# Exact Gaussian Moment Matching for Residual Networks: a Second-Order Method

**Simon Kuang and Xinfan Lin**
Department of Mechanical and Aerospace Engineering
University of California, Davis
Davis, CA 95616
{slku, lxflin}@ucdavis.edu

## Abstract

We study the problem of propagating the mean and covariance of a general multivariate Gaussian distribution through a deep (residual) neural network using layer-by-layer moment matching. We close a longstanding gap by deriving exact moment matching for the probit, GeLU, ReLU (as a limit of GeLU), Heaviside (as a limit of probit), and sine activation functions; for both feedforward and generalized residual layers. On large-variance random networks, median KL-divergence improvements over common baselines range from tens to thousands of times. On a variational Bayes neural network, we show that our method improves upon a state-of-the-art deterministic inference method by an order of magnitude in KL divergence from Monte Carlo ground truth. We also give a smooth-distance error bound showing that, under regularity assumptions, moment matching removes the leading low-variance errors and propagates higher-order local accuracy through the layers of a network.

## 1 Introduction

Given a Gaussian input distribution, what is the output distribution of a neural network? This fundamental problem arises in robustness analysis of a network under typical perturbations [Wright et al., 2024], making predictions on noisy data using a neural network trained on noiseless data [Bibi et al., 2018], computing the loss function of a variational Bayes neural network [Frey and Hinton, 1999, Wu et al., 2019, Petersen et al., 2024, Wright et al., 2024, Rui Li and Trapp, 2025], and understanding the wide limit of randomly initialized deep networks [He et al., 2015]. Concurrent work views this problem as a goalpost for mechanistic understanding of neural networks, with a view towards risk estimation in AI safety [Wu et al., 2026].

The baseline method is Monte Carlo, whose sample mean and covariance are guaranteed to be asymptotically exact in the limit of infinitely many samples. But Monte Carlo is computationally expensive and scales poorly with input dimension [Tang, 2024]. It is for this reason that all of the works cited above apply deterministic approximations to the mean and covariance of neural network outputs. Despite the fact that exactly evaluating a network's pushforward distribution is NP-hard, moment matching—which approximates the distribution of a hidden layer as Gaussian in order to derive a closed form for the moments of the activation function—has proven practically effective.

The main subtasks of moment matching are: (i) compute the mean vector $\mu$ and variance-covariance matrix $\Sigma$ of any hidden layer whose pre-activations have an arbitrary multivariate Gaussian distribution, and (ii) certify that the true post-activation distribution is approximately Gaussian.

Literature on moment matching in neural networks has responded to objective (i) by restricting the mean and covariance of the input distribution. Thus, the methodological and theoretical states of the art are inexact on networks where the hidden layer has more than one neuron, and omit the linear

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

| Assumption | References |
|---|---|
| $\Sigma$ is small (linearized) | Titensky et al. [2018], Nagel and Huber [2022]<br>Petersen et al. [2024], Jungmann et al. [2025]<br>Bergna et al. [2025], Rui Li and Trapp [2025] |
| $\Sigma$ is small (unscented) | Astudillo and Neto [2011], Abdelaziz et al. [2015] |
| $\Sigma$ is diagonal | Abdelaziz et al. [2015], Huber [2020]<br>Goulet et al. [2021], Rui Li and Trapp [2025]<br>Wagner et al. [2022], Akgül et al. [2025] |
| $\mu = 0$ | Cho and Saul [2009], Bibi et al. [2018]<br>NNGP literature [Han et al., 2022, Table 1] |
| $\mu \to \infty$ | Wu et al. [2019] |
| **no assumptions** | Wright et al. [2024]<br>this paper |

Table 1.1: Comparison of assumptions imposed on Gaussian approximations of neural network layers with input $\mathcal{N}(\mu, \Sigma)$.

| Activation function | References |
|---|---|
| piecewise linear | Cho and Saul [2009], Bibi et al. [2018]<br>Huber [2020], Wright et al. [2024]<br>Akgül et al. [2025] |
| logistic ($\approx$ piecewise exponential) | Astudillo and Neto [2011], Abdelaziz et al. [2015] |
| logistic ($\approx \Phi$) | Huber [2020] |
| Heaviside | Wu et al. [2019], Wright et al. [2024] |
| GeLU | Wright et al. [2024] |
| (ReLU, Heaviside) **+ affine** | Wu et al. [2019] |
| ($\sin$, $\Phi$, GeLU, ReLU, Heaviside) **+ affine** | this paper (**exact**) |

Table 1.2: Activation functions for which moment propagation has been approximated. Note two distinct approaches to approximating the logistic function.

