct\} \leq e^{-t}$ and $\mathbb{P}\{\mathbb{E}[X] - X > \sqrt{2v^2t} + ct\} \leq e^{-t}$, which tells us that depending on its parameters, it might concentrate around its mean at rates ranging from $\sqrt{2v^2t}$ to ct. Note that when $|X|$ is bounded by B, we have $v^2 = B^2/4$ and $c = 0$. Sub-gamma tail conditions are weaker than sub-Gaussianity but stronger than merely assuming finite variance.

2.1 Setting

In the following we will look at the parameterized predictors $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$; while $f(x;\theta)$ will stand for evaluation on an input $x \in \mathbb{R}^d$ given parameters $\theta \in \mathbb{R}^p$. In practice, parameters $\widehat{\theta}$ are found given training examples $(x_1, Y_1), \ldots, (x_n, Y_n) \in \mathbb{R}^d \times \mathbb{R}$. In this paper, we work in a *fixed-design regression setting* where inputs $x_1, \ldots, x_n$ are fixed, while responses $Y_1, \ldots, Y_n$ are random. In particular, each response is generated as

$$Y_i = f(x_i;\theta^\star) + Z_i \ , \qquad Z_i \sim \mathcal{N}(0,\sigma^2) \ . \tag{2}$$

where $f(\cdot\,;\theta^\star) : \mathbb{R}^d \to \mathbb{R}$ is some unknown fixed regression function. Then, the *empirical risk* of θ with respect to the square loss and its ℓ^2-regularized counterpart are respectively defined as

$$L(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f(x_i;\theta) - Y_i)^2 \ , \qquad L_\lambda(\theta) = L(\theta) + \frac{\lambda}{2}\|\theta\|^2 \ .$$

Given a *fixed* test point x our goal in this paper is to estimate the *point variance*, which we will see as a proxy to uncertainty within the model arising due to randomness in responses, and which is an important part in a general pointwise uncertainty estimation problem, Eq. (1). In this paper, we do not control a (deterministic) bias of a learning algorithm, unless in some sanity-check scenarios (such as when $f(x;\theta^\star)$ is linear). Controlling the bias typically requires smoothness assumptions on the regression function $f(x;\theta^\star)$ (such as membership in a certain Sobolev space), which in turn requires to show that the algorithm is able to learn well in very large (non-parametric) classes of functions. As a consequence, such bounds are often quite pessimistic and exhibit the curse of dimensionality (Györfi et al., 2006). Many gold-standard confidence estimation methods, such as the ones derived from bootstrap (Efron and Tibshirani, 1994), forego bias control.

In particular, in this paper we will study point variance estimation using $x \mapsto f(x\,;\widehat{\theta})$, where parameter vector $\widehat{\theta}$ is assumed to be a local minimizer of L_λ. Moreover, we require L_λ to be differentiable in θ. This implies that $\widehat{\theta}$ is a stationary point of L_λ, or in other words, $\nabla L_\lambda(\widehat{\theta}) = 0$. In turn, this implies an important property of *alignment* between the parameter vector and the gradient:

$$\lambda\widehat{\theta} = -\nabla L(\widehat{\theta}) \ . \tag{3}$$

Note that the stationarity is asymptotically achieved (and so is alignment) under appropriate smoothness and boundedness conditions by many optimization algorithms used in practice, such as Gradient Descent (GD) (and its stochastic counterpart) (Ghadimi and Lan, 2013), as well as adaptive ones such as Adam/Adagrad (Défossez et al., 2022; Li et al., 2023), and AdamW (Zhou et al., 2024). As an example, we show the alignment property for GD in Appendix B.

3. Main Results

Our first result is a high-probability concentration inequality on the point variance gap

$$\Delta(x;\widehat{\theta}) = f(x;\widehat{\theta}) - \mathbb{E}[f(x;\widehat{\theta})] \ .$$

Here we assume that $\widehat{\theta}$ is any local minimizer of L_λ, and we require the bound on the gap to be given in terms of a data-dependent quantity that captures sensitivity of the predictor to new observations. In the following, we will often refer to this quantity as the *weighted norm* with respect to $(\widehat{\theta}, x)$, and which is given by $\|\nabla_\theta f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}$ where we define a PSD matrix

$$M(\theta) = \big(\nabla^2 L(\theta) + \lambda I\big)^{-1} \left(\frac{1}{n}\sum_{i=1}^n \nabla_\theta f(x_i;\theta)\nabla_\theta f(x_i;\theta)^\top\right) \big(\nabla^2 L(\theta) + \lambda I\big)^{-1} \ .$$

Note that the weighted norm is random only in the responses $Y_1, \ldots, Y_n$, and moreover it can potentially be unbounded. To capture the fact that it is unbounded in our analysis, it is natural to assume standard tail conditions for such a scenario. In particular, here we will assume that the weighted norm is a *sub-gamma* random variable with variance factor v^2 and scale $c > 0$, which captures many cases of bounded and unbounded random variables (see Section 2 for definition and discussion).

Now we present our main result, which shows that the point variance gap is controlled in terms of the weighted norm (the proof is given in Section 4):[1]

Theorem 2 *Let $\widehat{\theta}$ be any stationary point of L_λ such that $\nabla^2 L_\lambda(\widehat{\theta})$ is invertible, and assume that $\theta \mapsto f(x;\theta)$ is differentiable at $\widehat{\theta}$. Assume that the weighted norm with respect to $(\widehat{\theta}, x)$ is a sub-gamma random variable with variance factor v^2 and scale c. Then, for any fixed test input x and for any $\delta \in (0,1)$*

$$\mathbb{P}\left\{|\Delta(x;\widehat{\theta})| > \sigma\sqrt{\frac{\pi^2\ln(2/\delta)}{n}\,\mathbb{E}\left[\|\nabla f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}\right]} + \sigma\,\frac{6\ln^{3/4}(2/\delta)\sqrt{v} + 9\ln(2/\delta)\sqrt{c}}{\sqrt{n}}\right\} \leq \delta$$

Note that an empirical bound can be stated from the above as a consequence of the sub-gamma assumption, where the expected weighted norm is replaced by $\|\nabla_\theta f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})} + \sqrt{2v^2\ln(2/\delta)} + c\ln(2/\delta)$.

Sanity check for linear least-squares First, consider a linear model where $f(x;\theta^\star) = x^\top\theta^\star$ for some fixed ground-truth parameter vector $\theta^\star$. Then $f(x;\theta) = x^\top\theta$ and we employ a well-known ridge regression estimator

$$\widehat{\theta} = \Sigma^{-1}\left(\frac{1}{n}\sum_{i=1}^n x_i Y_i\right) \qquad \text{where} \qquad \Sigma = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \lambda I \ .$$

Now, it is straightforward to check that (see Appendix C.1)

$$\mathbb{E}[f(x;\widehat{\theta})] = x^\top\theta^\star - \lambda\, x^\top\Sigma^{-1}\theta^\star \qquad \text{and} \qquad \mathbb{E}\left[\|\nabla_\theta f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}\right] \leq \|x\|^2_{\Sigma^{-1}}$$

1. One can also show a weaker result (with dependence $1/\sqrt{\delta}$ rather than $\ln(1/\delta)$), which does not require any tail assumptions on the weighted norm at all (see Appendix D.1).

and so we recover a confidence interval for such a case:

$$\mathbb{P}\left\{|x^\top(\widehat{\theta}-\theta^\star)| > \lambda\left|x^\top\Sigma^{-1}\theta^\star\right| + \sigma\sqrt{\frac{(\pi^2/2)\ln(2/\delta)}{n}}\,\|x\|_{\Sigma^{-1}}\right\} \leq \delta\ .$$

This confidence interval, importantly, scales with the weighted norm $\|x\|_{\Sigma^{-1}}$, has a bias term $\lambda\left|x^\top\Sigma^{-1}\theta^\star\right|$, and matches known high-probability confidence intervals for ridge regression (with slightly worse constant), see for instance (Lattimore and Szepesvári, 2020, Chap. 20). The dependence on this weighted norm is optimal in the fixed and adaptive design case (Lattimore, 2023).

Relationship of matrix $M(\theta)$ to inverse of the Fisher information matrix Matrix $M(\widehat{\theta})$ appearing in the weighting of $\|\nabla f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}$, is closely related to the inverse of the *Fisher information matrix* $\mathcal{I}(\theta)$ appearing in theory of parametric estimation (more on that in the next paragraph). In the context of non-linear least-squares model considered here, $\mathcal{I}(\theta) = \mathbb{E}[\nabla^2 L(\theta)]$, whose inverse is somewhat different from $M(\theta)$. However, asymptotically under some conditions, they coincide.

A better case for our regression model is when f predicts nearly as well as the regression function $f(x;\theta^\star)$, or generalizes well. In this case, one can show that $\mathbb{E}\,M(\theta^\star)$ is essentially replaced by the inverse of $\mathcal{I}(\theta^\star)$ assuming a particular data-dependent tuning of $\lambda \to 0$ (see Proposition 15),

$$\mathbb{E}\left[\|\nabla_\theta f(x;\theta^\star)\|^2_{M(\theta^\star)}\right] \leq \|\nabla_\theta f(x;\theta^\star)\|^2_{\mathcal{I}(\theta^\star)^{-1}} + C\cdot\frac{\sigma^2}{n}\ .$$

Note that typically we learn such a good predictor when $n\to\infty$ and as $\lambda\to 0$, and so the above dependence on n and λ becomes negligible.

Asymptotic efficiency and connection to MLE Now we look at another theme closely related to the Fisher information. Let $\widehat{\theta}^{\mathrm{MLE}}_n$ be the MLE of the true fixed parameter vector $\theta^\star$. It is well-known that MLE satisfies asymptotic normality under several standard assumptions, including *consistency* in the sense that $\widehat{\theta}^{\mathrm{MLE}}_n \longrightarrow \theta^\star$ in distribution as $n\to\infty$ (Shao, 2003). Then, in combination with the delta method,

$$\sqrt{n}\,\left(f(x;\widehat{\theta}^{\mathrm{MLE}}_n) - f(x;\theta^\star)\right) \xrightarrow{d} \mathcal{N}\left(0, \|\nabla_\theta f(x;\theta^\star)\|^2_{\mathcal{I}(\theta^\star)^{-1}}\right)\ .$$

In comparison, Theorem 2 can be thought of as a non-asymptotic variant of this result adapted to non-linear least-squares. To demonstrate this, consider the asymptotic bound implied by Theorem 2. Under some differentiability assumptions, and the convergence $\widehat{\theta}_n \xrightarrow{a.s.} \widehat{\theta}_\infty$, we almost surely have

$$\limsup_{n\to\infty}\frac{\sqrt{n}}{\ln(n)}\,\left|f(x;\widehat{\theta}_n) - \mathbb{E}[f(x;\widehat{\theta}_n)]\right| \leq \sigma\pi\sqrt{\mathbb{E}\,\|\nabla_\theta f(x;\widehat{\theta}_\infty)\|^2_{M(\widehat{\theta}_\infty)}}\ .$$

This bound carries a similar message to the one we get from the asymptotic normality of MLE: the prediction asymptotically concentrates around the expected prediction and the 'strength' of the concentration is attenuated by the direction x (see Proposition 16 in Appendix C for the proof).

However, this statement is weaker since we do not speak about convergence to the ground truth $f(x;\theta^\star)$. At this point, similarly to the case of the asymptotic normality of the MLE we can assume that the algorithm is consistent in a sense that $\widehat{\theta}_\infty = \theta^\star$. Then, the above asymptotic result combined with an earlier observation that $\mathbb{E}\, M(\widehat{\theta}_\infty)$ converges to the inverse of the Fisher information matrix,

$$\limsup_{n\to\infty} \frac{\sqrt{n}}{\ln(n)} \left| f(x;\widehat{\theta}_n) - f(x;\theta^\star) \right| \leq \sigma\pi \sqrt{\|\nabla_\theta f(x;\theta^\star)\|^2_{\mathcal{I}(\theta^\star)^{-1}}} \ .$$

Uniform bound over test inputs Theorem 2 holds only for a fixed x, however in practice, we might want to have a confidence bound that holds over all test inputs within some set simultaneously, for example, over a unit sphere $\mathbb{S}^{d-1}$. One standard way to achieve this is through the covering number technique that we employ here (see Appendix C.3 for details). As an example, here we look at the case when f is a multilayer network trained to stationarity of L_λ:

$$f(x;\theta) = {w_K}^\top h_{K-1}, \quad h_k = a(W_k h_{k-1}), \quad h_0 = x \qquad (k \in [K-1]), \tag{4}$$

where x represents the input. Here $W_1, \ldots, W_K$ are weight matrices collectively represented by the parameter vector $\theta = (\mathrm{vec}(W_1), \ldots, \mathrm{vec}(W_K))$, and $a : \mathbb{R}^m \to \mathbb{R}^m$ is either a ReLU or a linear activation function. To avoid potentially pathological non-differentiable cases we consider the following:

Assumption 1 *Assume that $\widehat{\theta}$ is a stationary point of L_λ and that no training example falls on the activation boundary, i.e., $\min_{i,j,k} |{x_k}^\top \widehat{w}_{i,j}| > 0$ where $k \in [n]$ indexes training inputs, $i \in [K-1]$ indexes layers, and j indexes rows of W_i,[2] and analogously for the test point we assume that $\min_{i,j} |x^\top \widehat{w}_{i,j}| > 0$.*

Proposition 3 *Let $\texttt{bound}(\cdot)$ denote the bound on $\Delta(x;\widehat{\theta})$ given in Theorem 2, as a function of the $\ln(2/\delta)$ term. Then, under Assumption 1 and that $L_\lambda(\widehat{\theta}) \leq L_\lambda(0) = C$, we have*

$$\mathbb{P}\left\{ \max_{x\in\mathbb{S}^{d-1}} |\Delta(x;\widehat{\theta})| > \texttt{bound}\left(d\,\ln\left(\frac{6n}{\delta}\right) + \frac{Kd}{2}\ln\left(\frac{C}{\lambda K}\right)\right) + \frac{1}{n} \right\} \leq \delta \ .$$

See Appendix C.3 for the proof. We pay a factor $\sqrt{Kd\ln(1/\lambda)}$ in the final bound, which is a mild cost compared to the linear case where we would only incur $\sqrt{d}$, considering that the number of parameters can be much larger than d.

