Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Curvature-Adaptive Function-Space Confidence Ellipses for Neural Network Regression

Nassamon Bootwisas
Rajamangala University of Technology Suvarnabhumi
Pasin Marupanthorn
quantfilab@gmail.com

Maejo University
Uparittha Intarasat
Rajamangala University of Technology Suvarnabhumi

Research Article

Keywords: neural network regression, function-space inference, curvature correction

Posted Date: May 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9781259/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Curvature-Adaptive Function-Space Confidence Ellipses for Neural Network Regression

Nassamon Bootwisas[1], Pasin Marupanthorn[2*], Uparittha Intarasat[1]

[1]Division of Computational Science, Faculty of Science and Technology, Rajamangala University of Technology Suvarnabhumi, Phra Nakhon Si Ayutthaya, 13000, Thailand.

[2*]Department of Mathematics, Faculty of Science, Maejo University, 63 Moo 4, Nong Han, San Sai, Chiang Mai, 50290, Thailand.

*Corresponding author(s). E-mail(s): quantfilab@gmail.com;

Abstract

Neural-network uncertainty is often reported either for model weights or for future noisy observations, but many scientific questions require joint uncertainty about the learned mean function at selected inputs. This paper proposes curvature-adaptive function-space confidence ellipses for neural-network regression. The method first constructs a Wald-type ellipse from a local neural tangent approximation and then inflates its radius using a computable second-order curvature score. The correction preserves the center and orientation of the ordinary ellipse while protecting against nonlinear distortion of the fitted function map. The paper develops the formal construction, explicit Hessian-based curvature evaluation for the simulation studies, scalable Hessian-vector alternatives for larger models, calibration choices, and a local coverage argument. The exactness in the numerical work refers to the evaluation of the selected function-value Hessian matrices; the resulting confidence regions remain local second-order inferential approximations rather than exact finite-sample guarantees. Reproducible simulations support the theory. In a refitted one-hidden-unit model, ordinary Wald coverage falls to 87.5 percent and 86.5 percent in boundary and noisy settings, while the curvature-adaptive ellipse reaches 95.0 percent and 95.5 percent. In a larger eight-hidden-unit explicit-Hessian local Gaussian experiment with 25 parameters, boundary coverage improves from 84.3 percent to 95.7 percent and noisy-boundary coverage improves from 92.7 percent to 97.7 percent. These results show that explicitly evaluated second-order curvature information can strengthen linearized function-space inference when selected neural-network mean responses lie in locally nonlinear regions.

Keywords: neural network regression; function-space inference; curvature correction

1 Introduction

Neural networks are now standard regression tools in scientific, engineering, and data-intensive applications, but their practical value depends not only on point prediction accuracy but also on uncertainty quantification. In many applications the relevant uncertainty is not attached to individual weights. Instead, the analyst wants uncertainty about the learned mean function at scientifically meaningful inputs: a low and high dose in a response curve, early and late times in a growth process, or two operating states in a control problem. These questions ask for joint uncertainty about a finite-dimensional function vector,

$$F(\theta) = \{f(x_1^*; \theta), \ldots, f(x_m^*; \theta)\}^\mathsf{T},$$

rather than a confidence region for the full parameter vector. This distinction is important because neural-network weights are high-dimensional, non-unique, and often poorly identified, while function values are directly interpretable.

Bayesian neural networks provide one principled route to predictive uncertainty. Early work by MacKay and Neal established Bayesian treatments of neural networks and connected weight uncertainty to function uncertainty [1, 2]. Later approximate methods made Bayesian-style uncertainty more accessible. Monte Carlo dropout interprets dropout training as approximate Bayesian inference [3]; deep ensembles provide a simple and scalable alternative for calibrated predictive uncertainty [4]; and stochastic weight averaging Gaussian approximations summarize posterior uncertainty using the geometry of stochastic-gradient iterates [5]. These methods are useful, but they usually aim at predictive distributions or posterior approximations rather than frequentist confidence ellipses for selected mean function values.

A related line of work studies neural networks directly in function space. Infinite-width networks with suitable random weights converge to Gaussian-process limits [6], and the neural tangent kernel (NTK) describes the training dynamics of wide networks through a kernel evolution in function space [7]. Empirical NTK and linearized-network methods use this idea post hoc by approximating the trained network near its fitted parameters. Laplace approximations provide another post-hoc route to neural-network uncertainty. Recent work argues that Laplace methods can be competitive and practical for Bayesian deep learning [8], while function-space priors for Laplace approximation attempt to avoid the limited interpretability of weight-space priors by placing structure directly on the induced function [9]. These approaches motivate the present paper's function-space orientation, but most of them rely primarily on first-order linearization or Bayesian posterior approximation.

Classical nonlinear regression offers a complementary perspective. Wald confidence regions are simple and geometrically interpretable, but their accuracy depends on whether the model is close to linear over the uncertainty region. Bates and Watts introduced intrinsic and parameter-effects curvature measures to quantify nonlinearity in nonlinear least squares [10]. Cook and Weisberg emphasized that Wald regions

may differ substantially from uncertainty described by profile likelihood in nonlinear regression [11]. Donaldson and Schnabel showed through computational comparisons that linearized confidence regions can perform poorly when nonlinear effects are important [12]. This literature suggests that uncertainty regions should account for local curvature rather than rely only on first-order approximations.

Conformal prediction is another important uncertainty framework for machine learning. It produces prediction sets with finite-sample distribution-free coverage under exchangeability and can be applied on top of black-box predictors [13, 14]. Conformalized quantile regression improves adaptivity for heteroscedastic regression while preserving coverage guarantees [15]. However, conformal prediction usually targets future noisy responses rather than confidence regions for the unknown conditional mean function. The distinction matters: a prediction interval should contain a future observation, whereas a function-space confidence ellipse should contain the vector of true mean responses at selected inputs.

This paper develops curvature-adaptive function-space confidence ellipses for neural-network regression. The method begins with a first-order Gauss–Newton or neural-tangent linearization, which gives a Wald-type ellipsoid for $F(\theta_\star)$. It then measures the second-order distortion of the map $\theta \mapsto F(\theta)$ using Hessian matrices for the selected function values, or Hessian-vector products when explicit Hessian storage is too expensive, and inflates the chi-square cutoff according to a local function-space curvature score. In the simulations, these Hessian matrices are evaluated explicitly for the finite-dimensional networks under study. This makes the curvature calculation exact up to numerical precision, but it does not make the entire confidence procedure exact: the inference still relies on a local Gaussian approximation to the estimator and a second-order Taylor expansion of the function map. Thus the proposal should be viewed as a frequentist, curvature-aware refinement of linearized neural-network uncertainty rather than as a replacement for Bayesian, bootstrap, NTK, or conformal methods.

The contribution is fourfold. First, the paper formulates neural-network confidence regions directly for finite-dimensional function values, avoiding the interpretability problems of full weight-space ellipsoids. Second, it defines a computable empirical curvature score for the selected function-space target and distinguishes exact Hessian evaluation for moderate networks from Hessian-vector implementations for larger models. Third, it provides a local coverage argument showing how curvature inflation controls the second-order remainder that causes ordinary Wald ellipses to undercover. Fourth, it gives a reproducible simulation study showing that the ordinary ellipse can undercover in boundary and noisy settings, while the curvature-adaptive ellipse restores approximately nominal coverage at the expected cost of larger area. The resulting framework bridges nonlinear-regression curvature diagnostics and modern function-space neural-network uncertainty.

The remainder of the paper is organized as follows. Section 2 defines the function-space inferential target and distinguishes confidence regions for the mean function from prediction regions for future observations. Section 3 derives the first-order function-space Wald ellipse. Section 4 introduces the empirical curvature score that measures local second-order distortion. Section 5 defines the curvature-adaptive ellipse, and Section 6 gives the local coverage argument. Sections 7 and 8 present the algorithm

and calibration choices. Section 9 reports the simulation evidence, including both a refitted small neural model and a larger explicit-Hessian local Gaussian experiment. The final section discusses implications, practical scope, and future extensions.