(residual) bypass architecture altogether. Given that many neurons are needed to learn arbitrary nonlinear functions, the prevailing assumptions severely limit the scope of moment matching.

In this work, we address both challenges. We derive the exact closed-form, analytical expressions for the means and covariances of both feedforward and residual layers, and certify that on multi-layer networks, moment matching is provably asymptotically more accurate than the delta method.

**Contribution** For an unrestricted multivariate Gaussian input, we derive closed-form mean and covariance matching for feedforward and residual layers with probit, GeLU, ReLU, Heaviside, and sine activation functions. The derivations rely on novel techniques involving integral and differential identities of the Gaussian function, resolving (i). We then prove that exact moment matching cancels the leading terms in a smooth fourth-order probability distance, and that the resulting higher-order local accuracy composes across layers, resolving (ii).

**Related work** Existing results on distribution propagation restrict the parameters of the Gaussian input distribution (Table 1.1). For certain activation functions including ReLU and GeLU, the mean and the diagonal of the covariance matrix can be computed explicitly, but (prior to this paper) there is no closed-form known for off-diagonal covariances of a hidden layer. Some works take the mean-field assumption by setting them to zero [Huber, 2020, Goulet et al., 2021, Wagner et al., 2022, Rui Li and Trapp, 2025, Bergna et al., 2025, Akgül et al., 2025, Rui Li and Trapp, 2025]. Bibi et al. [2018] uses an analytical approximation around zero mean, and Wu et al. [2019] uses an analytical approximation around infinite mean. For the logistic activation function $\sigma(x) = (1 + e^{-x})^{-1}$; Astudillo and Neto [2011], Abdelaziz et al. [2015], and Huber [2020] approximate $\sigma$ with another function having closed-form Gaussian moments, such as a piecewise exponential function or a rescaled Normal CDF $\Phi$. Table 1.2 catalogs the literature on moment approximations for activation

functions. We exemplify the failure modes of the above methods by giving single-hidden-layer counterexamples in §3.

In these prior works, the primary reason for taking the distributional restrictions is to get around the apparent intractability of $\mathbb{R}^2$ improper integrals which arise in moment matching. For example, Wu et al. [2019] expands the covariance to second order around degenerate cases $\mu = 0$ or $\mu = \infty$. The $\mu = 0$ case, frequently encountered in neural network Gaussian Process (NNGP) literature [Han et al., 2022], allows the simplification of two-dimensional integrals using polar coordinates. For a general activation function, Wright et al. [2024] uses a Fourier transform to derive the exact mean and covariance matrix of the output as a formal power series in inter-neuron correlation $\rho \approx 0$. The series, however, must be truncated, as each coefficient needs to be derived algebraically by hand. In this paper, we eschew the $\mathbb{R}^2$ setup altogether by working with probabilistic properties of Gaussian measure, such as the Stein's integration-by-parts identity and the Gaussian ODE $\phi'(x) = -x\phi(x)$.

The closest theoretical result is Petersen et al. [2024], which establishes optimal distributional approximation for a single ReLU neuron. In contrast, our theoretical guarantee applies jointly to all of the neurons in a layer, and all of the layers in a network.

Our work is also the first exact moment matching in residual networks, which requires resolving cross-covariances between the pre-activation and the activation. The closest prior work is Wu et al. [2019, Appendix C.1], which uses asymptotic approximations.

**Roadmap** §2 states the full uncertainty-propagation problem (Problem 1) and the reduced single-layer Gaussian moment matching problem (Problem 2), and then resolves the latter with exact mean and covariance formulas for feedforward and residual layers (Lemma 2.4). §3 clarifies the limitations of various Gaussian moment propagation methods, including our own, through example problems. §4 applies these formulas to address the full multilayer problem by recursive moment matching over layers and bounds the error in a smooth fourth-order integral probability metric. §5 validates the moment matching accuracy on single- and multilayer random networks, applies them to variational Bayes neural networks, and sketches an extension to stochastic activations.