Relationship to the Neural Tangent Kernel (NTK) To compare our approach to the kernel setting, we first briefly sketch the proof idea of Theorem 2. Our approach can be summarized as follows: the first component is Lemma 7, whose key part states that

$$\frac{\partial\widehat{\theta}}{\partial z_i} = \frac{1}{n}(\lambda I + \nabla^2 L(\widehat{\theta}))^{-1} \nabla_\theta f(x_i;\widehat{\theta}) \tag{5}$$

2. This condition does not require all neurons to be active; neurons may be strictly inactive. It only excludes the nondifferentiable case in which a preactivation is exactly zero.

where on the left we differentiate $\widehat{\theta}$ with respect to noise variable z_i. Then, the chain rule implies that

$$\|\nabla_{(z_1,\ldots,z_n)} f(x;\widehat{\theta})\|^2 = \frac{1}{n}\,\|\nabla_\theta f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}$$

and, finally, Lemma 10 states a concentration inequality in terms of $\mathbb{E}[\|\nabla_{(z_1,\ldots,z_n)} f(x;\widehat{\theta})\|^2]$. To make the connections with kernel regression clear, notice that Eq. (5) implies a linear approximation

$$\widehat{\theta} \approx \theta' + \frac{1}{n}(\lambda I + \nabla^2 L(\widehat{\theta}))^{-1} \sum_i Z_i \nabla_\theta f(x_i;\widehat{\theta})\,,$$

where θ' would be the solution if all noise terms were zero. With the choice of

$$\theta' = \frac{1}{n}\left(\lambda I + \nabla^2 L(\widehat{\theta})\right)^{-1} \sum_{i=1}^n \nabla_\theta f(x_i;\widehat{\theta}) \nabla_\theta f(x_i;\widehat{\theta})^\top \theta^\star$$

the above approximation has the exact same form as the least-squares solution when $\nabla_\theta f(x_i;\widehat{\theta})$ does not depend on the noise terms (that is, if we had initialized θ with $\widehat{\theta}$), and the model is a linear map with weight vector $\theta^\star$. (Note that this particular choice of θ' is meant to illustrate the analogy with the linear case and does not imply that θ' is known or computable.) The NTK is such a case, where the kernel is defined with feature maps that only depend on the initialization (Jacot et al., 2018),

$$K_0(x,x') = \frac{1}{n}\left\langle \nabla_\theta f(x;\theta_0), \nabla_\theta f(x';\theta_0)\right\rangle\,.$$

For Eq. (5) to hold, we need regularization (for the alignment condition of Eq. (3) to hold) and $\widehat{\theta}$ must be a stationary point, and no linearity is needed. Therefore, our result is more general and applies to the stationary point of the gradient descent process, $\widehat{\theta}$, which is equivalent to a kernel regression with feature map $\nabla_\theta f(x_i;\widehat{\theta})$. To see this, define time-varying kernel

$$K_t(x,x') = \frac{1}{n}\left\langle \nabla_\theta f(x;\theta_t), \nabla_\theta f(x';\theta_t)\right\rangle\,.$$

The update in the function space after a small gradient update with step size η can be written as

$$\begin{aligned} f(x;\theta_{t+1}) &\approx f(x;\theta_t) + \langle \nabla_\theta f(x;\theta_t), \theta_{t+1} - \theta_t\rangle \\ &= f(x;\theta_t) - \left\langle \nabla_\theta f(x;\theta_t), \frac{\eta}{n}\sum_i (f(x_i;\theta_t) - y_i)\nabla_\theta f(x_i;\theta_t)\right\rangle \\ &= f(x;\theta_t) - \eta \sum_i K_t(x,x_i)(f(x_i;\theta_t) - Y_i)\,. \end{aligned}$$

Our analysis is also more general. For obtaining uniform bounds in the linear case, the typical approach is to use the Cauchy-Schwarz inequality to separate the weighted norm from the noise, and then show a self-normalized bound for the noise term. This approach is heavily based on linearity. Our approach, on the other hand, requires the parameter solution to be differentiable with respect to noise, and is not applicable to, for example, Bernoulli noise.

Also, the weighted norm on the r.h.s. in the linear case does not involve an expectation and no sub-gamma assumption is needed. An alternative approach would be to split the data, train the network with the first half, learn the "features", and then apply kernel regression on the second half and construct confidence intervals. A common data-splitting approach in practice is to learn the feature maps up to the last layer, and then use extra data to construct confidence ellipsoids around the weight matrix of the last linear layer.

3.1 Role of Bias

Theorem 2 controls only the *point variance* component of $|f(x;\widehat{\theta}) - f(x;\theta^\star)|$, leaving the $\text{bias}(x) = \mathbb{E}[f(x;\widehat{\theta})] - f(x;\theta^\star)$ uncontrolled. In principle, bias could be dominant, for instance, when regularization is large or the model is misspecified, thereby limiting the practical utility of a tight variance-only bound. However, the point variance bound is most meaningful in the small-regularization, well-specified regime. For example, in the linear case as discussed above, the bias is exactly $\lambda|x^\top\Sigma^{-1}\theta^\star| = \mathcal{O}(\lambda)$, which vanishes as $\lambda \to 0$, and the variance term $\|x\|_{\Sigma^{-1}}/\sqrt{n}$ dominates the bias when $\lambda \ll \|x\|_{\Sigma^{-1}}/(\sqrt{n}\,|x^\top\Sigma^{-1}\theta^\star|)$, i.e., when $\lambda = o(1/\sqrt{n})$ with the test-point geometry treated as fixed. In the nonlinear case, under consistency ($\widehat{\theta}_n \to \theta^\star$ a.s., see discussion on asymptotic efficiency and connection to MLE above), the bias vanishes as $n \to \infty$ while the variance decays at rate $1/\sqrt{n}$ as captured by Theorem 2. Some numerical evidence presented in Section 5 shows good empirical coverage, which is consistent with the experiments operating in the variance-dominated regime where bias is small relative to the CI width.

One might wonder if bias can be estimated practically without theoretical quantification. To this end, consider bias-variance decomposition

$$\mathbb{E}\left[(Y - f(x;\widehat{\theta}))^2\right] = \text{bias}(x)^2 + \text{Var}(f(x;\widehat{\theta})) + \sigma^2 \ .$$

Thus, even if we had knowledge of noise rate σ^2 and managed to estimate point variance of an estimator (for instance, by using Theorem 2), we would not be able to estimate the squared bias unless we knew the prediction MSE at x, i.e., $\mathbb{E}[(f(x;\widehat{\theta}) - f^\star(x))^2]$. Indeed, in a fixed-design setting the observations only reveal information about $f(x_1;\theta^\star), \ldots, f(x_n;\theta^\star)$, while $f(x;\theta^\star)$ is not identified from the data when x is not one of the design points. Consequently, without either an independent test response Y at x or additional structural assumptions on $f^\star$, the pointwise squared bias $\text{bias}(x)^2$ cannot be estimated from the available data alone.

3.2 Discussion on Assumptions

Invertibility of regularized Hessian in $M(\widehat{\theta})$ In Theorem 2 we assumed that regularized Hessian is invertible which enables computation of the weighted norm. This might appear to be a strong assumption; for a sufficiently large λ (strong regularization) it is indeed satisfied, though possibly at the expense of a higher empirical risk.

Alternatively, while it is possible to replace the matrix inverse with the pseudo-inverse, this will introduce an additional term capturing the extent to which $\nabla_\theta f(x;\widehat{\theta})$ lies in the nullspace of the regularized Hessian. In practice it is possible to estimate whether this component is small numerically, however computationally this diagnostic is comparable in cost to computing weighted norm itself (see Section 3.3). Instead, a more practical way is to

rely on the behaviour of the numerical solver itself, for example failure to converge which indicates that nullspace component is non-zero. On the other hand, even with a singular Hessian one might still converge to a pseudo-inverse solution when the nullspace component is zero. In Section 3.3 where we discuss efficient computation of the weighted norm, we cover all such cases in Table 1.

On sub-gamma assumption Observe that if the weighted norm is bounded, then the sub-gamma condition is trivially satisfied (every bounded random variable satisfying $|X| \leq B$ a.s. is sub-gamma random variable with $v = B$, $c = 0$). To explore such cases in a bit more detail consider the following lemma:

Lemma 4 *For any $\theta \in \mathbb{R}^p$ and any x,*

$$\|\nabla_\theta f(x;\theta)\|^2_{M(\theta)} \leq \frac{\|\nabla_\theta f(x;\theta)\|^2}{4\left(\lambda - \sqrt{2L(\theta)\,\mu(\theta)}\right)_+} \qquad \textit{where} \qquad \mu(\theta) = \frac{1}{n}\sum_{i=1}^n \|\nabla^2_\theta f(x_i;\theta)\|^2_{\mathrm{op}} \,.$$

See Appendix C.4 for the proof. This lemma reveals one regime of interest where sub-gamma condition is trivially satisfied, namely assuming that $\|\nabla_\theta f(x;\widehat{\theta})\| \leq B$, $\|\nabla^2_\theta f(x;\widehat{\theta})\| \leq C$ almost surely and that $\lambda > C\sqrt{2L(\widehat{\theta})}$. While the first two assumptions might be enforced by designing the predictor, that is to have bounded first and second derivatives, the last one is problem dependent, but can be verified in practice by simply computing the training loss.

As a straightforward analytical example we consider a shallow Rectified Linear Unit (ReLU) neural network

$$f(x;\theta) = u^\top (W^\top x)_+ \qquad \text{where} \qquad u \in \mathbb{R}^m \quad \text{and} \quad \theta = \mathrm{vec}(W)\,, \quad W \in \mathbb{R}^{d\times m}\,.$$

In other words, only the hidden layer is trained, while the outer layer is fixed. Such setting is common in the analysis of neural networks (Jacot et al., 2018; Du et al., 2018; Arora et al., 2019). Then, under Assumption 1 and by Cauchy-Schwarz inequality we have that $B = \|x\|\|u\|$ and $C = 0$ and so by Lemma 4

$$\|\nabla_\theta f(x;\widehat{\theta})\|_{M(\widehat{\theta})} \leq \frac{\|x\|\|u\|}{2\sqrt{\lambda}}$$

which implies $(\|x\|\|u\|/\sqrt{4\lambda}, 0)$-sub-gamma condition.

Fixed vs random design The key component of our proof relies on the fact that differentiable functions of Gaussian vectors concentrate well if their gradients are well-behaved (Lemma 8). Here we carry out analysis by assuming that the noise in response is a component of a Gaussian vector. In order to extend Theorem 2 to the random design setting, one possibility is to follow the same proof technique assuming that covariates are Gaussian and apply relevant steps for the joint distribution of covariates and responses. However, this introduces a rather strong assumption.

A more general solution (and more challenging problem) is to study whether

$$g(X_1,\ldots,X_n) = \mathbb{E}[f(x;\widehat{\theta}) \mid X_1,\ldots,X_n]$$

concentrates well around its expectation using some property of the algorithm generating $\widehat{\theta}$, for example its *algorithmic stability* (Bousquet and Elisseeff, 2002). In particular, in our case the solution is algorithmically stable if replacement or removal of one covariate in the sample does not alter its prediction too much. Now we look at such an example more concretely. By the standard McDiarmid's inequality the desired concentration bound is

$$\mathbb{P}\left\{|g(X_1,\dots,X_n)-\mathbb{E}[g(X_1,\dots,X_n)]|>\sqrt{n\beta^2\,\frac{\ln(2/\delta)}{2}}\right\}\leq\delta$$

where a uniform stability coefficient $\beta>0$ is such that

$$\sup_{i,x_1,\dots,x_n,x'}|g(x_1,\dots,x_n)-g(x_1,\dots,x_{i-1},x',x_{i+1},\dots,x_n)|\leq\beta\ .$$

While there exists a vast literature studying stability of various algorithms (Devroye and Wagner, 1979; Bousquet and Elisseeff, 2002; Shalev-Shwartz and Ben-David, 2014; Bousquet et al., 2020; Vary et al., 2026) here we focus on scenarios where regularized empirical risk satisfies Polyak-Łojasiewicz (PL) condition (Karimi et al., 2016), namely there exists constant $\mu>0$ such that

$$\|\nabla L_\lambda(\theta)\|^2\geq 2\mu(L_\lambda(\theta)-L_\lambda(\widehat{\theta}))\qquad(\theta\in\mathbb{R}^p)$$

and moreover assume that $\widehat{\theta}$ is a *global minimizer* of L_λ. This condition is interesting in our context because it naturally encompasses linear least squares (strongly convex L_λ satisfies PL condition with $\mu=\lambda$) and some basic non-linear cases: it is often studied as a proxy to neural network learning settings, such as multilayer linear and leaky ReLU neural networks (Charles and Papailiopoulos, 2018; Frei and Gu, 2021). The following Proposition 5 controls stability coefficient for such a setting, ensuring concentration at a rate $\mathcal{O}(1/(\mu\sqrt{n}))$. Proposition 5 requires the following technical assumption, first used in the context of stability analysis by Charles and Papailiopoulos (2018), which holds when the global minimizer is unique (this is a strong assumption and serves as an idealization; note that an increasing ℓ_2 regularization pushes objective towards having a unique global minimum):

Assumption 2 *Let $\widehat{\theta}^{(i)}$ be a local minimizer of regularized empirical risk where ith covariate X_i is replaced by its independent copy. Assume that $\mathrm{Proj}(\widehat{\theta}^{(i)},\Theta_*)=\widehat{\theta}$ where $\mathrm{Proj}(\theta,\Theta_*)$ is a projection of θ onto the solution set $\Theta_*=\arg\min_{\theta\in\mathbb{R}^p}L_\lambda(\theta)$.*

Proposition 5 *Assume that L_λ satisfies μ-PL condition and that its gradient is Lipschitz (the objective is smooth). Assume that $\theta\mapsto f(x;\theta)$ is bounded by C and B-Lipschitz for all x. Recall that $Y_i=f^\star(X_i)+Z_i$ with $Z_i\sim\mathcal{N}(0,\sigma^2)$ and assume that $\|f^\star\|_\infty\leq 1$. Then, under Assumption 2*

$$\beta\leq\frac{2(1+C+\sigma)B^2}{\mu n}\ .$$

The proof is given in Appendix C.4.

Beyond Gaussian noise? Theorem 2 extends immediately from isotropic Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ to anisotropic $\mathcal{N}(0, \Sigma)$: following the proof, the weighting matrix becomes

$$M(\theta) = \frac{1}{n} \left(\nabla^2 L(\theta) + \lambda I\right)^{-1} \Phi \Sigma \Phi^\top \left(\nabla^2 L(\theta) + \lambda I\right)^{-1}$$

where $\Phi = (\nabla_\theta f(x_1; \theta), \ldots, \nabla_\theta f(x_n; \theta)) \in \mathbb{R}^{p \times n}$. A special case of this is a continuous heteroskedastic models of the form

$$Y_i = f(x_i; \theta^\star) + Z_i \qquad \text{where} \qquad Z_i \sim \mathcal{N}(0, \sigma^2(x_i)) \ .$$

Extending beyond Gaussianity faces two independent obstacles. The first concerns the concentration inequality. Our exponential-rate bound rests on Pisier's inequality (Lemma 8), which for a differentiable $F : \mathbb{R}^n \to \mathbb{R}$ and $Z \sim \mathcal{N}(0, I_n)$ bounds exponential moment $\mathbb{E} \exp(\lambda |F(Z) - \mathbb{E}[F(Z)]|)$ in terms of $\mathbb{E} \exp\left(c\lambda^2 \|\nabla F(Z)\|^2\right)$ for a universal constant $c > 0$. The critical feature is that $\|\nabla F(Z)\|^2$ on the right-hand side is *random*: in our setting, as shown in Lemma 7, $\|\nabla F(Z)\|^2 = (1/n)\|\nabla_\theta f(x; \widehat{\theta})\|^2_{M(\widehat{\theta})}$ varies with the noise realization. To our knowledge, no analogue of this inequality exists for general sub-Gaussian distributions in this form. The closest available results follow from the entropy method in the context of concentration inequalities (Ledoux, 2006; Boucheron et al., 2013): for distributions satisfying a log-Sobolev inequality with constant C_{LS}, one obtains $\ln \mathbb{E}[\exp(\lambda(F - \mathbb{E}\, F))] \le (C_{\mathrm{LS}}/2)\, \lambda^2 B^2$ when F is Λ-Lipschitz, i.e., $\|\nabla F\| \le B$ *deterministically*. This does not cover our setting, where $\|\nabla F(S)\|^2$ is itself a random quantity.