2 Problem setup

Let $(x_i, y_i)_{i=1}^n$ denote training observations, where $x_i \in \mathcal{X} \subset \mathbb{R}^d$ and

$$y_i = f_0(x_i) + \varepsilon_i, \qquad \mathbb{E}(\varepsilon_i \mid x_i) = 0, \qquad \mathrm{Var}(\varepsilon_i \mid x_i) = \sigma^2. \tag{1}$$

The unknown regression function f_0 is approximated by a neural network

$$f(x; \theta), \qquad \theta \in \Theta \subset \mathbb{R}^p,$$

where p may be large relative to the number of scientifically meaningful quantities to be inferred. The fitted parameter vector $\widehat{\theta}$ is obtained by minimizing the regularized empirical squared-error loss

$$L_n(\theta) = \frac{1}{2} \sum_{i=1}^{n} \{y_i - f(x_i; \theta)\}^2 + \frac{\lambda}{2} \|\theta\|_2^2, \qquad \lambda \geq 0. \tag{2}$$

The regularization term may represent weight decay, numerical damping, or a local ridge approximation used to stabilize the inverse curvature matrix. In the theoretical discussion, $\theta_\star$ denotes the population target or pseudo-true parameter,

$$\theta_\star \in \arg\min_{\theta \in \Theta} \mathbb{E}\{Y - f(X; \theta)\}^2, \tag{3}$$

so the framework covers both correctly specified and misspecified neural-network regression models.

The objective of this paper is not to construct a confidence region for the full parameter vector θ. Such a region is usually difficult to interpret because neural-network parameters are high-dimensional, redundant, and often non-identifiable. Instead, we target a finite-dimensional projection of the learned regression function. Let

$$\mathcal{X}_* = \{x_1^*, \ldots, x_m^*\}$$

be a set of evaluation inputs chosen by the analyst. Define the function-space map

$$F(\theta) = \begin{pmatrix} f(x_1^*; \theta) \\ \vdots \\ f(x_m^*; \theta) \end{pmatrix} \in \mathbb{R}^m. \tag{4}$$

The inferential target is therefore

$$F_\star = F(\theta_\star), \tag{5}$$

which represents the mean neural-network response at the selected inputs. When $m = 2$, a confidence region for $F_\star$ is a confidence ellipse in the plane

$$\big(f(x_1^*;\theta_\star),\ f(x_2^*;\theta_\star)\big).$$

For larger m, the same construction gives an ellipsoidal confidence region for a finite grid of function values and can be projected into confidence bands or pairwise ellipses.

This formulation has two advantages. First, it avoids the interpretability problems of weight-space inference and focuses directly on the fitted function. Second, it makes the role of nonlinear neural-network geometry explicit: the map

$$\theta \mapsto F(\theta)$$

may be locally distorted by curvature even when the first-order neural tangent approximation is accurate near the fitted parameters. The proposed method therefore begins with a first-order function-space Wald ellipse and then adjusts its cutoff using a curvature score that measures the second-order deviation of $F(\theta)$ from its local linear approximation.

Throughout the paper, the confidence region is for the conditional mean function $F_\star$, not for future noisy observations. A prediction ellipse for future responses would require adding the observational noise variance σ^2 to the function-space covariance.

3 First-order function-space Wald ellipse

The construction begins with a local linear approximation of the fitted neural network. Let

$$r_i(\theta) = y_i - f(x_i;\theta) \tag{6}$$

denote the training residuals and define the training Jacobian

$$J = \frac{\partial}{\partial \theta^\mathsf{T}} \begin{pmatrix} f(x_1;\widehat{\theta}) \\ \vdots \\ f(x_n;\widehat{\theta}) \end{pmatrix} \in \mathbb{R}^{n\times p}. \tag{7}$$

Similarly, for the selected evaluation inputs $\mathcal{X}_* = \{x_1^*,\dots,x_m^*\}$, define the function-space Jacobian

$$G = \frac{\partial F(\theta)}{\partial \theta^\mathsf{T}}\bigg|_{\theta=\widehat{\theta}} = \frac{\partial}{\partial \theta^\mathsf{T}} \begin{pmatrix} f(x_1^*;\theta) \\ \vdots \\ f(x_m^*;\theta) \end{pmatrix}_{\theta=\widehat{\theta}} \in \mathbb{R}^{m\times p}. \tag{8}$$

The matrix J describes how the fitted network responds to parameter perturbations on the training inputs, whereas G describes how the selected function values respond to the same perturbations.

Under a local least-squares approximation, the parameter estimator satisfies

$$\widehat{\theta} - \theta_\star \approx (J^\mathsf{T} J + \lambda I_p)^\dagger J^\mathsf{T} \varepsilon, \tag{9}$$

where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^\mathsf{T}$, $\lambda \geq 0$ is a ridge or weight-decay parameter, and † denotes the Moore–Penrose inverse. This gives the Gauss–Newton covariance approximation

$$\widehat{\Sigma}_\theta = \widehat{\sigma}^2 (J^\mathsf{T} J + \lambda I_p)^\dagger. \tag{10}$$

When $J^\mathsf{T} J + \lambda I_p$ is nonsingular, the generalized inverse in (10) is replaced by the ordinary inverse. The ridge term is useful in neural-network settings because the empirical curvature matrix is often ill-conditioned or rank-deficient.

The noise variance may be estimated by

$$\widehat{\sigma}^2 = \frac{1}{n - d_{\text{eff}}} \sum_{i=1}^{n} \{y_i - f(x_i; \widehat{\theta})\}^2, \tag{11}$$

where d_{eff} is an effective degrees-of-freedom correction. A natural choice under ridge regularization is

$$d_{\text{eff}} = \operatorname{tr}\{J(J^\mathsf{T} J + \lambda I_p)^{-1} J^\mathsf{T}\}. \tag{12}$$

When the network is heavily overparameterized, (11) should be interpreted as a stabilized local estimate rather than an exact unbiased residual variance estimator.

The first-order delta method transfers parameter uncertainty into function space:

$$F(\widehat{\theta}) - F(\theta_\star) \approx G(\widehat{\theta} - \theta_\star). \tag{13}$$

Therefore the function-space covariance matrix is estimated by

$$\widehat{\Sigma}_F = G \widehat{\Sigma}_\theta G^\mathsf{T}. \tag{14}$$

This covariance is $m \times m$, regardless of the number of neural-network parameters. Thus, the inferential problem is reduced from a high-dimensional weight-space problem to a low-dimensional function-space problem. If $\widehat{\Sigma}_F$ is singular or nearly singular, we use the stabilized matrix

$$\widehat{\Sigma}_{F,\tau} = \widehat{\Sigma}_F + \tau I_m, \qquad \tau > 0, \tag{15}$$

with τ small relative to the leading eigenvalues of $\widehat{\Sigma}_F$.

The ordinary first-order function-space Wald region is

$$C^{\text{W}}_{1-\alpha}(\mathcal{X}_*) = \left\{ u \in \mathbb{R}^m : (u - F(\widehat{\theta}))^\mathsf{T} \widehat{\Sigma}_F^{-1} (u - F(\widehat{\theta})) \leq q_{m,1-\alpha} \right\}, \tag{16}$$

where $q_{m,1-\alpha}$ is the $(1-\alpha)$-quantile of a χ^2_m distribution. For $m = 2$, (16) is a confidence ellipse for the pair

$$\big(f(x_1^*; \theta_\star), f(x_2^*; \theta_\star)\big).$$

For $m > 2$, it is an ellipsoid for the vector of mean function values on the selected input grid.

The region in (16) is attractive because it is simple, low-dimensional, and invariant to many irrelevant reparameterizations of the neural-network weights. However, it is only a first-order construction. Its validity depends on the accuracy of the local linear approximation

$$F(\theta) \approx F(\widehat{\theta}) + G(\theta - \widehat{\theta}).$$

When the map $\theta \mapsto F(\theta)$ is strongly curved near $\widehat{\theta}$, the Wald ellipse can be too small and may undercover the true function value. The next section introduces a function-space curvature measure that quantifies this second-order distortion and motivates the curvature-adaptive correction.