## 2 Exact, closed-form analytical expressions for mean and covariance

A neural network is a composition of layer functions $\mathbb{R}^n \to \mathbb{R}^m$

$$g(x; A, b, C, d) = \sigma(Ax + b) + Cx + d, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, C \in \mathbb{R}^{m\times n}, d \in \mathbb{R}^m$ are parameters; the activation function of a neural network is denoted by $\sigma : \mathbb{R} \to \mathbb{R}$ and applies elementwise. Except for parameters $A, b, C, d$, capital letters refer to random variables. The layers of a neural network are indicated by superscripts, e.g. $A^k$ is a matrix of parameters for the $k$th layer. If $X$ is a square-integrable random vector, the notation $\mathrm{N}\, X$ refers to a random variable distributed as $\mathcal{N}(\mathbb{E}\, X, \operatorname{Cov} X)$.

**Definition 2.1.** A neural network with $\ell$ layers is the function $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ defined by

$$\begin{aligned} f(x) &= f^\ell(x) \\ f^k(x) &= g(f^{k-1}(x); A^k, b^k, C^k, d^k) && \forall k \in \{1 \dots \ell\} \\ f^0(x) &= x \end{aligned}$$

Stated formally, the problem of uncertainty propagation studied in this paper is:

**Problem 1.** *Let $f$ be a neural network with $\ell$ layers. Given $X \sim \mathcal{N}(\mu, \Sigma)$, characterize the distribution of $Y_{\mathrm{true}} = f(X)$.*

After layer-wise Gaussian approximation, this problem reduces to:

**Problem 2.** *Given $X \sim \mathcal{N}(\mu, \Sigma)$ and $A, b, C, d$; find exact expressions for $\mathbb{E}\, g(X; A, b, C, d)$ and $\operatorname{Cov} g(X; A, b, C, d)$.*

### 2.1 Our analytic moment-matching method $Y_{\mathrm{ana}}$

In moment matching, we re-approximate each layer by a Gaussian sharing its first two moments:

**Definition 2.2.** Let $f$ be a neural network with $\ell$ layers. Given $X \sim \mathcal{N}(\mu, \Sigma)$, the moment-matching Gaussian approximation of $f(X)$, is the random variable $Y_{\text{ana}}$ defined by

$$\begin{aligned} Y_{\text{ana}} &= Y^\ell \\ Y^k &= \mathrm{N}\, g(Y^{k-1}; A^k, b^k, C^k, d^k) && \forall k \in \{1 \dots \ell\} \\ Y^0 &= X \end{aligned}$$

We specify three transcendental functions needed to compute the first two Gaussian moments of a layer defined by (1).

**Definition 2.3.** Given a nonlinear function $\sigma : \mathbb{R} \to \mathbb{R}$, the functions $M_\sigma : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}$ and $K_\sigma, L_\sigma : \mathbb{R}^2 \times \mathbb{R}^{2\times 2}_{\geq 0} \to \mathbb{R}$ are

$$M_\sigma(\mu; \nu) = \mathbb{E}\, \sigma(X), \qquad X \sim \mathcal{N}(\mu, \nu);$$

$$K_\sigma(\mu_1, \mu_2; \nu_{11}, \nu_{22}, \nu_{12}) = \mathrm{Cov}(\sigma(X_1), \sigma(X_2)), \qquad \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \nu_{11} & \nu_{12} \\ \nu_{12} & \nu_{22} \end{pmatrix} \right);$$

and

$$L_\sigma(\mu_1; \nu_{11}, \nu_{22}, \nu_{12}) = \mathrm{Cov}(\sigma(X_1), X_2), \qquad \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} \mu_1 \\ \star \end{pmatrix}, \begin{pmatrix} \nu_{11} & \nu_{12} \\ \nu_{12} & \nu_{22} \end{pmatrix} \right).$$

Approximating or computing these functions enables moment matching:

**Lemma 2.4.** *For some activation function $\sigma$, let $g_\sigma(x; A, b, C, d) = \sigma(Ax + b) + Cx + d$. Let $X \sim \mathcal{N}(\mu, \Sigma)$. Then*

$$\big(\mathbb{E}\, g_\sigma(X; A, b, C, d)\big)_i = M_\sigma(\mu_i; \nu_{ii}) + (C\mu)_i + d_i$$