As a natural extension of our inequality one might consider generalized linear models with *discrete* responses, such as logistic regression ($Y_i \in \{-1, 1\}$). The proof of Lemma 7 differentiates the local minimizer $\widehat{\theta}$ with respect to a *continuous* scalar noise variable Z_i by employing the implicit function theorem. While there exists a concentration inequality of a form of Lemma 8 for $\{-1, +1\}$-valued random variables (Ivanisvili et al., 2020), the chain-rule argument breaks down, and such a result is not immediately applicable to give an analogue of Theorem 2.

3.3 Efficient Computation of the Weighted Norm

The calculation of the weighted norm involves the matrix $M(\widehat{\theta})$ which in turn involves a $p \times p$ inverse Hessian matrix of the loss function L_λ, which presents significant computational challenges since p is often large. Explicitly forming the Hessian has a computational cost $\mathcal{O}(p^2 \cdot C_L)$ (where C_L is the cost of evaluating L_λ), storing it requires $\mathcal{O}(p^2)$ memory, and inverting it takes $\mathcal{O}(p^3)$ operations, all of which are often infeasible. To this end, we consider a well-known and efficient alternative that leverages the Conjugate Gradient method (CG) which can be used to iteratively solve the linear system $Hh = v$ for $h = H^{-1}v$ (Pearlmutter, 1994). CG avoids forming H, requiring only the computation of Hessian-vector products (HVPs). An HVP can be computed efficiently using automatic differentiation without forming H explicitly, and is easily supported by modern packages such as JAX and PyTorch. These packages compute an HVP at a cost proportional to that of a single gradient evaluation, thereby circumventing the $\mathcal{O}(p^2)$ bottleneck. While these techniques are well known, we adapt them to our case and include pseudocode for completeness in Appendix C.5.

In our case, for CG to converge we need $k = \Omega(\sqrt{\kappa(H)} \ln(1/\varepsilon))$ iterations, where $\kappa(H) = \lambda_{\max}(H)/\lambda_{\min}(H)$ is the condition number of $H = \nabla^2 L(\widehat{\theta}) + \lambda I$, and ε is the desired

Table 1: Behaviour of CG when computing weighted norm for various scenarios of Hessian spectrum. Here we abbreviate $H = \nabla^2 L_\lambda(\widehat{\theta})$ and $g = \nabla_\theta f(x_{\text{test}}; \widehat{\theta})$ where we assume that $\widehat{\theta}$ is a local minimizer of L_λ. Let P_0 denote the orthogonal projector onto $\text{Ker}(H)$, and let $\cdot^\dagger$ stand for pseudo-inverse. In Appendix C.6 we further explain each case in detail.

Situation	**CG behavior**
$H \succ 0$	Converges normally.
$H \succeq 0$, singular, $g \in \text{Range}(H)$	Converges to $H^\dagger g$ when initialized at zero.
$H \succeq 0$, singular, $g \notin \text{Range}(H)$	No exact solution; convergence stagnates with residual at least $\|P_0 g\|$.
H nearly singular	May converge very slowly; weighted norm can be huge.

precision. In the interpolation regime, the residual Hessian vanishes and $\kappa(H) \leq 1 + B_g^2/\lambda$ where $B_g = \max_i \|\nabla_\theta f(x_i; \widehat{\theta})\|$; outside this regime the condition number may be larger (see Table 1 for the nearly singular case). We include a pseudocode for the Hessian-inverse-vector product in Algorithm 2 in Appendix C.5, while its computational complexity is of the order

$$\mathcal{O}\left((C_L + p)\sqrt{\kappa(H)}\ln\left(\frac{1}{\varepsilon}\right)\right)$$

which is a stark difference compared to $\mathcal{O}(p^3)$. In addition, in Table 1 we discuss behaviour of CG when the Hessian matrix may be singular. In particular, in pathological cases—such as when the Hessian is singular and the gradient vector lies in its nullspace—the behaviour of CG can serve as a diagnostic for identifying cases in which the weighted norm cannot be computed reliably.

Finally, the main algorithm that computes the weighted norm (Algorithm 1) calls a Hessian-inverse-vector product operation once. Then, the total computational complexity of the main algorithm in the worst case is

$$\tilde{\mathcal{O}}\left((n + p)\sqrt{\kappa(H)} + np\right)$$

assuming that the cost of loss evaluation is $C_L = \mathcal{O}(n)$. Note that in the interpolation regime when $\lambda = 0$, the computational complexity of Algorithm 1 is only $\tilde{\mathcal{O}}((C_L + p)/\sqrt{\lambda_{\min}} + p)$ where $\lambda_{\min}$ is the smallest positive eigenvalue of the Hessian matrix.

After examining the proof of Theorem 2 (see Section 4 or Section 3 for the sketch), one can argue that it might be easier to compute $\|\nabla_{(z_1,\ldots,z_n)} f(x; \widehat{\theta})\|^2$ which is identical to the weighted norm. While $\nabla_{z_i} f(x; \widehat{\theta}) = \nabla_\theta f(x; \widehat{\theta})^\top (\partial\widehat{\theta}/\partial z_i)$, we might be tempted to use automatic differentiation to directly compute $(\partial\widehat{\theta}/\partial z_i)$. If we use plain GD and terminate the gradient updates after τ rounds (that is, to approximate $\widehat{\theta} = \theta_\tau$), this is indeed possible by taking the chain rule: $(\partial\theta_\tau/\partial z_i) = (\partial\theta_\tau/\partial\theta_{\tau-1}) \ldots (\partial\theta_1/\partial z_i)$ and boils down to τ Jacobian-vector product operations. This naive approach would require $\mathcal{O}(\tau\,(C_L + p))$ computation and $\mathcal{O}(\tau\, C_L)$ memory (assuming reverse-mode differentiation through the entire trajectory; forward-mode Jacobian-vector products would instead require only $\mathcal{O}(C_L + p)$ memory

per step, but must be applied once per noise coordinate z_i). Thus, if we *save all* vectors $(\partial\widehat{\theta}/\partial z_i)_{i=1}^n$ in memory, we can calculate the desired $\|\nabla_{(z_1,\ldots,z_n)} f(x;\widehat{\theta})\|^2$ term in $\mathcal{O}(np)$ time and memory. Instead, if we employ an adaptive algorithm such as Adam, this would require to store *the entire* parameter vector trajectory. Moreover, this method is not applicable when all we have is access to the final parameter vector but not to the intermediate iterations.

4. Proof of Theorem 2

In this section we first show a general result which holds beyond the regression model of Eq. (2) and the square loss. The following Theorem 6 holds for any differentiable loss function as long as it is differentiable in the noise variable. For the sake of generality, we will introduce loss functions $\ell_i : \mathbb{R}^p \times \mathcal{Z} \to \mathbb{R}$ for $i \in [n]$ where $\mathcal{Z}$ is a real field (this way we associate each ℓ_i with a fixed input x_i). Then, slightly abusing the notation, the empirical risk is now defined as

$$L(\theta) = \frac{1}{n}\sum_{i=1}^n \ell_i(\theta, Z_i) \qquad \text{and} \qquad L_\lambda(\theta) = L(\theta) + \frac{\lambda}{2}\|\theta\|^2 \ .$$

For such losses we make the following assumption:

Assumption 3 *Assume that the losses are twice continuously differentiable in θ, and that $\frac{\mathrm{d}}{\mathrm{d}z}\nabla_\theta \ell_i(\theta, z)$ exists and is continuous.*

Consequently, in this section we generalize $M(\theta)$ matrix from Section 3 to arbitrary losses:

$$M(\theta) = \left(\nabla^2 L(\theta) + \lambda I\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^n G_i(\theta) G_i(\theta)^\top\right) \left(\nabla^2 L(\theta) + \lambda I\right)^{-1} \ ,$$

$$\text{where} \qquad G_i(\theta) = \frac{\mathrm{d}}{\mathrm{d}z}\nabla_\theta \ell_i(\theta, z)\Big|_{z=Z_i} \ .$$

Theorem 6 *Let Assumption 3 hold and let $\widehat{\theta}$ be any stationary point of L_λ such that $\nabla^2 L_\lambda(\widehat{\theta})$ is invertible, and assume that $\theta \mapsto f(x;\theta)$ is differentiable at $\widehat{\theta}$. Let $\|\nabla_\theta f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}$ be a sub-gamma random variable with variance factor v^2 and scale c. Then, for any fixed test input x and for any $\delta \in (0,1)$,*

$$\mathbb{P}\left\{|\Delta(x;\widehat{\theta})| > \sqrt{\frac{\pi^2 \ln(2/\delta)}{n}\,\mathbb{E}\left[\|\nabla f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}\right]} + \frac{6\ln(2/\delta)^{3/4}\sqrt{v} + 9\ln(2/\delta)\sqrt{c}}{\sqrt{n}}\right\} \le \delta \ .$$

Then, Theorem 2 is a consequence of the above. Namely, specializing Theorem 6 to $\ell_i(\theta, Z_i) = (1/2)(f(x_i;\theta) - Y_i)^2 = (1/2)(f(x_i;\theta) - f^\star(x_i) - \sigma Z_i)^2$,

$$G_i(\theta) = \frac{\mathrm{d}}{\mathrm{d}z}\nabla_\theta \ell_i(\widehat{\theta}, z)\,|_{z=Z_i} = -\sigma\nabla_\theta f(x_i;\widehat{\theta}) \ .$$

Consequently, the $M(\widehat{\theta})$ matrix in Theorem 6 acquires a factor σ^2 compared to the matrix $M(\widehat{\theta})$ defined in Section 3, which accounts for the explicit σ factor appearing in Theorem 2. Note that the sub-gamma parameters (v^2, c) in Theorem 2 refer to the weighted norm using the $M(\widehat{\theta})$ from Section 3 (without the σ^2 factor).

4.1 Sensitivity of Predictor to Noise in the Response at a Stationary Point

The following key lemma is used in the proof of Theorem 6. The lemma can be seen as an Implicit Function Theorem, or simply differentiating in the noise variable and applying the chain rule.

Lemma 7 (Sensitivity lemma) *Let Assumption 3 hold. Let $\widehat{\theta}(z)$ be a stationary point of L_λ defined w.r.t. $z \in \mathbb{R}^n$ (each Z_i is replaced by z_i). Assuming that $\nabla^2 L(\widehat{\theta}(z)) + \lambda I$ is invertible, for any function $F : \mathbb{R}^p \to \mathbb{R}$ differentiable at $\widehat{\theta}(z)$ we have*

$$\|\nabla_z F(\widehat{\theta}(z))\|^2 = \frac{1}{n}\|\nabla_\theta F(\widehat{\theta}(z))\|^2_{M(\widehat{\theta}(z))} \ .$$

Proof By assumption $\widehat{\theta}(z)$ is a stationary point, namely

$$\lambda\widehat{\theta}(z) = -\nabla_\theta L(\widehat{\theta}(z)) \ .$$

We first need to establish that $\widehat{\theta}(z)$ is differentiable in z. Introduce

$$\Phi(\theta, z) = \nabla_\theta L(\theta) + \lambda\theta,$$

where the dependence of L on z is implicit. By stationarity, $\Phi(\widehat{\theta}(z), z) = 0$ and moreover $\nabla_\theta\Phi(\widehat{\theta}(z), z)$ is invertible. Hence, by the implicit function theorem, there exists a neighborhood of z on which $\widehat{\theta}(z)$ is a differentiable local branch of stationary points. Thus, we can differentiate $\widehat{\theta}$ in z. Now, observe that by the chain rule

$$\frac{\mathrm{d}}{\mathrm{d}z_i}F(\widehat{\theta}(z)) = \left(\frac{\mathrm{d}\widehat{\theta}(z)}{\mathrm{d}z_i}\right)^\top \nabla_\theta F(\widehat{\theta}(z)), \tag{6}$$

where $(\mathrm{d}\widehat{\theta}(z)/\,\mathrm{d}z_i) \in \mathbb{R}^p$ and $\nabla_\theta F(\widehat{\theta}(z)) \in \mathbb{R}^p$.

The rest of the proof deals with giving a closed-form expression for $(\mathrm{d}\widehat{\theta}(z)/\,\mathrm{d}z_i)$. By stationarity we have

$$\nabla_\theta L(\widehat{\theta}(z)) + \lambda\widehat{\theta}(z) = 0$$

and so differentiating w.r.t. z_i gives

$$\left(\nabla^2_\theta L(\widehat{\theta}(z)) + \lambda I\right)\frac{\mathrm{d}\widehat{\theta}(z)}{\mathrm{d}z_i} + \frac{1}{n}\left(\frac{\mathrm{d}}{\mathrm{d}z'}\nabla_\theta \ell_i(\widehat{\theta}(z), z')\Big|_{z'=z_i}\right) = 0.$$

Rearranging the above and treating it as a linear system, the solution is

$$\begin{aligned}\frac{\mathrm{d}\widehat{\theta}(z)}{\mathrm{d}z_i} &= -\frac{1}{n}\left(\nabla^2_\theta L(\widehat{\theta}(z)) + \lambda I\right)^{-1}\left(\frac{\mathrm{d}}{\mathrm{d}z'}\nabla_\theta \ell_i(\widehat{\theta}(z), z')\Big|_{z'=z_i}\right) \\ &= -\frac{1}{n}\left(\nabla^2 L(\widehat{\theta}(z)) + \lambda I\right)^{-1} G_i(z)\end{aligned}$$

while

$$\frac{\mathrm{d}\widehat{\theta}(z)}{\mathrm{d}z} = -\frac{1}{n}\left(\nabla^2_\theta L(\widehat{\theta}(z)) + \lambda I\right)^{-1} (G_1(z), \ldots, G_n(z)) \ .$$

Plugging this expression into Eq. (6)

$$\nabla_z F(\widehat{\theta}(z))^\top = -\frac{1}{n}\nabla_\theta F(\widehat{\theta}(z))^\top \left(\nabla^2_\theta L(\widehat{\theta}(z)) + \lambda I\right)^{-1} (G_1(z), \ldots, G_n(z)) \ .$$

and taking $\|\cdot\|^2$ on both sides gives the statement. ■

4.2 Proof of Theorem 6

The proof is based on the following bound on the exponential moment due to Pisier, which tells that a differentiable function of a Gaussian vector concentrates around its mean well if its gradient is well-behaved (Wainwright, 2019, Display below Lemma 2.27), (Boucheron et al., 2013, Exercise 5.16):

Lemma 8 (Pisier (1986)) *Let $Z = (Z_1, \ldots, Z_n) \sim \mathcal{N}(0, I_n)$. Then, for a differentiable function $F : \mathbb{R}^n \to \mathbb{R}$ and any $\lambda \in \mathbb{R}$ we have*

$$\mathbb{E}\exp\left(\lambda(F(Z) - \mathbb{E}[F(Z)])\right) \le \mathbb{E}\exp\left(\frac{\lambda^2\pi^2}{8}\|\nabla F(Z)\|^2\right) \ .$$

Lemma 9 *Let $B \ge 0$, $v \ge 0$, $c > 0$, and $t \ge 0$. Define*

$$\phi(t) = \sup_{\lambda \in [0, 1/\sqrt{c})} \left\{\lambda t - \lambda^2 B - \frac{\lambda^4 (v^2/2)}{1 - c\lambda^2}\right\}.$$

Then

$$\phi(t) \ge \frac{1}{8}\min\left\{\frac{t^2}{B}, \frac{t^{4/3}}{v^{2/3}}, \frac{t}{\sqrt{c}}\right\} \ .$$

The proof of Lemma 9 is given in Appendix D.