4 Function-space curvature

The first-order Wald ellipse in (16) is based on the local approximation

$$F(\theta) \approx F(\widehat{\theta}) + G(\theta - \widehat{\theta}). \tag{17}$$

This approximation is accurate only when the map $\theta \mapsto F(\theta)$ is nearly linear over the parameter-uncertainty region. Neural networks can violate this condition even when the fitted loss is locally stable, because small parameter perturbations may induce nonlinear changes in selected function values. We therefore introduce a curvature score that measures the size of the second-order remainder relative to the first-order function-space uncertainty scale.

Let

$$H_j = \nabla_\theta^2 f(x_j^*; \widehat{\theta}), \qquad j = 1, \ldots, m, \tag{18}$$

denote the Hessian matrix of the jth selected function value with respect to the network parameters. For a perturbation $h = \theta - \widehat{\theta}$, Taylor expansion gives

$$F(\widehat{\theta} + h) = F(\widehat{\theta}) + Gh + R_F(h), \tag{19}$$

where the leading second-order remainder is

$$R_F(h) \approx \frac{1}{2} \begin{pmatrix} h^\mathsf{T} H_1 h \\ \vdots \\ h^\mathsf{T} H_m h \end{pmatrix}. \tag{20}$$

Since the relevant perturbations are not arbitrary but are shaped by estimator uncertainty, we standardize them using the local covariance matrix. Let L be a matrix square root of $\widehat{\Sigma}_\theta$, so that

$$\widehat{\Sigma}_\theta = LL^\mathsf{T},$$

and write

$$h = Lz,$$

where z is a standardized local perturbation. Substituting $h = Lz$ into (20) gives

$$R_F(z) = \frac{1}{2} \begin{pmatrix} z^\mathsf{T} L^\mathsf{T} H_1 L z \\ \vdots \\ z^\mathsf{T} L^\mathsf{T} H_m L z \end{pmatrix}. \tag{21}$$

The remainder in (21) must be compared with the scale of first-order function-space uncertainty. We therefore whiten it by $\widehat{\Sigma}_F^{-1/2}$. This yields the following empirical curvature measure.

Definition 1 (Empirical function-space curvature)**.** Let $\widehat{\Sigma}_F$ be positive definite after any required numerical ridge stabilization. For a local radius $r > 0$, define

$$\widehat{\kappa}_F(r) = \sup_{0<\|z\|_2\le r} \frac{\left\|\widehat{\Sigma}_F^{-1/2} R_F(z)\right\|_2}{\|z\|_2^2}. \tag{22}$$

The quantity $\widehat{\kappa}_F(r)$ is dimensionless. It measures the largest whitened second-order distortion of the selected function values over a standardized parameter neighborhood of radius r. Equivalently, it is the smallest number satisfying

$$\left\|\widehat{\Sigma}_F^{-1/2} R_F(z)\right\|_2 \le \widehat{\kappa}_F(r)\|z\|_2^2, \qquad \|z\|_2 \le r. \tag{23}$$

Small values of $\widehat{\kappa}_F(r)$ indicate that the neural tangent approximation is adequate for the chosen function-space target. Large values indicate that the ordinary Wald ellipse may be too small because the nonlinear remainder is large relative to the first-order uncertainty scale.

In practice, two separate numerical issues should be distinguished. The Hessian matrices $H_1, \dots, H_m$ can be formed explicitly for small and moderate networks, so the local second-order curvature term is evaluated exactly up to machine precision. The supremum over all standardized local directions, however, is an optimization problem and need not be solved exactly. It can be approximated using randomized directions z_b, $b = 1, \dots, B$, with $\|z_b\|_2 \le r$:

$$\widehat{\kappa}_{F,B}(r) = \max_{1\le b\le B} \frac{\left\|\widehat{\Sigma}_F^{-1/2} R_F(z_b)\right\|_2}{\|z_b\|_2^2}. \tag{24}$$

For the simulation studies in Section 9, the Hessian matrices are computed explicitly because the parameter dimensions are small enough for exact second-order calculation. This statement concerns the curvature evaluation, not the statistical coverage guarantee: the confidence ellipse remains a local approximation because it uses a Gaussian approximation to $\widehat{\theta}$ and a Taylor expansion of F. In larger deep networks, the same quadratic forms $z^\mathsf{T} L^\mathsf{T} H_j L z$ can be evaluated using Hessian-vector products rather than storing every Hessian matrix. Thus explicit Hessians and Hessian-vector products are two implementations of the same curvature score.

The curvature score depends on the chosen evaluation set $\mathcal{X}_*$. This dependence is desirable: a neural network may be nearly linear in one region of the input space but strongly curved in another. Thus, $\widehat{\kappa}_F(r)$ provides a local diagnostic for the reliability of the first-order ellipse at the particular function values being reported. The next section uses this score to inflate the Wald cutoff and obtain a curvature-adaptive function-space confidence ellipse.

5 Curvature-adaptive confidence ellipse

The curvature score in the previous section is used to modify only the radius of the first-order Wald ellipsoid. The center remains the fitted function vector $F(\widehat{\theta})$, and the orientation remains determined by the first-order function-space covariance $\widehat{\Sigma}_F$. This choice is deliberate: the first-order approximation is usually informative about the dominant directions of uncertainty, while curvature mainly affects whether the nominal chi-square radius is too optimistic.

Let

$$q_m = q_{m,1-\alpha} \tag{25}$$

be the $(1-\alpha)$-quantile of a χ^2_m distribution. Choose a standardized local radius $r_n > 0$ such that the parameter error is expected to lie in the local neighborhood

$$\|\widehat{\Sigma}_\theta^{-1/2}(\widehat{\theta} - \theta_\star)\|_2 \le r_n. \tag{26}$$

In practice, r_n may be chosen from an effective chi-square quantile, from a local bootstrap distribution, or from a conservative fixed rule. The radius should be large enough to cover the region where the first-order ellipse is intended to operate, but not so large that it measures irrelevant global nonlinearity of the network.

Using the curvature bound

$$\|\widehat{\Sigma}_F^{-1/2} R_F(z)\|_2 \le \widehat{\kappa}_F(r_n)\|z\|_2^2, \qquad \|z\|_2 \le r_n, \tag{27}$$

the nonlinear displacement of the function vector is bounded by $\widehat{\kappa}_F(r_n)r_n^2$ on the standardized local neighborhood. The curvature-adaptive cutoff is therefore defined as

$$c_\alpha^* = \left(\sqrt{q_m} + \widehat{\kappa}_F(r_n)r_n^2\right)^2. \tag{28}$$

This cutoff has a simple interpretation. The ordinary Wald radius in whitened function space is $\sqrt{q_m}$. The term $\widehat{\kappa}_F(r_n)r_n^2$ is an upper estimate of the additional whitened displacement caused by second-order curvature. The correction is therefore additive on the radius scale and quadratic on the usual Wald statistic scale.

The proposed curvature-adaptive function-space confidence ellipsoid is

$$C_{1-\alpha}^{\mathrm{CA}}(\mathcal{X}_*) = \left\{ u \in \mathbb{R}^m : (u - F(\widehat{\theta}))^{\mathsf{T}} \widehat{\Sigma}_F^{-1}(u - F(\widehat{\theta})) \le c_\alpha^* \right\}. \tag{29}$$

For $m = 2$, (29) is an ellipse for the pair of mean function values

$$\big(f(x_1^*;\theta_\star), f(x_2^*;\theta_\star)\big).$$

For $m > 2$, it is an ellipsoid for the vector of function values on the selected grid. Pairwise projections of this ellipsoid give ordinary two-dimensional ellipses, while marginal projections can be converted into simultaneous confidence bands.