*and*

$$\begin{aligned} \big(\mathrm{Cov}\, g_\sigma(X; A, b, C, d)\big)_{i,j} = K_\sigma\left(\mu_i, \mu_j; \nu_{ii}, \nu_{jj}, \nu_{ij}\right) \\ + L_\sigma\left(\mu_i; \nu_{ii}, \tau_{jj}, \kappa_{ij}\right) + L_\sigma\left(\mu_j; \nu_{jj}, \tau_{ii}, \kappa_{ji}\right) + \tau_{ij}, \end{aligned}$$

*where for all valid indices $(i, j)$,*

$$\begin{aligned} \mu_i &= (A\mu + b)_i & \tau_{ij} &= (C\Sigma C^\intercal)_{i,j} \\ \nu_{ij} &= (A\Sigma A^\intercal)_{i,j} & \kappa_{ij} &= (A\Sigma C^\intercal)_{i,j} \end{aligned}$$

Given that their domain is five-dimensional, these functions are not amenable to tabulation, and are typically approximated in the state of the art. In this work, instead, we compute them analytically for hidden layers with the following activation functions:

**probit** in App. C by expressing $\Phi(x) = \mathbb{E}\,[Z \leq x]$ in terms of an auxiliary standard Normal $Z$, generalizing Huber [2020, § III.B] and Wright et al. [2024, App. B.4].

**GeLU** in App. D by repeated applications of the multivariate Stein's lemma and the Gaussian ODE $\phi'(x) + x\phi(x) = 0$, generalizing Wright et al. [2024, App. B.2].

**ReLU** in App. E by using the Dominated Convergence Theorem to take the pointwise limit

$$\mathrm{ReLU}(x) = \lim_{\lambda \to \infty} \lambda^{-1}\, \mathrm{GeLU}(\lambda x),$$

generalizing Frey and Hinton [1999, App. C.3], Cho and Saul [2009, App. A], Wu et al. [2019, App A.2.2], Huber [2020, § III.C], and Wright et al. [2024, App. B.2].

**Heaviside** in App. F by using the Dominated Convergence Theorem to take the pointwise limit

$$\mathrm{Heaviside}(x) = \lim_{\lambda \to \infty} \Phi(\lambda x),$$

generalizing Frey and Hinton [1999, App. C.2], Wu et al. [2019, App. A.2.1], and Wright et al. [2024, App. B.2].

**sine** in App. G by combining the characteristic function of the Normal distribution with the trigonometric identity $\sin(x) = \left(e^{ix} - e^{-ix}\right)/(2i)$, generalizing Sitzmann et al. [2020, App. 1].

The calculations are interesting in themselves because they are probabilistic in nature: we never resort to Riemann integrals against the Gaussian density. Instead we only use the fact that Gaussian density satisfies the ordinary differential equation $\phi'(x) = -x\phi(x)$ and the closely related Stein's identity $\mathbb{E}\, X f(Y) = \mathrm{Cov}(X, Y)\, \mathbb{E}\, f'(Y)$ for jointly Normal $(X, Y)$ and differentiable $f$, and derive from these a panoply of Gaussian integral identities in App. B which are ultimately used to solve all of the required integrals. To convey the flavor of these derivations, we list the most original result:

$$
\begin{aligned}
&K_{\text{GeLU}}(\mu_1, \mu_2; \nu_{11}, \nu_{22}, \nu_{12}) \\
&\quad= \left(\mu_1\nu_{12} + \mu_2\nu_{11} - \frac{\mu_1\nu_{12}\nu_{11}}{1+\nu_{11}}\right) \frac{1}{\sqrt{1+\nu_{11}}} \Phi_{2;1}\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&\quad+ \left(\mu_2\nu_{12} + \mu_1\nu_{22} - \frac{\mu_2\nu_{12}\nu_{22}}{1+\nu_{22}}\right) \frac{1}{\sqrt{1+\nu_{22}}} \Phi_{2;1}\left(\frac{\mu_2}{\sqrt{1+\nu_{22}}}, \frac{\mu_1}{\sqrt{1+\nu_{11}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&\qquad+ \frac{\nu_{11}\nu_{22} + \nu_{12}^2\left(1 - \frac{\nu_{11}}{1+\nu_{11}} - \frac{\nu_{22}}{1+\nu_{22}}\right)}{\sqrt{(1+\nu_{11})(1+\nu_{22})}} \phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&\quad+ (\mu_1\mu_2 + \nu_{12})\, \Phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) - M_{\text{GeLU}}(\mu_1; \nu_{11}) M_{\text{GeLU}}(\mu_2; \nu_{22}).
\end{aligned}
$$