Combining the above two results, we have the following concentration inequality:

Lemma 10 *Let $Z = (Z_1, \ldots, Z_n) \sim \mathcal{N}(0, I_n)$. Assume that $F : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function. Then, assuming that $\|\nabla F(Z)\|^2$ is a sub-gamma random variable with variance factor v^2 and scale c, for any $\delta \in (0,1)$ we have,*

$$\mathbb{P}\Bigg\{ |F(Z) - \mathbb{E}[F(Z)]| \\ > \frac{\pi}{\sqrt{8}}\max\left\{\sqrt{8\,\mathbb{E}[\|\nabla F(Z)\|^2]\ln(2/\delta)}, (8\ln(2/\delta))^{3/4}\sqrt{v}, 8\ln(2/\delta)\sqrt{c}\right\}\Bigg\} \le \delta$$

Proof Let[3]

$$\psi(\lambda) = \ln \mathbb{E} \exp\left(\lambda \left(\sqrt{8}/\pi\right) \left(F(Z) - \mathbb{E}[F(Z)]\right)\right) \qquad (\lambda > 0) \ .$$

By Lemma 8

$$\begin{aligned}
\psi(\lambda) &\leq \ln \mathbb{E} \exp\left(\lambda^2 \|\nabla F(Z)\|^2\right) \\
&= \lambda^2 \, \mathbb{E}[\|\nabla F(Z)\|^2] + \ln \mathbb{E} \exp\left(\lambda^2 \left(\|\nabla F(Z)\|^2 - \mathbb{E}[\|\nabla F(Z)\|^2]\right)\right) \\
&\leq \lambda^2 \, \mathbb{E}[\|\nabla F(Z)\|^2] + \frac{\lambda^4 (v^2/2)}{1 - c\lambda^2} \ . \qquad \text{(Sub-gamma assumption, valid for } \lambda < 1/\sqrt{c})
\end{aligned}$$

Using the Chernoff's method for any $t, \lambda > 0$

$$\mathbb{P}\left\{\left(\sqrt{8}/\pi\right) \left(F(Z) - \mathbb{E}[F(Z)]\right) > t\right\} \leq \mathbb{E} \exp\left(-\left(\lambda t - \psi(\lambda)\right)\right)$$

while focusing on the term in exp() on the right-hand side,

$$\lambda t - \psi(\lambda) \geq \lambda t - \lambda^2 \, \mathbb{E}[\|\nabla F(Z)\|^2] - \frac{\lambda^4 (v^2/2)}{1 - c\lambda^2}$$

and using Lemma 9 we get

$$\mathbb{P}\left\{\left(\sqrt{8}/\pi\right) \left(F(Z) - \mathbb{E}[F(Z)]\right) > t\right\} \leq \exp\left(-\frac{1}{8} \min\left\{\frac{t^2}{\mathbb{E}[\|\nabla F(Z)\|^2]}, \frac{t^{4/3}}{v^{2/3}}, \frac{t}{\sqrt{c}}\right\}\right)$$

while inverting this bound and using symmetry to get bound on both tails yields the statement. ■

Proof [Completing the proof of Theorem 6] Abbreviate $f(\theta) = f(x; \theta)$. We will apply Lemma 10 with $F(Z) = f(\widehat{\theta})$, and first observe that

$$\nabla F(Z) = \nabla_z f(\theta(Z))$$

and so by Lemma 7

$$\|\nabla F(Z)\|^2 = \frac{1}{n} \|\nabla f(\widehat{\theta}(Z))\|^2_{M(\widehat{\theta}(Z))}$$

while taking expectation on both sides

$$\mathbb{E} \, \|\nabla F(Z)\|^2 = \frac{1}{n} \, \mathbb{E} \, \|\nabla f(\widehat{\theta})\|^2_{M(\widehat{\theta})} \ .$$

By assumption $\|\nabla f(\widehat{\theta})\|^2_{M(\widehat{\theta})}$ is a sub-gamma random variable with variance factor v^2 and scale c, and so $\|\nabla F(Z)\|^2$ is a sub-gamma random variable with $(v^2/n^2, c/n)$. Then, Lemma 10 gives the statement. ■

3. Throughout this proof λ denotes the Chernoff (variational) parameter, not the regularization parameter used elsewhere in the paper.

5. Experiments

This section details the experimental setup used to evaluate the proposed pointwise confidence estimation method (Theorem 2 where weighted norm is computed by Algorithm 1) against a standard baseline in a controlled regression setting with distributional shift. We aim to verify the hypothesis that Theorem 2 results in confidence intervals with good coverage of a non-linear ground-truth function, which are still narrow in the support of the training data. In particular, we are interested in its behaviour in regions with a lack of the training data. We will start by looking at a one-dimensional regression example, where we train a neural network to fit the data. Despite its simplicity, this setting is meaningful because the sensitivity of the confidence bound to the test point (as captured by the weighted norm) is measured in a high-dimensional implicit feature space (in other words, the neural network first projects a scalar input into a high-dimensional feature space). Then, we will look at the performance with inputs of a higher dimension.

Data We consider regression problems of various input dimensions. Inputs are drawn from a standard normal distribution, $\mathcal{N}(0, I_d)$. The corresponding output Y is generated according to the model Eq. (2), where the ground-truth function $f(x; \theta^\star)$ is a function sampled (and fixed throughout) from a Matérn Reproducing kernel Hilbert space (RKHS) (Rasmussen and Williams, 2006), providing a smooth, non-linear target function. For all experiments, the length scale, output scale, and smoothness (ν) parameters are respectively $(1, 1, 1.5)$.[4] The standard deviation of the regression noise is $\sigma = 0.1$.

A key aspect of our experimental design introduces a distributional shift between training and evaluation data, which is inspired by the approach of Antorán et al. (2022). Once the training and validation sets are drawn as described above, we remove all examples whose inputs lie within a box $[-0.5, 0.5]^d$. The validation set used for hyperparameter search is sampled in the same way as the training set. Meanwhile, the test inputs are either drawn from the original Gaussian distribution, without removing points (for $d > 1$) or form a uniform grid (for $d = 1$). This way we are able to test sensitivity of the confidence bound to the region of the input space where no observations were made.

Model and training We employ a fully connected neural network with 2 hidden layers (1024 and 512 neurons respectively) and the `ReLU` activation function in the case of the $d = 1$ experiments, and a neural network with 1024-sized hidden layer for $d > 1$.

For $d = 1$ we use variance scaling with standard deviation 100 (this results in a less smooth predictor), while for $d > 1$ we initialize parameters using the Glorot scheme and train a neural network by minimizing the average square loss with ℓ_2 regularization (L_λ). The optimization is performed using the AdamW optimizer (Loshchilov and Hutter, 2019) with parameters with a step size $\eta = 10^{-3}$, and weight decay $\lambda = 10^{-8}$ for $d = 1$, $\lambda = 10^{-3}$ for $d = 10$, and $\lambda = 10^{-4}$ for $d = 100$. We used a cosine learning rate schedule over the course of training. The model is trained for a total of 10^4 passes over the training data (without mini-batching). All experiments are performed on a platform with an Nvidia P100 GPU.

The hyperparameter λ is tuned by minimizing the mean Winkler interval score (see below) on the validation set. Experiments in Tables 2 and 3 are performed on 5 independent

4. For experiments with $d > 1$ we set output scale to 0.1.

trials with respect to draws of the data, while the tables report metric averages or medians with standard deviations over these trials.

Confidence estimation We evaluate pointwise confidence intervals generated by our proposed method. As a baseline for comparison, we consider the standard non-parametric bootstrap method (Efron and Tibshirani, 1994). For the bootstrap, we generated 10 bootstrap replicates of the training dataset sampled with replacement, trained a separate network on each replicate using the same architecture and training procedure, and derived empirical confidence intervals, using the quantile method with $\alpha = 0.01$, from the resulting ensemble of predictions at each test point. Confidence intervals constructed from the bootstrap are a good baseline as many other methods based on ensembles can be viewed as a proxy to it (Lakshminarayanan et al., 2017; Gal and Ghahramani, 2016; Osband et al., 2023). The 'proposed method' is a bound in Theorem 2 with $v = c = 1$ and $\delta = 0.01/\#$ test points (where the latter division accounts for a union bound over test points). In particular, to compute the weighted norm $\|\nabla_\theta f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}$ we used Algorithm 1 with 10^3 CG iterations and tolerance 10^{-12}.

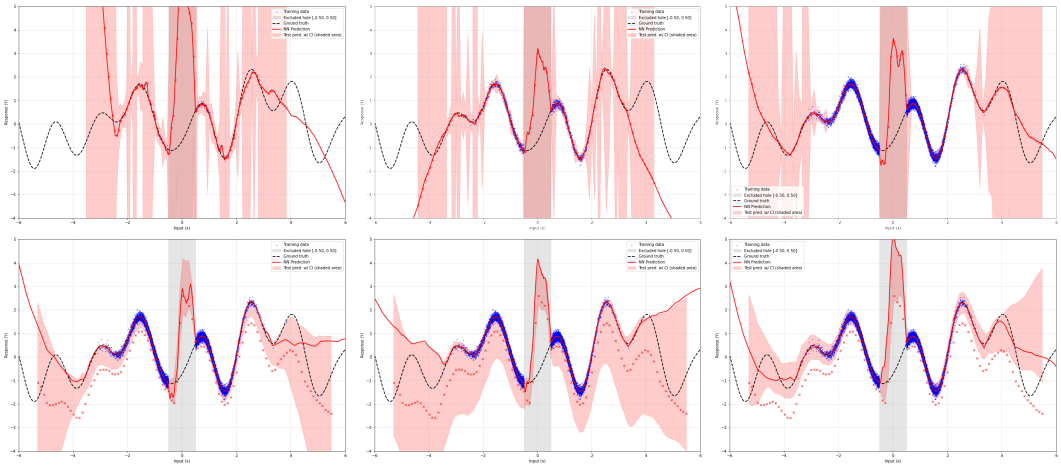
Evaluation The primary goal is to assess the quality of the pointwise confidence intervals, particularly within the unobserved regions. For the $d = 1$ case, we visualize the results in Figure 1. In this case, we provide plots for various sample sizes. The expectation is that a reliable estimate should yield wider CIs in regions without observations compared to regions well-covered by training data.

In problems with $d > 1$, we rely on metrics such as coverage, width, and their combination as a single metric known as the mean Winkler interval score (Winkler, 1972). Here, coverage is computed as an average of indicators $\mathbf{1}\{\textsc{lb}(x) \le f(x;\theta^\star) \le \textsc{ub}(x)\}$ over all test inputs x where $\textsc{ub}()$ and $\textsc{lb}()$ are upper and lower bounds. The width is computed as an average of widths $\textsc{ub}(x_{\text{nn}}(x)) - \textsc{lb}(x_{\text{nn}}(x))$ over all test inputs x, where $x_{\text{nn}}(x)$ is a nearest neighbour of x among all training inputs (but after filtering out outliers).[5] This ensures that we compute the width only in the regions where we have sufficiently many observations. Finally, the mean Winkler interval score is defined as $W_\alpha = (1/n)\sum_i[(\textsc{ub}(x_i) - \textsc{lb}(x_i)) + (2/\alpha)\max(0, Y_i - \textsc{ub}(x_i)) + (2/\alpha)\max(0, \textsc{lb}(x_i) - Y_i)]$ where α is the nominal miscoverage rate. Clearly, W_α is small when we have a good coverage, yet intervals are narrow, and it can be computed on any sample, which makes it useful for hyperparameter selection.

Discussion First, focusing on Figure 1, we indeed observe that the proposed method yields wide intervals in regions where few to none observations were made (see left and right sides of the graph as well as the central 'cut-out' part), thus confirming our hypothesis on weighted norms. At the same time, intervals provide an adequate coverage where many observations are available, despite the fact that we do not control the bias term. Second, confidence intervals seem to narrow with increase of the sample size, which is an expected behavior.

In contrast, as we can see in the second row of Figure 1, bootstrap fails to produce a good confidence interval on this task even when n is large. This is a common problem with bootstrap, which might yield overconfident intervals; bias-corrected or studentized variants

5. Let ε be a 99th percentile of $\{\min_j \|x_i - x_j^{\text{test}}\| \ : \ i \in [n]\}$. Then, evaluation is done on all inputs indexed by $\{\arg\min_j \|x_i - x_j^{\text{test}}\| \ : \ i \in [n] \ , \ \min_j \|x_i - x_j^{\text{test}}\| \le \varepsilon\}$.

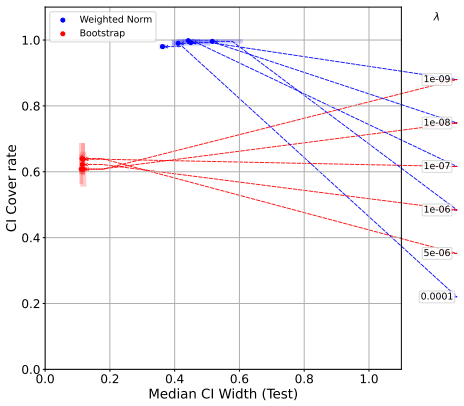


$n = 100$ $n = 1000$ $n = 10000$

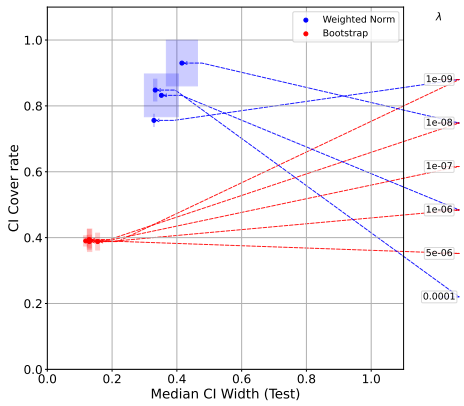
Figure 1: The first row: proposed confidence interval evaluation with increasing sample size. The second row: bootstrap confidence interval evaluation. In this case confidence intervals are formed from quantiles over predictions from 10 models trained on draws from the training sample with replacement.

might improve coverage but are computationally prohibitive. Looking at higher dimensional cases in Tables 2 and 3 and Figure 2, we note that the median CI width is generally narrowing with an increase in the sample size, while maintaining a good coverage of the ground truth. In contrast, similarly to the $d = 1$ case, bootstrap confidence intervals again demonstrate over-confidence, which results in a substantially reduced coverage (well below the target $1 - \delta$). Note that in these tables λ is fixed across all sample sizes; the optimal λ generally depends on n, so occasional non-monotone behaviour in CI width across n can arise and would be resolved by tuning λ individually for each sample size. Finally, Figure 2 is intended to gauge the sensitivity of the proposed method to the choice of the regularization parameter $\lambda \in \{10^{-6}, 10^{-5}, \ldots, 10^{-2}\}$. This figure also captures the trade-off between coverage and CI width, with the best outcome being the top-left corner. It is evident that λ controls this trade-off and can be tuned in practice to obtain the trade-off as close as possible to the top-left corner. On the other hand, bootstrap appears to be insensitive to various choices of λ possibly because the underlying neural network bootstrap replications are not much affected by λ in terms of their smoothness.