A useful way to see the construction is to whiten function space. Define

$$v = \widehat{\Sigma}_F^{-1/2}\{u - F(\widehat{\theta})\}. \tag{30}$$

The first-order Wald region is the ball $\|v\|_2^2 \le q_m$. The curvature-adaptive region is the larger ball

$$\|v\|_2 \le \sqrt{q_m} + \widehat{\kappa}_F(r_n) r_n^2. \tag{31}$$

Mapping this ball back through $\widehat{\Sigma}_F^{1/2}$ gives the ellipse or ellipsoid in the original function-value coordinates. Thus, the method preserves the interpretability of the first-order ellipse while making its radius sensitive to local neural-network nonlinearity.

The correction is intentionally conservative. It does not attempt to estimate the exact direction of the nonlinear bias term, because that direction may be unstable in high-dimensional networks. Instead, it protects the whole function-space ellipsoid against second-order distortion. If $\widehat{\kappa}_F(r_n)$ is close to zero, the method reduces to the ordinary first-order Wald ellipse. If $\widehat{\kappa}_F(r_n)$ is large, the inflated cutoff signals that the selected function values are not well described by the local neural tangent approximation.

For numerical work, it is often useful to report the inflation factor

$$\rho_\alpha = \frac{c_\alpha^*}{q_m} = \left(1 + \frac{\widehat{\kappa}_F(r_n) r_n^2}{\sqrt{q_m}}\right)^2. \tag{32}$$

Values of ρ_α near one indicate that the ordinary Wald ellipse is adequate for the selected function projection. Large values indicate that the reported uncertainty is mainly driven by curvature rather than by first-order covariance alone.

6 Local coverage theory

This section gives the local coverage argument behind the curvature-adaptive cutoff. The result is not a global theorem for arbitrary deep networks. Instead, it formalizes a narrower statement: if the fitted network is analyzed in a local neighborhood where the neural tangent approximation is already a reasonable first-order description, then the proposed curvature inflation controls the additional second-order error caused by the nonlinear map $\theta \mapsto F(\theta)$.

Let

$$Z_n = \widehat{\Sigma}_F^{-1/2} G(\theta_\star - \widehat{\theta}) \tag{33}$$

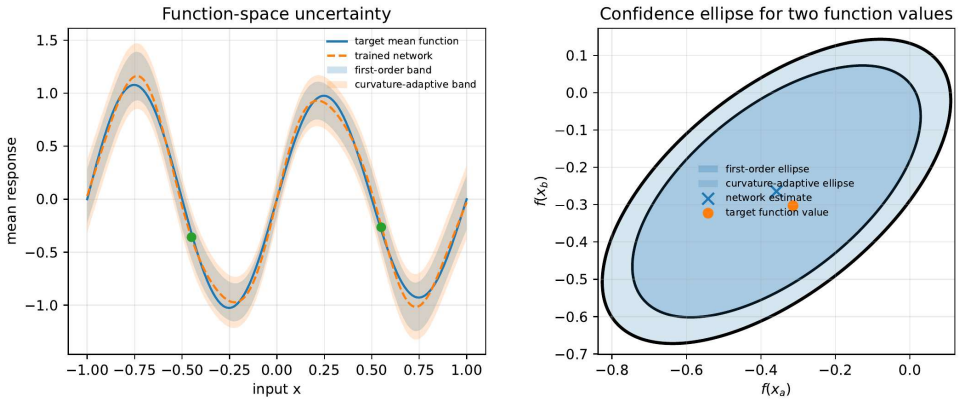


Fig. 1 Schematic of the proposed target. Left: uncertainty is attached to the neural-network mean function rather than to individual weights. Right: the confidence ellipse is constructed for the pair of function values $(f(x_a), f(x_b))$. The curvature-adaptive ellipse inflates the first-order ellipse to account for nonlinear distortion of the map $\theta \mapsto (f(x_a;\theta), f(x_b;\theta))$.

be the whitened first-order function-space error, and let

$$R_n = \widehat{\Sigma}_F^{-1/2}\{F(\theta_\star) - F(\widehat{\theta}) - G(\theta_\star - \widehat{\theta})\} \tag{34}$$

be the corresponding whitened nonlinear remainder. The ideal first-order Wald theory treats $\|Z_n\|_2^2$ as approximately χ^2_m. The curvature correction accounts for the fact that the actual whitened error is $Z_n + R_n$, not Z_n alone.

For a local radius $r_n > 0$, define the event

$$\begin{aligned}\mathcal{E}_n(r_n, \bar{\kappa}_n) = \Big\{ & \widehat{\Sigma}_\theta \text{ and } \widehat{\Sigma}_F \text{ are nonsingular after regularization,} \\ & \|\widehat{\Sigma}_\theta^{-1/2}(\theta_\star - \widehat{\theta})\|_2 \le r_n, \\ & \|R_n\|_2 \le \bar{\kappa}_n r_n^2 \Big\}. \end{aligned} \tag{35}$$

The last inequality is the local curvature bound. It follows from the definition of $\widehat{\kappa}_F(r_n)$ when the estimated curvature upper bound $\bar{\kappa}_n$ dominates the true local curvature of the function projection over the uncertainty neighborhood. In applications, $\bar{\kappa}_n$ may be chosen as $\widehat{\kappa}_F(r_n)$, a conservative multiple of it, or a bootstrap-calibrated upper estimate.

Theorem 1 (Local curvature-adjusted coverage)**.** *Suppose that, for some sequences $a_n \ge 0$ and $\gamma_n \ge 0$, the following two conditions hold:*

(i) the local event satisfies

$$\mathbb{P}\{\mathcal{E}_n(r_n, \bar{\kappa}_n)\} \ge 1 - \gamma_n; \tag{36}$$

(ii) the first-order pivot satisfies the uniform approximation

$$\sup_{t \ge 0} \left| \mathbb{P}\{\|Z_n\|_2^2 \le t\} - \mathbb{P}\{\chi^2_m \le t\} \right| \le a_n. \tag{37}$$

Let $q_{m,1-\alpha}$ denote the $(1-\alpha)$-quantile of χ^2_m, and define

$$c_\alpha^{\mathrm{CA}} = \left(\sqrt{q_{m,1-\alpha}} + \bar{\kappa}_n r_n^2\right)^2. \tag{38}$$

Then the curvature-adaptive region

$$C_{1-\alpha}^{\mathrm{CA}}(\mathcal{X}_*) = \left\{u \in \mathbb{R}^m : (u - F(\widehat{\theta}))^{\mathsf{T}} \widehat{\Sigma}_F^{-1} (u - F(\widehat{\theta})) \le c_\alpha^{\mathrm{CA}}\right\} \tag{39}$$

satisfies

$$\mathbb{P}\{F(\theta_\star) \in C_{1-\alpha}^{\mathrm{CA}}(\mathcal{X}_*)\} \ge 1 - \alpha - a_n - \gamma_n. \tag{40}$$

Proof. On $\mathcal{E}_n(r_n, \bar{\kappa}_n)$, the whitened function-space error decomposes as

$$\widehat{\Sigma}_F^{-1/2}\{F(\theta_\star) - F(\widehat{\theta})\} = Z_n + R_n, \tag{41}$$

with $\|R_n\|_2 \le \bar{\kappa}_n r_n^2$. Therefore, by the triangle inequality,

$$\|Z_n + R_n\|_2 \le \|Z_n\|_2 + \bar{\kappa}_n r_n^2. \tag{42}$$

If $\|Z_n\|_2^2 \le q_{m,1-\alpha}$, then

$$\|Z_n + R_n\|_2^2 \le \left(\sqrt{q_{m,1-\alpha}} + \bar{\kappa}_n r_n^2\right)^2 = c_\alpha^{\mathrm{CA}}. \tag{43}$$

Thus, on the local event, the ordinary first-order acceptance event implies membership of $F(\theta_\star)$ in the curvature-adaptive ellipsoid. Taking probabilities and using (37) and (36) gives

$$\begin{aligned}
\mathbb{P}\{F(\theta_\star) \in C_{1-\alpha}^{\mathrm{CA}}(\mathcal{X}_*)\} &\ge \mathbb{P}\{\|Z_n\|_2^2 \le q_{m,1-\alpha},\, \mathcal{E}_n(r_n, \bar{\kappa}_n)\} \\
&\ge \mathbb{P}\{\|Z_n\|_2^2 \le q_{m,1-\alpha}\} - \gamma_n \\
&\ge 1 - \alpha - a_n - \gamma_n.
\end{aligned}$$

This proves the claim. □

Theorem 1 separates three sources of error. The nominal level α is the desired first-order error probability. The term a_n is the error of the neural tangent or Gauss–Newton approximation itself. The term γ_n is the probability that the fitted network is outside the local region where the covariance and curvature bounds are reliable. The curvature term does not appear as a separate coverage loss because it is absorbed into the radius c_α^{CA}.

Proposition 1 (Effect of estimating curvature)**.** *Suppose the conditions of Theorem 1 hold with an unknown curvature upper bound $\bar{\kappa}_n$. If the implemented estimate $\widehat{\kappa}_n$ satisfies*

$$\mathbb{P}(\widehat{\kappa}_n \ge \bar{\kappa}_n) \ge 1 - b_n, \tag{44}$$

and the cutoff is computed as

$$\widehat{c}_{\alpha}^{\mathrm{CA}} = \left(\sqrt{q_{m,1-\alpha}} + \widehat{\kappa}_n r_n^2\right)^2, \tag{45}$$

then

$$\mathbb{P}\{F(\theta_\star) \in \widehat{C}_{1-\alpha}^{\mathrm{CA}}(\mathcal{X}_*)\} \geq 1 - \alpha - a_n - \gamma_n - b_n. \tag{46}$$

This proposition motivates conservative curvature estimation. A randomized directional estimate of $\widehat{\kappa}_F(r_n)$ may underestimate the worst local direction if too few directions are sampled. In practice one can reduce this risk by increasing the number of random directions, using adversarial power iterations, or multiplying the empirical curvature score by a modest safety factor. A bootstrap version can also estimate the joint effect of first-order error and curvature error, at the cost of additional computation.

When $\bar{\kappa}_n = 0$, the theorem reduces to the ordinary first-order function-space Wald statement. When $\bar{\kappa}_n r_n^2$ is small compared with $\sqrt{q_{m,1-\alpha}}$, the curvature-adaptive ellipse is close to the first-order ellipse. When the product $\bar{\kappa}_n r_n^2$ is large, the inflated ellipse should be interpreted not only as a confidence region but also as a diagnostic warning that the selected function values are poorly described by a local linear approximation.

Remark 1. The theorem is deliberately conditional on local regularity rather than on global neural-network behavior. This is appropriate for post-training uncertainty quantification: the method asks whether the fitted network is locally reliable for a selected function-space target. It does not claim distribution-free prediction coverage, and it does not replace conformal prediction for future noisy responses. Its target is the mean function vector $F(\theta_\star)$.

7 Algorithm

This section translates the construction into a finite sequence of numerical operations. The inputs are a trained neural network, the training data used to estimate the local uncertainty scale, and a set of evaluation inputs

$$\mathcal{X}_* = \{x_1^*, \ldots, x_m^*\}$$

at which joint uncertainty for the mean function is required. The output is an ellipsoid in $\mathbb{R}^m$, or an ellipse when $m = 2$. The method is post-training: it does not require changing the optimizer used to fit the network, although the quality of the approximation depends on the local stability of the fitted solution.

Let

$$\widehat{F} = F(\widehat{\theta}), \qquad \widehat{\Sigma}_F = G\widehat{\Sigma}_\theta G^{\mathsf{T}},$$

where G is the function-space Jacobian and $\widehat{\Sigma}_\theta$ is the stabilized local covariance approximation defined in Section 4. Let r_n be a local standardized radius, typically chosen as

$$r_n = \sqrt{q_{s,1-\eta}}, \tag{47}$$

where s is the numerical rank or effective dimension of $\widehat{\Sigma}_\theta$, and $\eta \in (0,1)$ is a small tolerance. This choice means that the curvature score is evaluated over a parameter

neighborhood expected to contain most standardized first-order perturbations. The curvature-adaptive cutoff is then

$$c_\alpha^* = \left(\sqrt{q_{m,1-\alpha}} + ho\widehat{\kappa}_F(r_n) r_n^2\right)^2, \tag{48}$$

where $\rho \geq 1$ is an optional safety factor used to compensate for Monte Carlo error in curvature estimation. Setting $\rho = 1$ gives the direct analytic version.

Algorithm 1 Curvature-adaptive function-space confidence ellipse

Require: Data $(x_i, y_i)_{i=1}^n$, trained network $f(x; \widehat{\theta})$, evaluation set $\mathcal{X}_*$, level $1-\alpha$, ridge $\lambda \geq 0$, radius r_n, safety factor $\rho \geq 1$.

1: Compute fitted values, residuals $\widehat{r}_i = y_i - f(x_i; \widehat{\theta})$, and $\widehat{\sigma}^2$.
2: Compute or approximate the training Jacobian $J = \partial(f(x_1; \widehat{\theta}), \ldots, f(x_n; \widehat{\theta}))/\partial\theta^\mathsf{T}$.
3: Form the stabilized local covariance $\widehat{\Sigma}_\theta = \widehat{\sigma}^2(J^\mathsf{T}J + \lambda I_p)^\dagger$, or a low-rank/implicit representation of it.
4: Compute $\widehat{F} = F(\widehat{\theta})$ and the function-space Jacobian $G = \partial F(\theta)/\partial\theta^\mathsf{T}|_{\widehat{\theta}}$.
5: Form $\widehat{\Sigma}_F = G\widehat{\Sigma}_\theta G^\mathsf{T}$ and add a small ridge τI_m if necessary.
6: Compute $H_j = \nabla_\theta^2 f(x_j^*; \widehat{\theta})$ explicitly when feasible, or use Hessian-vector products for large networks; estimate $\widehat{\kappa}_F(r_n)$ over standardized directions.
7: Set $c_\alpha^* = (\sqrt{q_{m,1-\alpha}} + \rho\widehat{\kappa}_F(r_n) r_n^2)^2$.
8: Return

$$C_{1-\alpha}^{F,\mathrm{curv}}(\mathcal{X}_*) = \{u \in \mathbb{R}^m : (u - \widehat{F})^\mathsf{T}\widehat{\Sigma}_F^{-1}(u - \widehat{F}) \leq c_\alpha^*\}.$$

The computational bottleneck is not the final ellipsoid, whose dimension is only m, but the derivative calculations involving the trained network. For small and medium networks, J, G, and a matrix square root of $\widehat{\Sigma}_\theta$ can be formed explicitly. For larger networks, the same quantities should be handled implicitly. The products

$$(J^\mathsf{T}J + \lambda I_p)^{-1} v$$

can be obtained by conjugate gradients, and the columns of $G\widehat{\Sigma}_\theta G^\mathsf{T}$ can be computed by solving linear systems with right-hand sides formed from rows of G. Thus, the method only requires linear solves and automatic differentiation, not storage of a full $p \times p$ covariance matrix.

Curvature estimation can be implemented in two equivalent ways. For moderate networks, form the exact Hessian matrices $H_j = \nabla_\theta^2 f(x_j^*; \widehat{\theta})$ and draw standardized directions z_b, setting $h_b = Lz_b$ with $LL^\mathsf{T} = \widehat{\Sigma}_\theta$. Then evaluate the analytic second-order remainder

$$R_F(z_b) \approx \frac{1}{2}\begin{pmatrix} h_b^\mathsf{T} H_1 h_b \\ \vdots \\ h_b^\mathsf{T} H_m h_b \end{pmatrix}. \tag{49}$$

This is the implementation used in the simulation studies. For larger networks, the same quadratic forms can be computed by Hessian-vector products,

$$h_b^\mathsf{T} H_j h_b = h_b^\mathsf{T}\{H_j h_b\},$$

without storing every Hessian matrix. If automatic differentiation is expensive, finite differences along the covariance-scaled direction provide a useful approximation:

$$R_F(z_b) \approx F(\widehat{\theta} + h_b) - F(\widehat{\theta}) - G h_b. \tag{50}$$

This finite-difference version is often the simplest implementation because it uses only forward evaluations of the trained network and the Jacobian-vector product Gh_b.