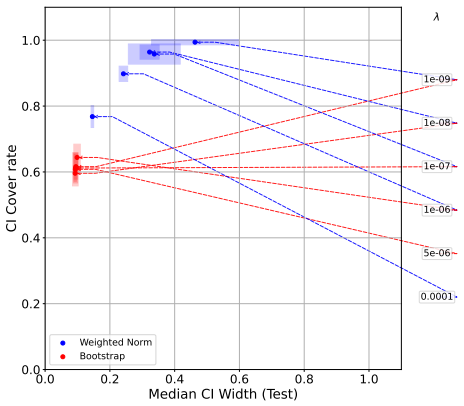
Wallclock timings We report wallclock training and CI computation times (in seconds, on an Nvidia P100 GPU) in Tables 2 to 5. For bootstrap, the reported Train Time corresponds to training a *single* model, not the full ensemble; the cost of training all bootstrap replications is absorbed into the CI Time. This explains why bootstrap CI Times are substantially larger than those of the proposed method. Moreover, the CI Time for the proposed (weighted norm) method grows with the number of training points n, whereas the bootstrap CI Time remains roughly constant in n. This is a direct consequence of the HVP-based algorithm described in Section 3.3: for each test point, the weighted norm is evaluated with respect to *all* training points via Hessian–vector products, so the cost scales with n. An alternative of



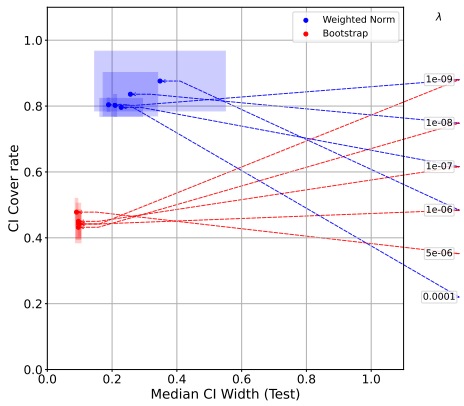
$n = 100,\ d = 10$



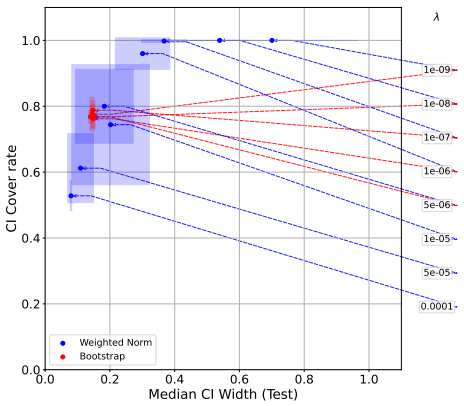
$n = 100,\ d = 100$



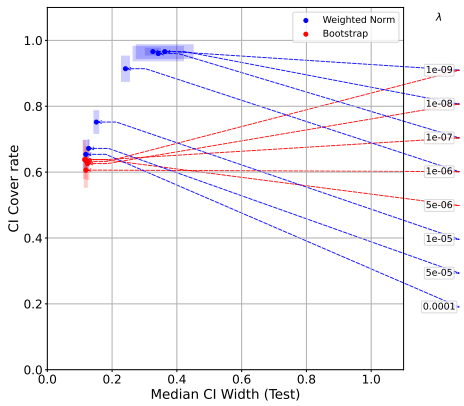
$n = 1000,\ d = 10$



$n = 1000,\ d = 100$



$n = 10000,\ d = 10$



$n = 10000,\ d = 100$

Figure 2: Trade-off between CI width and coverage, when varying regularization parameter $\lambda \in \{10^{-6}, 10^{-5}, \ldots, 10^{-2}\}$. The best outcome is the top-left corner. Each figure captures median CI width on the test set (horizontal axis) against coverage rate (vertical axis). Each point corresponds to evaluation given a specific regularization parameter λ. In particular, a point is an average of evaluations, while the shaded region represents standard deviations. Blue points stand for the proposed method while red points are bootstrap. Each row corresponds to a fixed sample size n, with $d = 10$ (left) and $d = 100$ (right).

Table 2: Proposed method vs bootstrap quantile CI on a 10-dimensional synthetic problem, $\delta = 0.01$, with $\lambda = 10^{-3}$ in all cases. Train Time and CI Time are wallclock times in seconds measured on an Nvidia P100 GPU.

	N Train	Test MSE	Avg CI Width	Median CI Width	Mean Interval Score ($W_{0.01}$)	CI Cover (%)	Train Time (s)	CI Time (s)
Proposed	100	0.0312 ± 0.0046	2.1295 ± 1.2149	0.5162 ± 0.0920	2.3052 ± 1.0058	$99.60\% \pm 0.49\%$	31.58 ± 1.38	1.55 ± 0.16
	1000	0.0246 ± 0.0054	0.4846 ± 0.0120	0.2418 ± 0.0131	2.9416 ± 1.1417	$89.80\% \pm 2.32\%$	32.95 ± 0.87	1.95 ± 0.10
	10000	0.0270 ± 0.0043	1.0030 ± 0.5659	0.3010 ± 0.0840	2.1816 ± 1.1668	$96.00\% \pm 4.77\%$	32.39 ± 0.28	28.62 ± 0.36
	20000	0.0233 ± 0.0049	0.6623 ± 0.3477	0.2333 ± 0.0891	3.5615 ± 2.3433	$89.60\% \pm 10.67\%$	33.09 ± 0.47	49.25 ± 9.22
Bootstrap	100	0.0265 ± 0.0045	0.2389 ± 0.0032	0.1153 ± 0.0051	10.0971 ± 1.7505	$62.20\% \pm 2.79\%$	88.61 ± 3.81	856.12 ± 41.79
	1000	0.0209 ± 0.0056	0.2004 ± 0.0142	0.0987 ± 0.0099	9.3941 ± 2.0575	$64.40\% \pm 4.03\%$	86.71 ± 0.45	867.48 ± 42.01
	10000	0.0198 ± 0.0029	0.3008 ± 0.0060	0.1489 ± 0.0055	6.2975 ± 1.3709	$76.20\% \pm 2.71\%$	87.10 ± 3.77	849.81 ± 38.75
	20000	0.0209 ± 0.0037	0.3006 ± 0.0085	0.1469 ± 0.0041	5.9437 ± 0.9310	$77.80\% \pm 1.94\%$	86.42 ± 1.37	841.43 ± 14.87

Table 3: Same setting as in Table 2 but on a 100-dimensional problem, with $\lambda = 10^{-4}$ in all cases. Train Time and CI Time are wallclock times in seconds measured on an Nvidia P100 GPU.

	N Train	Test MSE	Avg CI Width	Median CI Width	Mean Interval Score ($W_{0.01}$)	CI Cover (%)	Train Time (s)	CI Time (s)
Proposed	100	0.0734 ± 0.0071	3.4774 ± 4.5786	0.7123 ± 0.7343	8.2162 ± 2.3118	$84.40\% \pm 9.52\%$	32.08 ± 0.45	2.36 ± 0.07
	1000	0.0406 ± 0.0053	0.6456 ± 0.1690	0.2281 ± 0.0658	7.6975 ± 1.2885	$79.60\% \pm 2.73\%$	32.03 ± 0.94	4.64 ± 0.06
	10000	0.0279 ± 0.0027	0.7453 ± 0.3018	0.3425 ± 0.0777	1.9889 ± 0.2441	$96.00\% \pm 2.19\%$	31.95 ± 0.59	40.39 ± 0.92
	20000	0.0268 ± 0.0035	0.6712 ± 0.1748	0.3042 ± 0.0083	1.7652 ± 0.3218	$94.40\% \pm 1.36\%$	32.16 ± 0.48	76.62 ± 0.24
Bootstrap	100	0.0763 ± 0.0073	0.2835 ± 0.0089	0.1307 ± 0.0067	23.5876 ± 2.4808	$39.20\% \pm 3.43\%$	84.59 ± 0.53	827.78 ± 8.65
	1000	0.0406 ± 0.0052	0.2128 ± 0.0155	0.0960 ± 0.0069	16.5697 ± 1.4079	$45.00\% \pm 5.51\%$	85.47 ± 1.02	831.47 ± 4.48
	10000	0.0249 ± 0.0038	0.2433 ± 0.0040	0.1187 ± 0.0027	9.4523 ± 1.7771	$63.80\% \pm 3.87\%$	85.63 ± 1.24	829.27 ± 12.77
	20000	0.0224 ± 0.0022	0.2361 ± 0.0071	0.1147 ± 0.0065	8.6572 ± 1.1176	$64.00\% \pm 3.35\%$	85.11 ± 0.93	833.17 ± 10.24

explicitly inverting the Hessian once would yield a fixed CI time, but at a prohibitive cost in memory and in the time required for the inversion itself.

Real data We also perform evaluations on the real datasets. Note that such evaluations are artificial in some sense, because we do not have access to the ground truth function $f^\star$, yet our goal is to evaluate its coverage (we only have access to responses of a form $Y_i = f^\star(x_i) + \varepsilon_i$, that is, *with additive noise*). Here we mimic such evaluation by fitting a non-linear regressor on the entire dataset in the hope that regression estimation is consistent.

The following is an evaluation on the California housing dataset (Pedregosa et al., 2011), Table 4. The dataset is standardized and the 'hole' extraction is applied as described in the experimental section to simulate the missing data. Then, we fit histogram gradient boosted regressor (`sklearn.ensemble.HistGradientBoostingRegressor`) on the entire sample to simulate the regression function. The rest of the evaluation matches the protocol as in the synthetic case. In addition, we run experiments on a higher-dimensional real dataset, Table 5. We used the QSAR-TID-11 (chemistry-related) dataset with 1024 covariates and approx. 5000 records which can be found in the OpenML repository (Vanschoren et al., 2013). Observe that the coverage trade-off message remains consistent with previous experiments. The same timing characteristics as in the synthetic case hold here: bootstrap Train Time is for a single model only, with the ensemble cost folded into CI Time, and the proposed

Table 4: Proposed method vs bootstrap quantile CI on the California housing dataset (Pedregosa et al., 2011) ($d = 8$), $\delta = 0.01$, with $\lambda = 10^{-6}$ in all cases. Dataset is standardized; a 'hole' is extracted to simulate missing data, and a histogram gradient boosted regressor is fitted on the full sample to proxy $f^\star$. Train Time and CI Time are wallclock times in seconds measured on an Nvidia P100 GPU.

	N Train	Test MSE	Median CI Width (Test)	CI Cover (%)	Train Time (s)	CI Time (s)
Proposed	100	1.6617 ± 1.9605	5.5511 ± 3.3646	$96.40\% \pm 3.72\%$	76.22 ± 1.15	3.78 ± 0.09
	1000	0.5797 ± 0.1844	16.4668 ± 4.1686	$100.00\% \pm 0.00\%$	75.20 ± 0.17	16.58 ± 0.23
	10000	0.3247 ± 0.0521	0.4677 ± 0.0615	$85.00\% \pm 4.47\%$	53.86 ± 2.00	59.65 ± 0.52
	20000	0.2569 ± 0.0674	0.2445 ± 0.0139	$74.20\% \pm 2.71\%$	51.38 ± 2.41	114.08 ± 0.79
Bootstrap	100	1.6622 ± 1.9614	0.5631 ± 0.0698	$65.60\% \pm 5.95\%$	87.83 ± 1.17	845.70 ± 6.10
	1000	0.6393 ± 0.2950	0.5565 ± 0.0280	$82.60\% \pm 4.59\%$	87.43 ± 1.75	842.59 ± 5.33
	10000	0.3205 ± 0.0523	0.3105 ± 0.0114	$78.80\% \pm 2.04\%$	87.72 ± 0.81	849.91 ± 9.63
	20000	0.2636 ± 0.0617	0.2382 ± 0.0075	$72.40\% \pm 3.14\%$	86.63 ± 1.82	846.59 ± 16.37

Table 5: Proposed method vs bootstrap quantile CI on the QSAR-TID-11 dataset (Vanschoren et al., 2013) ($d = 1024$), $\delta = 0.01$, with $\lambda = 5.5 \times 10^{-3}$ in all cases. The same evaluation protocol as in Table 4 is applied. Train Time and CI Time are wallclock times in seconds measured on an Nvidia P100 GPU.

	N Train	Test MSE	Median CI Width (Test)	CI Cover (%)	Train Time (s)	CI Time (s)
Proposed	100	0.1089 ± 0.0101	0.2991 ± 0.0019	$62.60\% \pm 3.50\%$	60.94 ± 7.50	3.47 ± 1.13
	2500	0.0280 ± 0.0058	0.1514 ± 0.0085	$70.80\% \pm 2.79\%$	83.13 ± 39.48	43.00 ± 10.46
	5000	0.0173 ± 0.0019	0.1154 ± 0.0034	$70.60\% \pm 3.67\%$	62.06 ± 4.31	62.88 ± 8.99
Bootstrap	100	3218.83 ± 198.56	12.39 ± 0.52	$15.80\% \pm 3.19\%$	87.75 ± 0.81	842.26 ± 10.13
	2500	625.71 ± 107.30	11.18 ± 0.74	$40.60\% \pm 3.93\%$	87.16 ± 1.03	833.89 ± 5.13
	5000	308.75 ± 61.58	9.20 ± 0.71	$43.00\% \pm 2.37\%$	86.92 ± 1.68	843.73 ± 4.13

method's CI Time grows with n due to the HVP-based algorithm (see the timing discussion above). We also note that in all real-data tables λ is held fixed across all sample sizes. This can give rise to occasional anomalies in CI width, such as the jump in median CI width for the proposed method between $n = 100$ and $n = 1000$ in Table 4 (from 5.55 to 16.47, followed by a sharp drop at $n = 10000$). Such behaviour occurs because a single λ cannot be simultaneously optimal for every sample size; tuning λ individually for each n would be expected to yield more uniformly well-behaved results.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments. The authors declare no competing interests and no external funding sources.

Appendix A. Additional Related Work

Confidence intervals from asymptotic normality of MLE The standard theory of parametric statistics establishes that MLE, $\widehat{\theta}_n$, are asymptotically normal (Van der Vaart, 2000). This pivotal result is often stated as $\sqrt{n}(\widehat{\theta}_n - \theta^\star) \to \mathcal{N}(0, \mathcal{I}(\theta^\star)^{-1})$ in distribution, where $\theta^\star$ is the true parameter value and $\mathcal{I}(\theta^\star)$ is the Fisher Information Matrix. One of the most prevalent confidence sets based on this result are *Wald-type confidence sets* that typically take an ellipsoidal form for a vector parameter θ:

$$\left\{\theta \in \Theta : n(\widehat{\theta}_n - \theta)^\top \widehat{\mathcal{I}}_n (\widehat{\theta}_n - \theta) \le \chi^2_{k,\alpha}\right\}$$

where $\chi^2_{k,\alpha}$ is the upper α-quantile of the chi-squared distribution with k degrees of freedom (the dimension of θ) and where $\widehat{\mathcal{I}}_n$ is an estimate of the Fisher Information Matrix $\mathcal{I}(\theta^\star)$.

Despite their simplicity, Wald-type confidence sets often exhibit poor finite sample performance, leading to unreliable coverage probabilities, and can produce intervals/sets that extend beyond valid parameter boundaries, particularly when the true parameter is near a boundary. Furthermore, they lack invariance to parameter transformations, which can lead to different inferential conclusions based on the chosen parameterization.

In this work we do not explicitly construct confidence sets for parameters, but instead give confidence intervals for the predictor on a given test point directly. However, our results are related to asymptotic normality of MLE, in a sense that the shape of the confidence interval is elliptical in nature as well, and is characterized by the weighting matrix that is different from $\mathcal{I}(\theta^\star)$ (but matches it asymptotically under certain conditions). One major distinction is that our result is non-asymptotic and so must enjoy better coverage in the presence of a small sample. In Section 3 we discuss two approaches more formally.