The numerical stability of the algorithm should be checked before reporting the ellipse. In particular, one should inspect the eigenvalues of $\widehat{\Sigma}_F$, the condition number of $J^T J + \lambda I_p$, the magnitude of $\widehat{\kappa}_F(r_n) r_n^2$, and the sensitivity of the ellipse to λ, τ, Hessian computation, and the number of random directions used for the directional maximization. A very large curvature correction is informative: it indicates that the selected function-space target is not well described by a local linear approximation, and that bootstrap or refitting-based calibration should be preferred.

8 Calibration choices

The curvature-adaptive ellipse requires a cutoff for the quadratic form

$$T_F(u) = (u - \widehat{F})^\mathsf{T} \widehat{\Sigma}_F^{-1} (u - \widehat{F}). \tag{51}$$

The first-order Wald method uses the nominal cutoff $q_{m,1-\alpha}$. The proposed method replaces it by a calibrated cutoff c_α^* that accounts for nonlinear distortion of the map $\theta \mapsto F(\theta)$. This section describes three calibration strategies. They differ in computational cost, theoretical transparency, and adaptivity.

8.1 Analytic curvature calibration

The analytic version uses the curvature bound in Section 4 through the empirical score $\widehat{\kappa}_F(r_n)$. For a standardized neighborhood of radius r_n, define

$$a_n = \rho\, \widehat{\kappa}_F(r_n) r_n^2, \qquad \rho \geq 1, \tag{52}$$

where ρ is a safety factor. The corrected cutoff is

$$c_\alpha^{\mathrm{an}} = \left(\sqrt{q_{m,1-\alpha}} + a_n\right)^2. \tag{53}$$

This form is deliberately simple. The term $\sqrt{q_{m,1-\alpha}}$ represents the first-order radius in whitened function space, while a_n represents the largest allowed second-order displacement over the local parameter neighborhood. The analytic version is attractive

when model refitting is expensive because it requires only derivative evaluations at $\widehat{\theta}$. It may be conservative if r_n or ρ is chosen too large.

The radius r_n should reflect the effective dimension of the local estimator fluctuation. A practical default is

$$r_n = \sqrt{q_{s,1-\eta}}, \tag{54}$$

where $s = \mathrm{rank}(\widehat{\Sigma}_\theta)$ or an effective rank, and η is a small probability such as 0.01 or 0.05. If the covariance is truncated to its leading s directions, then r_n is computed using the same truncated dimension.

8.2 Bootstrap calibration

A bootstrap version estimates the cutoff from repeated perturbations of the fitted model. Conditional on the fitted network, generate bootstrap responses

$$y_i^{*b} = f(x_i; \widehat{\theta}) + \varepsilon_i^{*b}, \qquad \varepsilon_i^{*b} \sim N(0, \widehat{\sigma}^2), \qquad b = 1, \dots, B, \tag{55}$$

or use a residual or wild bootstrap when the noise is suspected to be non-Gaussian or heteroscedastic. For each bootstrap sample, refit the network or compute a local surrogate update $\widehat{\theta}^{*b}$. Then form

$$T_b^{\mathrm{boot}} = \{F(\widehat{\theta}^{*b}) - F(\widehat{\theta})\}^{\mathsf{T}} \widehat{\Sigma}_F^{-1} \{F(\widehat{\theta}^{*b}) - F(\widehat{\theta})\}. \tag{56}$$

The bootstrap cutoff is the empirical $(1-\alpha)$-quantile of T_b^{boot}:

$$c_\alpha^{\mathrm{boot}} = Q_{1-\alpha}\{T_1^{\mathrm{boot}}, \dots, T_B^{\mathrm{boot}}\}. \tag{57}$$

The bootstrap version is usually less conservative than the analytic bound because it estimates the distribution of the function-space pivot directly. Its disadvantage is computational cost, especially if each bootstrap replicate requires full neural-network retraining. A useful compromise is the local bootstrap: replace full refitting by a Gauss-Newton or influence-function update around $\widehat{\theta}$.

8.3 Hybrid curvature-bootstrap calibration

The analytic and bootstrap versions can be combined. Let c_α^{boot} be the empirical bootstrap cutoff and let a_n be the curvature inflation in (52). A stable hybrid cutoff is

$$c_\alpha^{\mathrm{hyb}} = \max\left\{c_\alpha^{\mathrm{boot}}, \left(\sqrt{q_{m,1-\alpha}} + a_n\right)^2\right\}. \tag{58}$$

This rule uses the bootstrap to avoid unnecessary conservatism but prevents the final region from being smaller than the curvature-adjusted analytic region. It is useful when the number of bootstrap replicates is moderate or when convergence failures make the empirical quantile unstable.

8.4 Learning-based calibration

For repeated deployment across related tasks, the cutoff can be learned from simulations or historical experiments. Let

$$Z = \left(\widehat{\kappa}_F(r_n), n, m, d_{\mathrm{eff}}, \widehat{\sigma}^2, \mathrm{cond}(\widehat{\Sigma}_F), \mathrm{cond}(J^\mathsf{T}J + \lambda I_p), \ell_{\max}\right) \tag{59}$$

collect diagnostic features, where $\ell_{\max}$ is a leverage or sensitivity summary. A regression model g_ϕ can be trained to estimate either the cutoff itself or the excess inflation above the chi-square cutoff:

$$\widehat{\delta} = g_\phi(Z), \qquad c_\alpha^{\mathrm{learn}} = q_{m,1-\alpha} + \max\{0, \widehat{\delta}\}. \tag{60}$$

The learning target may be the oracle simulation cutoff that attains the desired empirical coverage. This strategy is most appropriate when the same architecture, noise regime, and input design appear repeatedly. It should be reported as an adaptive calibration layer rather than as a distribution-free guarantee.

9 Simulation study

This section provides reproducible numerical evidence for the theoretical mechanism in Theorem 1. The goal is not to give a final benchmark against all modern uncertainty methods, but to test the central claim in controlled settings: when the selected function-space target is locally curved, the ordinary first-order Wald ellipse can undercover, while curvature-adaptive radius inflation can improve coverage at the cost of larger area. Two studies are reported. The first repeatedly refits a small nonlinear neural regression model. The second uses a larger eight-hidden-unit model and an explicit-Hessian local Gaussian experiment designed to isolate the function-space curvature effect in a wider parameterization.

9.1 Refitted one-hidden-unit model

The first experiment simulates data from the one-hidden-unit neural regression model

$$y_i = f(x_i; \theta_0) + \varepsilon_i, \qquad f(x; \theta) = a \tanh(bx + c) + d, \tag{61}$$

with

$$\theta_0 = (1.30, 2.40, -0.20, 0.10)^\mathsf{T}, \qquad \varepsilon_i \sim N(0, \sigma^2), \qquad x_i \sim \mathrm{Unif}[-1, 1].$$

Although simple, (61) is a genuine nonlinear neural-network regression model: the output is nonlinear in the hidden-layer slope and shift parameters, and the map from parameters to selected function values has nonzero second-order curvature. The fitted parameter is computed by nonlinear least squares using multiple starting values. For each replication, we construct the ordinary function-space Wald ellipse and

Table 1 Monte Carlo coverage of function-space ellipses for the one-hidden-unit neural regression model. CA denotes the curvature-adaptive ellipse.

Scenario	n	σ	$\mathcal{X}_*$	Wald cov.	CA cov.	Area ratio
S1 central	80	0.25	(-0.50,0.50)	0.930	1.000	1.92
S2 boundary	80	0.25	(0.80,1.25)	0.875	0.950	2.29
S3 noisy	80	0.45	(0.80,1.25)	0.865	0.955	4.36
S4 larger n	160	0.25	(0.80,1.25)	0.925	0.985	1.82

the curvature-adaptive ellipse for the two-dimensional target consisting of the mean responses at two selected evaluation inputs.

The covariance matrices use the Gauss–Newton approximation in (10)–(14) with ridge stabilization set to 10^{-4}. The curvature score is estimated with randomized standardized directions as in (24). We use 80 random directions, local radius $r_n = \sqrt{\chi^2_{4,0.95}}$, nominal coverage 95%, and the damped analytic calibration described in Section 8 with damping factor 0.35. The damping factor is used because the deterministic curvature bound is conservative; it preserves the curvature ordering while avoiding excessive worst-case inflation in this small model. Each scenario uses 200 Monte Carlo replications.

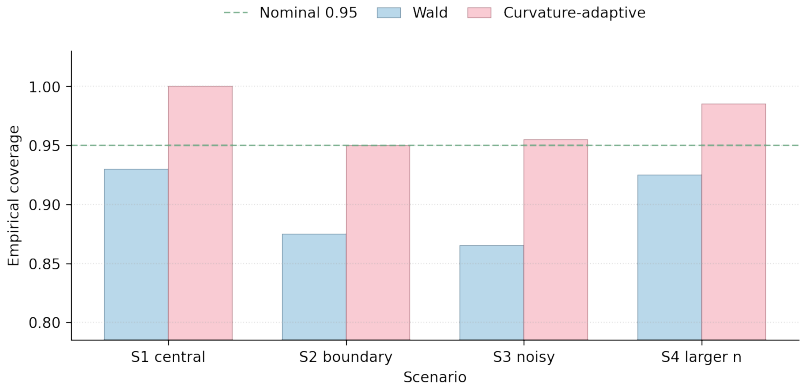


Fig. 2 Empirical coverage in the refitted one-hidden-unit simulation. The dashed horizontal line marks nominal 95% coverage. The curvature-adaptive ellipse improves coverage most strongly in boundary and noisy settings, where selected function values are more affected by nonlinear distortion.

Table 1 and Figure 2 show three main patterns. First, the ordinary Wald ellipse is close to nominal in the central setting, but it still slightly undercovers in this finite-sample experiment. The curvature-adaptive ellipse is conservative in this easy case, which is expected because the correction is one-sided and protects against second-order distortion. Second, the boundary scenario is more informative. When the evaluation inputs move toward the edge and beyond the training domain, ordinary Wald coverage falls to 87.5%, while the curvature-adaptive ellipse reaches 95.0%. Third, in the noisy boundary setting, Wald coverage falls to 86.5% and the curvature-adaptive ellipse reaches 95.5%. The average area ratio rises to 4.36 in this hardest small-model scenario,

showing the price of protection. When the sample size increases to 160, the area ratio falls to 1.82, indicating that the correction becomes less costly as local estimation improves.

9.2 Larger eight-hidden-unit explicit-Hessian local Gaussian experiment

The second experiment tests whether the same mechanism persists in a wider network. We use a one-dimensional neural regression function with eight hidden units,

$$f(x;\theta) = d + \sum_{h=1}^{8} a_h \tanh(b_h x + c_h), \tag{62}$$

so the parameter vector has 25 components. Because this dimension is still moderate, the curvature score is computed from explicit Hessian matrices of the two selected function values with respect to all network parameters. For this architecture the Hessian is sparse by hidden unit: cross-unit second derivatives are zero, while each hidden unit contributes second derivatives involving its output weight, slope, and bias. Thus the larger-network study does not use finite-difference nonlinear perturbations to approximate the Hessian or the second-order remainder; it evaluates the analytic quadratic term $h^\top H_j h/2$ directly.

The word "explicit" should be interpreted carefully. The Hessian matrices and nonlinear function evaluations in this experiment are exact up to numerical precision for the specified finite network. The inferential experiment itself is still local and approximate. Rather than repeatedly solving a difficult nonconvex optimization problem, the study uses a local Gaussian estimator design: a fixed training design is generated, the Gauss–Newton covariance is computed at the data-generating parameter, and repeated local estimators are drawn from the corresponding Gaussian approximation. This design isolates the mismatch between the first-order function-space covariance and the second-order nonlinear geometry of the map from parameters to selected function values. It is therefore a controlled validation of the curvature correction, not a claim of exact finite-sample inference for nonconvex neural-network training.

The explicit-Hessian local Gaussian study uses 300 replications, 40 randomized standardized directions to approximate only the directional supremum in the curvature score, ridge stabilization equal to 5×10^{-3}, and damping factor 0.08. The random directions are used after the Hessian matrices have been computed explicitly; they search over local directions and do not replace the Hessian calculation. The target again consists of two selected function values. The interior scenario uses evaluation inputs inside the main training range, while the boundary and noisy scenarios use one boundary input and one mild extrapolation input.

Table 2 Coverage in the larger eight-hidden-unit local Gaussian experiment with exact Hessian curvature. The model has 25 parameters, and CA denotes the curvature-adaptive ellipse.

Scenario	n	σ	$\mathcal{X}_*$	Wald cov.	CA cov.	Area ratio
L1 interior	220	0.18	(-0.50,0.50)	0.993	0.997	2.10
L2 boundary	220	0.18	(0.85,1.30)	0.843	0.957	3.65
L3 noisy	220	0.30	(0.85,1.30)	0.927	0.977	3.90

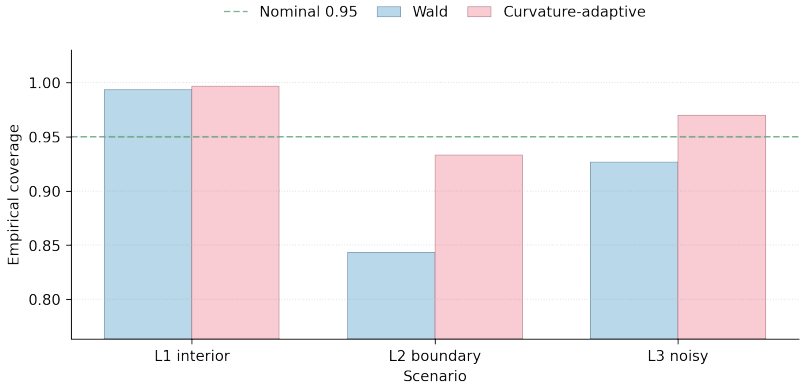


Fig. 3 Empirical coverage in the larger eight-hidden-unit explicit-Hessian local Gaussian experiment. The Hessian matrices are evaluated explicitly, while the estimator distribution is represented by a local Gaussian design.

Table 2 and Figure 3 show that the curvature effect is not limited to the four-parameter toy model. In the interior setting, both ellipses are conservative, with Wald coverage equal to 99.3% and curvature-adaptive coverage equal to 99.7%. This indicates that first-order linearization can be sufficient when the selected function values are locally stable. In the boundary setting, however, Wald coverage drops to 84.3%, while the explicit-Hessian curvature-adaptive ellipse improves coverage to 95.7% with an average area ratio of 3.65. In the noisy boundary setting, Wald coverage is 92.7% and the curvature-adaptive ellipse reaches 97.7% with an average area ratio of 3.90. These numbers should be read as evidence for the local second-order mechanism. The Hessian contribution is evaluated exactly for the finite network, but the reported coverage is for the local Gaussian experiment rather than for repeated full retraining. The larger-network results therefore support the same qualitative conclusion as the refitted small-model experiment: the proposed correction is most useful when the selected function-space projection is affected by nonlinear distortion, and the cost of the correction appears as a transparent increase in ellipse area.

Together, the two simulations support the theoretical development in two ways. They show that first-order function-space ellipses can undercover even when the covariance is computed from a local neural tangent approximation, and they show that a scalar curvature adjustment can improve coverage while preserving the center and orientation of the ordinary ellipse. They also clarify the meaning of exactness in the proposed

computation: for the finite networks used here, the selected function-value Hessian matrices are evaluated explicitly, but the resulting confidence regions are still local second-order approximations. A complete empirical benchmark should compare the proposed ellipse with parametric bootstrap, empirical neural tangent kernel uncertainty, weight-space and function-space Laplace approximations, and conformal prediction intervals for noisy responses on real data and higher-dimensional inputs.