Bootstrap is a classical tool in statistics (Efron and Tibshirani, 1994), often treated as a gold-standard method for the construction of confidence sets when dealing with complex estimation problems where no other well-known alternatives are available. The construction of confidence sets using bootstrap is often done through quantiles. The fundamental principle involves approximating the unknown sampling distribution of an estimator, say $\widehat{\theta}_n$, with the empirical distribution of bootstrap replications. Confidence intervals are then formed by taking the appropriate quantiles from this bootstrap distribution. For instance, the percentile method, a direct quantile-based approach, uses the $\alpha/2$ and $1-\alpha/2$ quantiles of the bootstrap distribution of $\widehat{\theta}^*_n$ to form a $1-\alpha$ confidence interval. Standard textbooks such as those of Van der Vaart (2000), Shao (2003), and Wasserman (2004) establish the asymptotic validity of such methods, demonstrating that under regularity conditions, the bootstrap quantiles asymptotically mimic the quantiles of the true sampling distribution, leading to valid confidence sets. This method is particularly attractive because it is nonparametric and does not require explicit knowledge of the estimator's limiting distribution, making it broadly applicable even when analytic forms are intractable. On the other hand, the bootstrap may fail to provide accurate coverage in small samples or in cases where the estimator is not asymptotically normal, as the convergence of the bootstrap distribution to the true distribution can be slow or even inconsistent in such settings. Moreover, the bootstrap can be sensitive to the presence of *bias* and to the structure of the data, such as dependence or heavy tails, which may result in under- or overcoverage of the true parameter. In particular, the

percentile method (using direct quantiles of the bootstrap distribution) can be suboptimal. There are alternative approaches, such as bias-corrected or studentized intervals, which may yield improved performance. Another major limitation of bootstrap is the need to have a sufficiently many replications to reduce the variance, which is extremely prohibitive in many applications, especially involving neural networks with millions or billions of parameters.

Ensemble methods represent a family of techniques that draw upon the concept of bootstrap for estimating confidence. These methods involve training multiple predictors, each on a slightly altered version of the original data (for a comprehensive review on ensembling, see also Dwaracherla et al. 2023). Common approaches include using different subsamples of the dataset for each model (bootstrap) and perturbing the observations (Osband et al., 2016), or directly perturbing the model parameters (Lakshminarayanan et al., 2017; Gal and Ghahramani, 2016; Osband et al., 2018, 2023). The empirical distribution derived from the resulting estimates is subsequently employed to construct confidence intervals. While many of these methods can be viewed as sample-based approximations to Bayesian approaches, the selection of model hyperparameters (such as the scale of perturbations or the learning rate for models) in ensemble methods is typically performed using validation on withheld data (a subset of the training data). Many of these methods are heuristic and seldom offer rigorous bounds for the construction of confidence sets. However, while good performance might be expected when test inputs share a similar distribution with the training set, the resulting confidence intervals for test points distant from the training data may be unreliable. Similar to bootstrap, these methods necessitate training multiple predictors from scratch, a requirement that does not align well with applications involving a large number of parameters.

Bayesian methods are designed by assuming a prior distribution over the ground-truth (regression) function. This prior, when combined with observed data through a specified likelihood function, yields a posterior distribution. From this posterior, one can then compute *credible intervals* (Berger, 1985). Note that this is a rather different from what is considered in this work. Prominent methods within the machine learning literature in this category include Bayes-by-Backprop (BBB, Blundell et al., 2015), the linearized Laplace method (Mackay, 1992; Khan et al., 2019; Immer et al., 2021; Antorán et al., 2022; Holzmüller et al., 2023), and Stochastic Gradient MCMC (Nemeth and Fearnhead, 2021).

The linearized Laplace method is related to confidence intervals obtained from asymptotic normality of MLE, and to our approach, as it involves weighting by the inverse of the (regularized) Hessian matrix. Note that (similarly to the case of the normality of the MLE) these results are asymptotic and rely on the Laplace approximation of the posterior. Moreover, these results generally do not argue about high-probability credible intervals. Here, in contrast, we develop high-probability non-asymptotic bounds from the frequentist viewpoint that do not require any approximations at all. Interestingly, the shape of the weighting matrix is slightly different in our case.

Confidence intervals for non-linear least-squares by solving constrained optimization problems Several works explore an approach where the prediction of the ground truth function on x lies in the interval $[\min_{\theta\in\widehat{\Theta}} f(x;\theta), \max_{\theta\in\widehat{\Theta}} f(x;\theta)]$ with high probability for appropriately chosen data-dependent confidence set $\widehat{\Theta}$. These methods were explored in the context of non-linear least squares (Russo and Van Roy, 2014; Foster et al., 2018; Kong

et al., 2021; Ayoub et al., 2020). However, in all of these cases, parameter confidence sets are constructed based on the conservative (data-free) covering number estimates. While this approach proved useful in the design of reinforcement learning and bandit algorithms for under-parameterized problems, for overparameterized problems this leads to vacuous estimates. A related line of work concerns the construction of confidence intervals for non-linear ground truth function, assuming that they lie in the RKHS of a kernel function (Chowdhury and Gopalan, 2017; Csáji and Horváth, 2019).

Appendix B. Details from Section 2

Alignment achieved by Gradient Descent In Section 2 we discussed that any optimization algorithm that converges to the stationary point satisfies the alignment condition Equation (3). In this section, as an example, we demonstrate this for the GD algorithm.

In particular, given the step size $\eta > 0$ and initial parameter vector $\theta_0 \in \mathbb{R}^p$ such that $L_\lambda(\theta_0) < \infty$, the parameter vector $\widehat{\theta}$ is obtained as the limit of a sequence $\theta_1, \theta_2, \ldots$ defined through the standard GD update rule

$$\theta_{t+1} = (1 - \lambda\eta)\theta_t - \eta \nabla L(\theta_t) \qquad \text{for} \quad t = 1, 2, \ldots \ .$$

It is well known that the sequence converges under mild conditions, and so $\widehat{\theta}$ is a *stationary point*. This is implied by the well-known descent lemma:

Lemma 11 (Descent lemma) *Assuming that ∇L_λ is β-Lipschitz w.r.t. ℓ_2 norm and that η satisfies $\eta \leq 1/\beta$, we have*

$$\frac{1}{2\beta} \|\nabla L_\lambda(\theta_t)\|^2 \leq L_\lambda(\theta_t) - L_\lambda(\theta_{t+1}) \ .$$

In particular, $\widehat{\theta}$ in this case satisfies the *alignment* property (asymptotically aligns with the gradient):

Lemma 12 (Alignment of GD) *Assume L_λ is differentiable, ∇L_λ is β-Lipschitz w.r.t. ℓ_2 norm, and that the step size satisfies $\eta \leq 1/\beta$. Then, the limiting point $\widehat{\theta} = \theta_\infty$ satisfies $\lambda\widehat{\theta} = -\nabla L(\widehat{\theta})$.*

Proof Since L_λ is differentiable and β-smooth with $\beta > 0$, by the Descent Lemma, summing over all iterates and using the fact that losses are non-negative,

$$\frac{1}{2\beta} \sum_{t=0}^{\infty} \|\nabla L_\lambda(\theta_t)\|^2 \leq L_\lambda(\theta_0) \ .$$

Boundedness of the above sum implies that $\|\nabla L_\lambda(\theta_t)\| \to 0$ as $t \to \infty$. Therefore, at the limit, we have the alignment of the weight and the gradient vectors, $\lambda\theta = -\nabla L(\theta)$. ■

Appendix C. Details from Section 3

In this section we show auxiliary statements appearing Section 3.

C.1 Linear least squares

In case of ridge regression

$$M(\widehat{\theta}) = \left(\frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \lambda I\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^n x_i x_i^\top\right) \left(\frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \lambda I\right)^{-1}$$
$$= \left(\frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \lambda I\right)^{-1} - \lambda \left(\frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \lambda I\right)^{-2}$$

and so $\mathbb{E}\left[\|\nabla_{\widehat{\theta}} f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}\right] \leq \|x\|^2_{\Sigma^{-1}}$. Now,

$$\begin{aligned}
\mathbb{E}[f(x)] &= x^\top \mathbb{E}[\widehat{\theta}] \\
&= x^\top \left(\frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \lambda I\right)^{-1} \frac{1}{n}\sum_{i=1}^n x_i \,\mathbb{E}[Y_i] \\
&= x^\top \left(\frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \lambda I\right)^{-1} \frac{1}{n}\sum_{i=1}^n x_i x_i^\top \theta^\star \\
&= x^\top \theta^\star - \lambda x^\top \left(\frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \lambda I\right)^{-1} \theta^\star \,.
\end{aligned}$$

Finally, note that

$$\mathrm{Var}(\|\nabla_\theta f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}) \leq \mathbb{E}\left[\|\nabla_\theta f(x;\widehat{\theta})\|^4_{M(\widehat{\theta})}\right] - \mathbb{E}^2\left[\|\nabla_\theta f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}\right] = 0$$

and applying Theorem 2 in this case, we have $c = 0$ and $v = 0$.

C.2 Relationship to Fisher Information Matrix and Asymptotics

Throughout this section, for symmetric matrices A, B we will use $A \preceq B \iff B - A \succeq 0$ to indicate that $\forall x \,:\, x^\top (B - A)x \geq 0$. First, we will need the following technical lemma.

Lemma 13 *Let $A \in \mathbb{R}^{p\times p}$ be an invertible matrix and let $X = \frac{1}{n}\sum_{i=1}^n Z_i B_i$ be such that $Z_1, \ldots, Z_n \sim \mathcal{N}(0, \sigma^2)$ and $B_1, \ldots, B_n \in \mathbb{R}^{p\times p}$ are fixed symmetric matrices with $\|B_i\|_{\mathrm{op}} \leq C$. Then, for any fixed $\alpha > 0$ having setting $\widehat{\lambda} = \alpha + |\min\{0, \lambda_{\min}(X)\}|$, we have*

$$\mathbb{E}[(A + X + \widehat{\lambda} I)^{-1}] \preceq (A + \alpha I - \varepsilon_n I)^{-1} + \frac{\sigma^2}{n} I$$

and where

$$\varepsilon_n = 2\sigma \sqrt{\frac{C \ln\left(\frac{n\sqrt{2p}}{\alpha}\right)}{n}} \,.$$

Proof We will use a concentration inequality on spectral norms of Gaussian series, which says that

Theorem 14 (Tropp et al. (2015, Theorem 4.6.1)) *Under conditions of the lemma on Z_i, B_i, for any $t > 0$,*

$$\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^{n} Z_i B_i\right\|_{\mathrm{op}} \geq t\right\} \leq p\exp\left(-\frac{nt^2}{2\sigma^2 C}\right) \ .$$

Define event $\mathcal{E} = \{\|X\|_{\mathrm{op}} \leq t\}$ for some $t > 0$ to be determined later. Then,

$$\begin{aligned}
\mathbb{E}[(A_\lambda + X)^{-1}] &= \mathbb{E}[(A_\lambda + X)^{-1}\mathbf{1}_{\mathcal{E}}] + \mathbb{E}[(A_\lambda + X)^{-1}\mathbf{1}_{\neg\mathcal{E}}] \\
&\preceq \mathbb{E}[(A_\lambda - tI)^{-1}\mathbf{1}_{\mathcal{E}}] + \sqrt{\mathbb{E}[\|(A_\lambda + X)^{-1}\|_{\mathrm{op}}^2]}\sqrt{\mathbb{P}\left\{\neg\mathcal{E}\right\}}\, I \\
&\preceq \mathbb{E}[(A_\lambda - tI)^{-1}] + \sqrt{\mathbb{E}\left[\frac{1}{|\lambda + \lambda_{\min}(X)|^2}\right]}\sqrt{2p}\exp\left(-\frac{nt^2}{4\sigma^2 C}\right) I
\end{aligned}$$

where we used a Cauchy-Schwarz inequality and a tail bound in Theorem 14. Now we will set λ to depend on $\lambda_{\min}(X)$ to keep the inverse well-defined in a deterministic sense. Set $\lambda = \widehat{\lambda} = \alpha + |\min\{0, \lambda_{\min}(X)\}|$ where $\alpha > 0$ is a free variable to be set later. In this case

$$\mathbb{E}[(A_{\widehat{\lambda}} + X)^{-1}] \preceq \mathbb{E}[(A_{\widehat{\lambda}} - tI)^{-1}] + \frac{1}{\alpha}\sqrt{2p}\exp\left(-\frac{nt^2}{4\sigma^2 C}\right) I$$

Putting all together and choosing

$$t = 2\sigma\sqrt{\frac{C\ln\left(\frac{n\sqrt{2p}}{\alpha}\right)}{n}}$$

we get the statement. ■

An asymptotic bound presented in Section 3 is obtained from the following Proposition 15 assuming boundedness of $\|\nabla f\|, \|\nabla^2 f\|_{\mathrm{op}}$ at $(\theta^\star, x)$.

Proposition 15 *Assume that $\max_i \|\nabla_\theta^2 f(x_i;\theta^\star)\|_{\mathrm{op}} \leq C$. Then, there exists a data-dependent tuning of regularization parameter $\widehat{\lambda}$ such that*

$$\mathbb{E}[M(\theta^\star)] \preceq \mathcal{I}(\theta^\star)^{-1} + \frac{\sigma^2}{n} I \ .$$

Proof Note that

$$\begin{aligned}
\nabla^2 L(\theta^\star) &= \mathcal{I}(\theta^\star) + X \ , \qquad \text{where} \\
\mathcal{I}(\theta^\star) &= \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta f(x_i;\theta^\star)\nabla_\theta f(x_i;\theta^\star)^\top \ , \qquad X = \frac{1}{n}\sum_{i=1}^{n} Z_i \nabla_\theta^2 f(x_i;\theta^\star)
\end{aligned}$$

and so

$$\begin{aligned}\mathbb{E}[M(\theta^\star)] &= \mathbb{E}\left[(\nabla^2 L(\theta^\star)+\lambda I)^{-1}\mathcal{I}(\theta^\star)(\nabla^2 L(\theta^\star)+\lambda I)^{-1}\right] \\ &= \mathbb{E}\left[(\nabla^2 L(\theta^\star)+\lambda I)^{-1}\right] - \mathbb{E}\left[(\nabla^2 L(\theta^\star)+\lambda I)^{-1}(X+\lambda I)(\nabla^2 L(\theta^\star)+\lambda I)^{-1}\right] \\ &\preceq \mathbb{E}\left[(\nabla^2 L(\theta^\star)+\widehat{\lambda} I)^{-1}\right]\end{aligned}$$

where we set $\lambda = \widehat{\lambda} = \alpha + |\min\{0, \lambda_{\min}(X)\}|$ for a free parameter $\alpha > 0$ which ensures that $X + \lambda I$ is PSD. Finally applying Lemma 13 we get

$$\mathbb{E}[M(\theta^\star)] \preceq (\mathcal{I}(\theta^\star) + \alpha I - \varepsilon_n I)^{-1} + \frac{\sigma^2}{n} I$$

where ε_n is given in Lemma 13. But now let us tune α to satisfy

$$\alpha > \varepsilon_n = 2\sigma\sqrt{\frac{C\ln\left(\frac{n\sqrt{2p}}{\alpha}\right)}{n}} \ .$$