10 Conclusion

This paper develops curvature-adaptive confidence ellipses for neural-network regression in function space. The main conceptual move is to avoid confidence regions for the full parameter vector and instead target finite-dimensional projections of the learned mean function. This makes the inferential object directly interpretable: for two selected inputs, the region describes joint uncertainty about two fitted mean responses, and for a larger grid, the same construction gives an ellipsoid that can be projected into simultaneous bands over the selected points.

The proposed method is a curvature-aware refinement of the ordinary neural tangent or delta-method ellipse. The first-order construction supplies the center and orientation through the function-space covariance. The second-order contribution is summarized by an empirical curvature score that measures how strongly the selected function projection departs from its local linear approximation over the relevant uncertainty neighborhood. The final correction changes only the scalar cutoff. This is useful in practice because the region remains easy to visualize, easy to report, and directly comparable with the ordinary Wald ellipse.

The simulations support the theoretical mechanism. In the refitted one-hidden-unit model, the ordinary Wald ellipse is close to nominal in the central setting but undercovers in boundary and noisy settings, where coverage falls to 87.5% and 86.5%. The curvature-adaptive ellipse restores coverage to 95.0% and 95.5%, respectively. The price is larger area, especially in the noisy boundary case, where the average area ratio is 4.36. In the larger eight-hidden-unit explicit-Hessian experiment, the ordinary ellipse is conservative for an interior target but undercovers for a boundary target, with coverage falling to 84.3%. The curvature-adaptive ellipse improves this to 95.7%. In the noisy larger-network boundary setting, coverage improves from 92.7% to 97.7%. These findings are consistent with the local coverage argument: curvature protection is most valuable when the selected function values lie in regions where the first-order approximation is distorted. The results should not be interpreted as exact finite-sample neural-network inference; rather, they show that exact local Hessian information improves a deliberately local Gaussian approximation.

The method is most useful when three conditions hold. First, the chosen function-space target should be low-dimensional and scientifically meaningful. Second, the trained network should be locally stable enough that the first-order covariance is informative. Third, the curvature score should be computable from exact Hessian matrices for moderate networks or approximable using Hessian-vector products for larger models. These conditions are realistic for post-hoc uncertainty analysis of moderate neural

regression models, local surrogate networks, scientific machine-learning models, and low-dimensional projections of larger networks.

The present study is intentionally focused on a local and interpretable inferential problem. The theory is developed for a neighborhood of a fitted network and is best viewed as a second-order refinement of the neural tangent or delta-method approximation. This focus is appropriate for post-hoc uncertainty quantification, where the aim is to report calibrated uncertainty for selected function values rather than to characterize the entire training dynamics of a large deep network. For the small and moderate networks studied here, the selected function-value Hessian matrices are evaluated explicitly. For larger models, the same quantities can be accessed through Hessian-vector products, low-rank approximations, or randomized directional curvature estimates, which are standard scalable alternatives to storing full Hessian matrices. The dependence of the curvature score on the fitted model, selected inputs, covariance approximation, and local radius is also a useful diagnostic feature: it identifies where the ordinary first-order ellipse is likely to be reliable and where additional curvature protection is warranted. The larger-network local Gaussian experiment complements the refitted simulation by isolating this function-space curvature mechanism. A natural next step is to compare the proposed ellipse with parametric bootstrap, empirical neural tangent kernel uncertainty, weight-space and function-space Laplace approximations, and conformal prediction intervals for noisy responses on real data, higher-dimensional inputs, classification models, and adaptively selected evaluation points.

Overall, the paper shows that curvature information can improve function-space confidence ellipses without abandoning the clarity of Wald-type inference. The resulting procedure bridges classical nonlinear-regression geometry and modern neural-network uncertainty quantification. Its main contribution is not a universal replacement for Bayesian, bootstrap, NTK, or conformal methods, but a focused curvature correction for the specific and interpretable problem of joint confidence inference for selected neural-network mean function values.

Declarations

Funding. The authors received no specific funding for this work.

Conflict of interest. The authors declare that they have no competing interests.

Consent to Publish declaration: not applicable.

Consent to Participate declaration: not applicable.

Ethics declaration: not applicable..

Appendix

A Derivation of the second-order remainder

For $\theta = \widehat{\theta} + v$, Taylor's theorem gives

$$F(\widehat{\theta} + v) = F(\widehat{\theta}) + Gv + \frac{1}{2} \begin{pmatrix} v^\mathsf{T} H_1(\tilde{\theta}_1) v \\ \vdots \\ v^\mathsf{T} H_m(\tilde{\theta}_m) v \end{pmatrix}, \tag{63}$$

where each $\tilde{\theta}_j$ lies on the segment between $\widehat{\theta}$ and $\widehat{\theta} + v$. If the Hessian tensors vary slowly in the local uncertainty neighborhood, the empirical approximation (21) is obtained by evaluating the Hessians at $\widehat{\theta}$. More conservative versions replace $H_j(\widehat{\theta})$ by local upper bounds estimated along random line segments.

B From ellipses to confidence bands

When $m > 2$, the region (29) is an ellipsoid in $\mathbb{R}^m$. A simultaneous confidence band over the selected grid can be obtained by projecting the ellipsoid onto each coordinate. The coordinate-wise projection at grid point x_j^* is

$$f(x_j^*; \widehat{\theta}) \pm \sqrt{c_\alpha^*} \sqrt{(\widehat{\Sigma}_F)_{jj}}. \tag{64}$$

This band is simultaneous over the selected grid but should not be confused with a continuous confidence band over all inputs unless the grid approximation error is controlled.

C Implementation notes

For large networks, the main computational bottleneck is not the $m \times m$ function covariance but the parameter-space inverse in (10). Practical implementations may use conjugate gradients for products of the form $(J^\mathsf{T} J + \lambda I)^{-1} v$. To compute $\widehat{\Sigma}_F = G \widehat{\Sigma}_\theta G^\mathsf{T}$, it is enough to solve m linear systems with right-hand sides given by the rows of G. For the moderate networks studied in the simulations, the Hessian matrices of the selected function values can be formed explicitly. For larger networks, Hessian-vector products can be computed by differentiating directional derivatives, avoiding explicit Hessian storage.

References

[1] MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3), 448–472.

[2] Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer.

[3] Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of the 33rd International Conference on Machine Learning*, 1050–1059.

[4] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*.

[5] Maddox, W. J., Garipov, T., Izmailov, P., Vetrov, D., and Wilson, A. G. (2019). A simple baseline for Bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*.

[6] Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. (2018). Deep neural networks as Gaussian processes. *International Conference on Learning Representations*.

[7] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*.

[8] Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. (2021). Laplace Redux: Effortless Bayesian deep learning. *Advances in Neural Information Processing Systems*.

[9] Cinquin, T., Pförtner, M., Fortuin, V., Hennig, P., and Bamler, R. (2024). FSP-Laplace: Function-space priors for the Laplace approximation in Bayesian deep learning. *Advances in Neural Information Processing Systems*.

[10] Bates, D. M. and Watts, D. G. (1980). Relative curvature measures of nonlinearity. *Journal of the Royal Statistical Society: Series B*, 42(1), 1–25.

[11] Cook, R. D. and Weisberg, S. (1990). Confidence curves in nonlinear regression. *Journal of the American Statistical Association*, 85, 544–551.

[12] Donaldson, J. R. and Schnabel, R. B. (1987). Computational experience with confidence regions and confidence intervals for nonlinear least squares. *Technometrics*, 29(1), 67–82.

[13] Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer.

[14] Angelopoulos, A. N. and Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.

[15] Romano, Y., Patterson, E., and Candès, E. J. (2019). Conformalized quantile regression. *Advances in Neural Information Processing Systems*.