The solution to the above inequality involves the Lambert function and further upper-bounding it using the bound of Hoorfar and Hassani (2008, Theorem 2.3 with $y = 1$), we eventually have

$$\alpha > c\sqrt{\ln(b/\alpha)} \quad \Longrightarrow \quad \alpha > \sqrt{\frac{b^2}{2b^2/c^2+1}} \qquad (b, c > 0) \ .$$

Thus, we have

$$\alpha > \sqrt{\frac{2n^2 p}{\frac{n^3 p}{C\sigma^2}+1}}$$

and so the complete setting of λ that gives the statement is

$$\widehat{\lambda} = \lambda = \sqrt{\frac{2n^2 p}{\frac{n^3 p}{C\sigma^2}+1}} + |\min\{0, \lambda_{\min}(X)\}| > 0 \ .$$

■

Proposition 16 *Assume that $\theta \mapsto f(x;\theta)$ is continuously differentiable, and that L_λ is twice continuously differentiable. Assume that the sequence $(\mathbb{E}[\|\nabla_\theta f(x;\widehat{\theta}_n)\|^2_{M(\widehat{\theta}_n)}])_{n=1}^\infty$ converges. Then, almost surely*

$$\limsup_{n\to\infty} \frac{\sqrt{n}}{\ln(n)} \left| f(x;\widehat{\theta}_n) - \mathbb{E}[f(x;\widehat{\theta}_n)] \right| \le \sigma\pi\sqrt{\mathbb{E}\,\|\nabla_\theta f(x;\widehat{\theta}_\infty)\|^2_{M(\widehat{\theta}_\infty)}}$$

Proof Setting $\delta = 2/n^2$ in Theorem 2 we have

$$\sum_{n=1}^{\infty} \mathbb{P}\left\{ \sqrt{\frac{n}{2\sigma^2 \ln(n)}} |\Delta(x;\widehat{\theta}_n)| > \sqrt{\frac{\pi^2}{2}\, \mathbb{E}[\|\nabla_\theta f(x;\widehat{\theta}_n)\|^2_{M(\widehat{\theta}_n)}]} + c_3 \sqrt{\frac{2\ln(n)}{n}} \right\} \leq \frac{\pi^2}{3}$$

which by the Borel-Cantelli lemma gives

$$\mathbb{P}\left\{ \limsup_{n\to\infty} \sqrt{\frac{n}{2\sigma^2 \ln(n)}}\, |\Delta(x;\widehat{\theta}_n)| > \sqrt{\frac{\pi^2}{2} \lim_{n\to\infty} \mathbb{E}[\|\nabla_\theta f(x;\widehat{\theta}_n)\|^2_{M(\widehat{\theta}_n)}]} \right\} = 0$$

where we assumed that the sequence $(\mathbb{E}[\|\nabla_\theta f(x;\widehat{\theta}_n)\|^2_{M(\widehat{\theta}_n)}])_{n=1}^{\infty}$ converges. Finally, invoking DCT, and assuming that f is continuously differentiable, and that L_λ is twice continuously differentiable,

$$\lim_{n\to\infty} \mathbb{E}[\|\nabla_\theta f(x;\widehat{\theta}_n)\|^2_{M(\widehat{\theta}_n)}] = \mathbb{E}[\|\nabla_\theta f(x;\widehat{\theta}_\infty)\|^2_{M(\widehat{\theta}_\infty)}]\ .$$

■

C.3 Uniform Bound Over Test Inputs on the Unit Sphere

Let $\texttt{bound}(\ln(2/\delta))$ denote the bound on $\Delta(x;\widehat{\theta})$ given in Theorem 2, as a function of the $\ln(2/\delta)$ term. Then, it is straightforward to show that

$$\mathbb{P}\left\{\max_{x\in\mathbb{S}^{d-1}}|\Delta(x;\widehat{\theta})| > \texttt{bound}\left(d\,\ln\left(\frac{3n\mathrm{Lip}(\Delta)}{\delta}\right)\right)+\frac{1}{n}\right\}\leq\delta$$

where $\mathrm{Lip}(\Delta)$ is a Lipschitz constant of the gap $x\mapsto\Delta(x;\widehat{\theta})$.

Proof Let $\mathcal{C}_\varepsilon$ be an ε-cover of $\mathbb{S}^{d-1}$. It is well-known that $|\mathcal{C}_\varepsilon|\leq(3/\varepsilon)^d$. Then, for all $\varepsilon>0$,

$$\begin{aligned}
\max_{x\in\mathbb{S}^{d-1}}\Delta(x) &= \max_{x\in\mathbb{S}^{d-1}}\min_{y\in\mathcal{C}_\varepsilon}\{\Delta(x)-\Delta(y)+\Delta(y)\}\\
&\leq \max_{x\in\mathbb{S}^{d-1}}\min_{y\in\mathcal{C}_\varepsilon}\mathrm{Lip}(\Delta)\|x-y\|+\max_{y\in\mathcal{C}_\varepsilon}\Delta(y)\\
&\leq \left\{\mathrm{Lip}(\Delta)\,\varepsilon+\texttt{bound}\left(\ln\left(\frac{|\mathcal{C}_\varepsilon|}{\delta}\right)\right)\right\}\\
&\leq \left\{\mathrm{Lip}(\Delta)\,\varepsilon+\texttt{bound}\left(d\,\ln\left(\frac{3}{\varepsilon\delta}\right)\right)\right\}\\
&\leq \frac{1}{n}+\texttt{bound}\left(d\,\ln\left(\frac{3n\mathrm{Lip}(\Delta)}{\delta}\right)\right)\ .
\end{aligned}$$

where we (lazily) set $\varepsilon=1/(n\mathrm{Lip}(\Delta))$. ■

By examining this bound, note that we incur a factor $\sqrt{d}$ and the logarithm of the Lipschitz constant of the gap, compared to the case when x is fixed. Note that the latter can be simply related to the Lipschitz constant of the predictor in the input, which in practice can be controlled by the training procedure (say, by adding a constraint $\mathrm{Lip}(f(\cdot\,;\widehat{\theta}))\leq B$). On the other hand, in some cases, the Lipschitz constant is already implicitly controlled by the algorithm, which is often the consequence of ℓ^2 regularization.

Lipschitz constant of a neural network Neural networks defined in Eq. (4) can conveniently be written as

$$h_k = D_kW_kD_{k-1}W_{k-1}\cdots D_1W_1x,$$

where D_k is a diagonal matrix defined as $D_k=\mathrm{diag}(a'(W_kh_{k-1}))$, and both (D,h) depend implicitly on the input x. Then, the derivative with respect to a weight matrix has a form:

$$\frac{\mathrm{d}f(x;\theta)}{\mathrm{d}W_k}=(W_KD_{K-1}\cdots W_{k+1}D_k)^\top(h_{k-1})^\top\ .$$

Now we give a bound on the Lipschitz constant of a neural network.

Lemma 17 *Assume that $f(x;\widehat{\theta})$ is a K-layer neural network with ReLU or linear activations as in Eq. (4). Then, any parameter vector θ at the stationary point of L_λ satisfies*

$$\mathrm{Lip}(f(\cdot\,;\theta))\leq\left(\frac{L(\theta)}{\lambda K}\right)^{\frac{K}{2}}\ .$$

Proof Focusing on the Lipschitz constant of a neural network f, we first compute

$$\frac{\mathrm{d}h_k}{\mathrm{d}x} = a'(W_k h_{k-1}) W_k \frac{\mathrm{d}h_{k-1}}{\mathrm{d}x} = \cdots = \prod_{i=1}^{k} a'(W_i h_{i-1}) W_i \ .$$

Then, assuming that a' is uniformly bounded by 1,

$$\left\| \frac{\mathrm{d}h_K}{\mathrm{d}x} \right\| \leq \prod_{k=1}^{K} \|W_k\|_{\mathrm{op}} \ .$$

Therefore,

$$\begin{aligned}
\mathrm{Lip}(f(\cdot;\theta)) &\leq \prod_{k=1}^{K} \|W_k\|_{\mathrm{op}} \\
&= \left(\prod_{k=1}^{K} \|W_k\|_{\mathrm{op}}^2 \right)^{1/2} \\
&\leq \left(\frac{1}{K} \sum \|W_k\|_F^2 \right)^{K/2} = \left(\frac{\|\theta\|_2^2}{K} \right)^{K/2} && \text{(AM-GM inequality)} \\
&= \left(\frac{L(\theta)}{\lambda K} \right)^{K/2} \ .
\end{aligned}$$

■

Proof [Proof of Proposition 3] Finally we control the Lipschitz constant of Δ as

$$\begin{aligned}
\mathrm{Lip}(\Delta) &= \sup_{x \in \mathbb{S}^{d-1}} \|\nabla_x f(x;\widehat{\theta}) - \mathbb{E}\, \nabla_x f(x;\widehat{\theta})\| \\
&\leq \sup_{x \in \mathbb{S}^{d-1}} \|\nabla_x f(x;\widehat{\theta})\| + \sup_{x \in \mathbb{S}^{d-1}} \|\nabla_x \mathbb{E}[f(x;\widehat{\theta})]\| \\
&= \mathrm{Lip}(f(x;\widehat{\theta})) + \mathrm{Lip}(\mathbb{E}[f(x;\widehat{\theta})]) \\
&\leq \left(\frac{L_\lambda(0)}{\lambda K} \right)^{\frac{K}{2}} + \mathbb{E} \left(\frac{L_\lambda(0)}{\lambda K} \right)^{\frac{K}{2}} \\
&= 2 \left(\frac{C}{\lambda K} \right)^{\frac{K}{2}}
\end{aligned}$$

where we have that $L(\theta) \leq L_\lambda(\theta) \leq L_\lambda(0) = C$, and where the penultimate inequality comes by Lemma 17. ■

C.4 Omitted Proofs from Discussion on Assumptions

Proof [Proof of Lemma 4] Consider

$$\nabla^2 L_\lambda(\theta) = \underbrace{\frac{1}{n}\sum_{i=1}^n \nabla_\theta f(x_i;\theta)\nabla_\theta f(x_i;\theta)^\top}_{G(\theta)\succeq 0} + \underbrace{\frac{1}{n}\sum_{i=1}^n (f(x_i;\theta)-Y_i)\,\nabla^2_\theta f(x_i;\theta)}_{A(\theta)} + \lambda I \ .$$

By Cauchy-Schwarz inequality

$$\frac{1}{n}\sum_{i=1}^n (f(x_i;\theta)-Y_i)\,\nabla^2_\theta f(x_i;\theta) \le \sqrt{\mu(\theta)\,\frac{1}{n}\sum_{i=1}^n |f(x_i;\theta)-Y_i|^2}$$

and so

$$\nabla^2 L_\lambda(\theta) \succeq G(\theta) + \left(\lambda - \sqrt{2\mu(\theta)\,L(\theta)}\right) I$$

Finally the statement comes by Cauchy-Schwarz inequality and observing that

$$M(\theta) \preceq \frac{1}{4}\left(\lambda - \sqrt{2L(\theta)\,\mu(\theta)}\right)^{-1} I \ .$$

■

Proof [Proof of Proposition 5] Throughout the proof let $L_\lambda^{(i)}$ be regularized empirical risk defined w.r.t. sample

$$((X_1,Y_1),\dots,(X_{i-1},Y_{i-1}),(X_i',Y_i'),(X_{i+1},Y_{i+1}),\dots,(X_n,Y_n))$$

where X_i' is independent copy of X_i. Denote covariate sample by

$$S = (X_1,\dots,X_n) \ .$$

Then let $\widehat{\theta}^{(i)}$ be a local minimizer of $L_\lambda^{(i)}$. Then, observe that

$$\begin{aligned}
\left|g(S) - g(S^{(i)})\right| &= \left|\mathbb{E}\left[f(x;\widehat{\theta}) \,\middle|\, S\right] - \mathbb{E}\left[f(x;\widehat{\theta}^{(i)}) \,\middle|\, S^{(i)}\right]\right| \\
&= \left|\mathbb{E}\left[f(x;\widehat{\theta}) - f(x;\widehat{\theta}^{(i)}) \,\middle|\, S, X_i'\right]\right| \\
&\le B\,\mathbb{E}\left[\|\widehat{\theta} - \widehat{\theta}^{(i)}\| \,\middle|\, S, X_i'\right]
\end{aligned}$$

by Lipschitzness of f in the parameter vector.

Next we will use an *error bound* property implied by the PL condition (Karimi et al., 2016, Theorem 2):

$$\mu\|\mathrm{Proj}(\theta,\Theta_*) - \theta\| \le \|\nabla L_\lambda(\theta)\| \qquad (\theta\in\mathbb{R}^p)$$

where $\mathrm{Proj}(\theta, \Theta_*)$ is a projection of θ onto the solution set $\Theta_* = \arg\min_\theta L_\lambda(\theta)$. Now focus on the parameter gap. Under Assumption 2 and using the error bound property discussed above

$$
\begin{aligned}
&\|\widehat{\theta} - \widehat{\theta}^{(i)}\| \\
&\quad = \|\Pi_{\Theta_*}(\widehat{\theta}^{(i)}) - \widehat{\theta}^{(i)}\| \\
&\quad \leq \frac{1}{\mu} \|\nabla L_\lambda(\widehat{\theta}^{(i)})\| \\
&\quad = \frac{1}{\mu} \left\| \nabla L_\lambda^{(i)}(\widehat{\theta}^{(i)}) + \frac{1}{n}\nabla \ell(\widehat{\theta}^{(i)}, (X_i, Y_i)) + \lambda \widehat{\theta}^{(i)} - \left(\frac{1}{n}\nabla \ell(\widehat{\theta}^{(i)}, (X_i', Y_i')) + \lambda \widehat{\theta}^{(i)} \right) \right\| \\
&\quad = \frac{1}{n\mu} \left\| \left(f(X_i; \widehat{\theta}^{(i)}) - Y_i \right) \nabla_\theta f(X_i; \widehat{\theta}^{(i)}) - \left(f(X_i'; \widehat{\theta}^{(i)}) - Y_i' \right) \nabla_\theta f(X_i'; \widehat{\theta}^{(i)}) \right\| \\
&\quad \leq \frac{(2C + |Y_i| + |Y_i'|)B}{n\mu}
\end{aligned}
$$

while noting that $\mathbb{E}[|Y_i| \mid X_i] \leq |f^\star(X_i)| + \sigma \leq 1 + \sigma$. ■

C.5 Algorithm Pseudocode from Section 3.3

Algorithm 1 Efficient computation of the weighted norm

Input: Loss L_λ, parameter vector $\widehat{\theta} \in \mathbb{R}^p$, inputs $x_1, \ldots, x_n, x \in \mathbb{R}^d$, error tolerance $\varepsilon > 0$.
Output: Approximation of a weighted norm $\widehat{V} \approx \|\nabla_\theta f(x; \widehat{\theta})\|^2_{M(\widehat{\theta})}$.

Compute $h \leftarrow (\nabla^2 L(\widehat{\theta}) + \lambda I)^{-1} \nabla_\theta f(x; \widehat{\theta})$ by running Algorithm 2 with $v = \nabla_\theta f(x; \widehat{\theta})$, initialization $h_0 = 0$, and error tolerance ε.
if *interpolation* **then**
    # Note: $\widehat{V} \geq \|\nabla_\theta f(x; \widehat{\theta})\|^2_{M(\widehat{\theta})}$ with equality when $\lambda = 0$.[6]
    Compute $\widehat{V} \leftarrow \nabla_\theta f(x; \widehat{\theta})^\top h$ ▷ $\mathcal{O}(p)$
else
    Compute $\widehat{V} \leftarrow \frac{1}{n} \sum_{i=1}^n (\nabla_\theta f(x_i; \widehat{\theta})^\top h)^2$ ▷ $\mathcal{O}(pn)$
end if
return $\widehat{V}$

Algorithm 2 Efficient Hessian-Inverse-Vector Product computation using autodiff and CG

Input: Loss L_λ, parameters $\widehat{\theta} \in \mathbb{R}^p$, target vector $v \in \mathbb{R}^p$, initialization $h_0 \in \mathbb{R}^p$, error tol. $\varepsilon > 0$.
Output: Approximation $h \approx [\nabla^2 L_\lambda(\widehat{\theta})]^{-1} v$.

function HessianVectorProduct(L, θ, z)
    # Computes the product Hz without forming H
    Compute gradient $g(\theta) = \nabla L(\theta)$ ▷ $\mathcal{O}(C_L)$
    Compute scalar $s(\theta) = g(\theta)^\top z$ ▷ $\mathcal{O}(p)$
    Compute $Hz = \nabla s(\theta) = \nabla_\theta[(\nabla L(\theta))^\top z]$ ▷ Using autodiff $\mathcal{O}(C_L)$, not $\mathcal{O}(p^2)$
    return Hz
end function
Define the matrix-vector product operation $A(z)$ for $H = \nabla^2 L(\widehat{\theta}) + \lambda I$:
$A(z) :=$ HessianVectorProduct(L, θ, z) $+ \lambda z$
Solve the linear system $Hh = v$ for h using CG:
$h \leftarrow$ ConjugateGradient(A, v, h_0) ▷ $k = \Omega\left(\sqrt{\kappa(H)} \ln(\frac{1}{\varepsilon})\right)$ iterations
return h

6. In the interpolation regime with $\lambda > 0$, the Hessian simplifies to $H = G + \lambda I$ and so $\nabla_\theta f(x; \widehat{\theta})^\top H^{-1} \nabla_\theta f(x; \widehat{\theta}) = \|\nabla_\theta f(x; \widehat{\theta})\|^2_{M(\widehat{\theta})} + \lambda \|h\|^2$, yielding a conservative upper bound.

C.6 Behaviour of CG Outlined in Table 1

1. **Case $H \succ 0$.**

   In this case H is symmetric positive definite, so the linear system $Hs = g$ has a unique solution
   $$s = H^{-1}g.$$
   This is the standard setting for CG, and therefore CG converges normally.

2. **Case $H \succeq 0$, singular, and $g \in \mathrm{Range}(H)$.**

   Although H^{-1} does not exist, the system
   $$Hs = g$$
   is still consistent. If CG is initialized at 0, then all iterates remain in $\mathrm{Range}(H)$. Since the unique solution of $Hs = g$ lying in $\mathrm{Range}(H)$ is the minimum-norm solution, CG converges to
   $$s = H^{\dagger}g.$$

3. **Case $H \succeq 0$, singular, and $g \notin \mathrm{Range}(H)$.**

   In this case the system $Hs = g$ is inconsistent. Indeed, for every s, we have $Hs \in \mathrm{Range}(H)$, so the component of g in $\mathrm{Ker}(H)$ cannot be matched. Therefore the residual satisfies
   $$\|Hs - g\| \geq \|P_0 g\|.$$
   Thus CG cannot converge to an exact solution, and the residual stagnates.

4. **Case H nearly singular.**

   When H has very small nonzero eigenvalues, the system may still be solvable, but it is ill-conditioned. Components of g along these eigendirections are amplified by the inverse, since
   $$H^{-1}g$$
   contains factors of order $1/\mu$, where μ is a small eigenvalue of H. Consequently, CG may converge slowly, and the resulting weighted norm can become very large.

Appendix D. Other Omitted Proofs

This section contains proofs of minor statements not included in the main body.

D.1 Pointwise Bound with Polynomial Dependence on δ

Theorem 18 *Assume that $\widehat{\theta}$ is a local minimizer of L_λ and that the test input x is fixed. Then, for any $\delta \in (0,1)$*

$$\mathbb{P}\left\{|f(x;\widehat{\theta}) - \mathbb{E}[f(x;\widehat{\theta})]| > \sigma\sqrt{\frac{1}{\delta n}\,\mathbb{E}\left[\|\nabla_\theta f(x;\widehat{\theta})\|^2_{M(\widehat{\theta})}\right]}\right\} \leq \delta$$

To prove this theorem, we start with a basic concentration inequality which tells us the function of multiple random elements concentrates well around its mean when its conditional variances are small:

Proposition 19 *Let $S = (Z_1, \ldots, Z_n)$ be a tuple of independent random elements and let S' be its independent copy. Then, for continuously differentiable $g : \mathcal{Z}^n \to \mathbb{R}$ and for any $\delta \in (0,1]$,*

$$\mathbb{P}\left\{|g(S) - \mathbb{E}[g(S)]| > \sqrt{\frac{1}{\delta}\sum_{i=1}^n \mathbb{E}\,\mathrm{Var}(g(S) \mid S^{\setminus i})}\right\} \leq \delta\ .$$

Proof For i.i.d. random variables X, Y we have $\mathrm{Var}(X) = \frac{1}{2}\,\mathbb{E}[(X-Y)^2]$. Thus,

$$\mathrm{Var}(g(S) \mid S^{\setminus i}) = \mathbb{E}\left[(g(S) - \mathbb{E}[g(S) \mid S^{\setminus i}])^2 \;\middle|\; S^{\setminus i}\right] = \frac{1}{2}\,\mathbb{E}\left[\left(g(S) - g(S^{(i)})\right)^2 \;\middle|\; S^{\setminus i}\right]$$

that is where we took expectation only with respect to Z_i, Z_i'. On the other hand, by the Efron-Stein inequality (Boucheron et al., 2013)

$$\mathrm{Var}(g(S)) \leq \frac{1}{2}\sum_{i=1}^n \mathbb{E}[(g(S) - g(S^{(i)}))^2] = \sum_{i=1}^n \mathbb{E}\,\mathrm{Var}(g(S) \mid S^{\setminus i})\ .$$

Finally, the application of Chebyshev's inequality completes the proof. ■

We first make a connection between conditional variances and gradients through the following key general inequality (Boucheron et al., 2013):

Lemma 20 (Gaussian Poincaré inequality) *Let Z be a Gaussian random vector, distributed according to $\mathcal{N}(0, I_d)$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be any continuously differentiable function. Then,*

$$\mathrm{Var}(f(Z)) \leq \mathbb{E}[\|\nabla f(Z)\|^2]\ .$$

Combining this lemma with Proposition 19 already establishes that concentration is controlled by the squares of the gradient norms. All that is left to do is to analyze them for $f(x;\widehat{\theta})$, which is done in the following lemma. Putting together Proposition 19 and Lemmas 7 and 20 completes the proof.

Proof [Proof of Lemma 9] The case $t = 0$ is immediate, since $\phi(0) \geq 0$. Hence assume $t > 0$. Choose solution

$$\lambda_* = \min\left\{\frac{t}{4B}, \left(\frac{t}{4v^2}\right)^{1/3}, \frac{1}{\sqrt{2c}}\right\},$$

Then, $0 \leq \lambda_* \leq 1/\sqrt{2c} < 1/\sqrt{c}$ so λ_* is feasible. Since $c\lambda_*^2 \leq 1/2$, we have $1 - c\lambda_*^2 \geq 1/2$ and so

$$\frac{\lambda_*^4(v^2/2)}{1 - c\lambda_*^2} \leq v^2\lambda_*^4$$

and thus

$$\phi(t) \geq \lambda_* t - \lambda_*^2 B - v^2\lambda_*^4 \,.$$

Now, by the definition of λ_*, if $B > 0$, then $\lambda_* \leq t/(4B)$ and hence

$$B\lambda_*^2 \leq \frac{\lambda_* t}{4}.$$

If $B = 0$, the same bound is trivial. Similarly, if $v > 0$, then

$$\lambda_* \leq \left(\frac{t}{4v^2}\right)^{1/3},$$

so

$$v^2\lambda_*^4 \leq \frac{\lambda_* t}{4}.$$

If $v = 0$, this inequality is again trivial. Combining these estimates gives

$$\phi(t) \geq \lambda_* t - \frac{\lambda_* t}{4} - \frac{\lambda_* t}{4} = \frac{\lambda_* t}{2}.$$

Substituting the definition of λ_*, we obtain

$$\phi(t) \geq \frac{t}{2}\min\left\{\frac{t}{4B}, \left(\frac{t}{4v^2}\right)^{1/3}, \frac{1}{\sqrt{2c}}\right\}.$$

Thus

$$\phi(t) \geq \min\left\{\frac{t^2}{8B}, \frac{t^{4/3}}{2 \cdot 4^{1/3}v^{2/3}}, \frac{t}{2\sqrt{2c}}\right\}$$

and relaxing some numerical constants we get the statement. ■

References

Javier Antorán, David Janz, James U Allingham, Erik Daxberger, Riccardo Rb Barbano, Eric Nalisnick, and José Miguel Hernández-Lobato. Adapting the Linearised Laplace Model Evidence for Modern Deep Learning. In *International Conference on Machine Learing (ICML)*, 2022.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. In *International Conference on Machine Learing (ICML)*, 2019.

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-Based Reinforcement Learning with Value-Targeted Regression. In *International Conference on Machine Learing (ICML)*, 2020.

James O Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Science & Business Media, 1985.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Network. In *International Conference on Machine Learing (ICML)*, 2015.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Olivier Bousquet and André Elisseeff. Stability and Generalization. *Journal of Machine Learning Research*, 2(Mar):499–526, 2002.

Olivier Bousquet, Yegor Klochkov, and Nikita Zhivotovskiy. Sharper Bounds for Uniformly Stable Algorithms. In *Conference on Computational Learning Theory (COLT)*, 2020.

Zachary Charles and Dimitris Papailiopoulos. Stability and Generalization of Learning Algorithms that Converge to Global Optima. In *International Conference on Machine Learing (ICML)*, 2018.

Sayak Ray Chowdhury and Aditya Gopalan. On Kernelized Multi-Armed Bandits. In *International Conference on Machine Learing (ICML)*, 2017.

Balázs Csanád Csáji and Bálint Horváth. Distribution-Free Uncertainty Quantification for Kernel Methods by Gradient Perturbations. *Machine Learning*, 108(8):1677–1699, 2019.

Alexandre Défossez, Léon Bottou, Francis R Bach, and Nicolas Usunier. A Simple Convergence Proof of Adam and Adagrad. *Transactions on Machine Learning Research (TMLR)*, 2022.

Luc Devroye and Terry Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207, 1979.

Simon Du, Xiyu Zhai, Barnabás Póczos, and Aarti Singh. Gradient Descent Provably Optimizes Over-parameterized Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Vikranth Dwaracherla, Zheng Wen, Ian Osband, Xiuyuan Lu, Seyed Mohammad Asghari, and Benjamin Van Roy. Ensembles for Uncertainty Estimation: Benefits of Prior Functions and Bootstrapping. In *Transactions on Machine Learning Research (TMLR)*, 2023.

Bradley Efron and Robert J Tibshirani. *An introduction to the Bootstrap*. Chapman and Hall/CRC, 1994.

Dylan Foster, Alekh Agarwal, Miroslav Dudik, Haipeng Luo, and Robert Schapire. Practical Contextual Bandits with Regression Oracles. In *International Conference on Machine Learing (ICML)*, 2018.

Spencer Frei and Quanquan Gu. Proxy Convexity: A Unified Framework for the Analysis of Neural Networks Trained by Gradient Descent. *Advances in Neural Information Processing Systems*, 2021.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learing (ICML)*, 2016.

Saeed Ghadimi and Guanghui Lan. Stochastic First- and Zeroth-order Methods for Nonconvex Stochastic Programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Science & Business Media, 2006.

David Holzmüller, Viktor Zaverkin, Johannes Kästner, and Ingo Steinwart. A Framework and Benchmark for Deep Batch Active Learning for Regression. *Journal of Machine Learning Research*, 24:1–81, 2023.

Abdolhossein Hoorfar and Mehdi Hassani. Inequalities on the Lambert w function and hyperpower function. *J. Inequal. Pure and Appl. Math*, 9(2):5–9, 2008.

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of Bayesian neural nets via local linearization. In Arindam Banerjee and Kenji Fukumizu, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Paata Ivanisvili, Ramon Van Handel, and Alexander Volberg. Rademacher type and enflo type coincide. *Annals of mathematics*, 192(2):665–678, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *Advances in Neural Information Processing Systems*, 2018.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-Łojasiewicz Condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2016.

Mohammad Emtiyaz Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. Approximate Inference Turns Deep Networks into Gaussian Processes. In *Advances in Neural Information Processing Systems*, 2019.

Dingwen Kong, Ruslan Salakhutdinov, Ruosong Wang, and Lin F Yang. Online Sub-Sampling for Reinforcement Learning with General Function Approximation. In *Workshop on Reinforcement Learning Theory at ICML*, 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems*, 2017.

Tor Lattimore. A Lower Bound For Linear and Kernel Regression with Adaptive Covariates. In *Conference on Computational Learning Theory (COLT)*, 2023.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Michel Ledoux. Concentration of measure and logarithmic Sobolev inequalities. In *Seminaire de probabilites XXXIII*, pages 120–216. Springer, 2006.

Haochuan Li, Alexander Rakhlin, and Ali Jadbabaie. Convergence of Adam Under Relaxed Assumptions. *Advances in Neural Information Processing Systems*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

David John Cameron Mackay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, USA, 1992.

Christopher Nemeth and Paul Fearnhead. Stochastic gradient Markov chain Monte Carlo. *Journal of the American Statistical Association*, 116(533):433–450, 2021.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep Exploration via Bootstrapped DQN. *Advances in Neural Information Processing Systems*, 2016.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized Prior Functions for Deep Reinforcement Learning. *Advances in Neural Information Processing Systems*, 2018.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Epistemic Neural Networks. *Advances in Neural Information Processing Systems*, 2023.

Barak A Pearlmutter. Fast Exact Multiplication by the Hessian. *Neural computation*, 6(1): 147–160, 1994.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Gilles Pisier. Probabilistic methods in the geometry of Banach spaces. *Lecture Notes in Mathematics*, pages 167–241, 1986.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT press, 2006.

Daniel Russo and Benjamin Van Roy. Learning to Optimize Via Posterior Sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Jun Shao. *Mathematical Statistics*. Springer, 2003.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *International Conference on Machine Learing (ICML)*, 2010.

Joel A Tropp et al. An Introduction to Matrix Concentration Inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Aad W Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: Networked Science in Machine Learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

Simon Vary, Tyler Farghly, Ilja Kuzborskij, and Patrick Rebeschini. On-Average Stability of Multipass Preconditioned SGD and Effective Dimension. In *Conference on Computational Learning Theory (COLT)*, 2026. To appear.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Larry A Wasserman. *All of Statistics: A Concise Course in Statistical Inference*, volume 26. Springer, 2004.

Robert L Winkler. A Decision-Theoretic Approach to Interval Estimation. *Journal of the American Statistical Association*, 67(337):187–191, 1972.

Yasin Abbasi Yadkori. *Online Learning for Linearly Parametrized Control Problems*. Phd, University Of Alberta, 2012.

Pan Zhou, Xingyu Xie, Zhouchen Lin, and Shuicheng Yan. Towards Understanding Convergence and Generalization of AdamW. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.