### 2.2 Targets: ground truth $Y_{\text{true}}$ and pseudo-truth $Y_{\text{pseudo}}$

We benchmark our analytic approach and baseline methods (listed in Appendix A) against the **true distribution** $Y_{\text{true}} = f(X)$ and the **pseudo-true distribution** $Y_{\text{pseudo}} = \mathcal{N}\left(\mathbb{E}\, Y_{\text{true}}, \mathrm{Cov}\, Y_{\text{true}}\right)$. While $Y_{\text{true}}$ is the ideal answer to Problem 1, $Y_{\text{pseudo}}$ is the closest (by KL divergence) Gaussian approximation to $Y_{\text{true}}$.

## 3 Explicit examples

In order to motivate our theoretical results, we present some explicit examples which both show why analytic moment propagation is necessary and define its fundamental limitations. First, linear, unscented, and mean-field propagation (baselines presented in Appendix A) can be arbitrarily wrong:

**Example 1** (for linear propagation)**.** *Consider the network $Y = \sin(X)$, where $X \sim \mathcal{N}(0, \sigma^2)$. Then $Y_{\text{lin}} = \mathcal{N}(0, \sigma^2)$. But in fact $\mathrm{Var}\, Y = (1 - e^{-2\sigma^2})/2$ which tends to $1/2$ for large $\sigma^2$. So by increasing $\sigma^2$, we can make $Y_{\text{lin}}$ arbitrarily wrong.*

**Example 2** (for unscented propagation)**.** *Suppose that $X \sim \mathcal{N}(0, 1)$, and the sigma points are $X \in \{-\alpha, 0, \alpha\}$ for some $\alpha > 0$. Then on the neural network $Y = \sin(\alpha^{-1}\pi X)$, $Y_{\text{u95}}$ and $Y_{\text{u02}}$ will be identically zero and arbitrarily wrong.*

**Example 3** (for mean-field propagation)**.** *Consider the following (linear) network, with scalar input $X$, hidden $Y^1 \in \mathbb{R}^m$, and scalar output $Y$:*

$$
Y = \frac{1}{m}\sum_{i=1}^{m} Y_i^1, \qquad Y_i^1 = X, \qquad X \sim \mathcal{N}(0, 1).
$$

*The mean-field approximation treats each $Y_i^1$ as independent $\mathcal{N}(0, 1)$, so it concludes $Y \sim \mathcal{N}(0, m^{-1})$. But $Y$ is identical to $X \sim \mathcal{N}(0, 1)$. So by increasing $m$, we can make $Y_{\text{mfa}}$ arbitrarily wrong.*

Whereas mean-field propagation is exact only for single neurons, the mean and variance of our method $Y_{\text{ana}}$ are exact on all single-hidden-layer networks, which includes all of the examples above. But there is no free lunch (moment matching is $\sharp P$-hard), so we cannot expect it to be exact for all multi-layer networks. In order to understand our method's limitations, we push it past the breaking point by constructing an explicit multi-layer network combining strong non-normality with strong nonlinearity.

**Example 4** (for $Y_{\text{ana}}$)**.** *Consider the following network, where input $X$, output $Y$, and hidden $Y^1$ are scalars; $u(x) = \mathbf{1}_{x \geq 0}$ is the Heaviside function; and $\alpha$ is a weight:*

$$Y = \alpha u(Y^1 - 3), \qquad Y^1 = 2u(X), \qquad X \sim \mathcal{N}(0, 1).$$

*At the hidden layer, $Y^1$ is approximated by $\mathcal{N}(1, 1)$. Therefore $Y_{\text{ana}}$ is approximated by some non-degenerate Normal distribution, scaled by $\alpha$. By increasing $\alpha$, we can make $Y_{\text{ana}}$ arbitrarily wrong: $Y_{\text{true}}$ is identically zero because $Y^1 - 3$ is always negative.*

## 4 Second-order accuracy of moment matching

We quantify the error from the Gaussian Ansatz using a smooth integral probability metric. For $h : \mathbb{R}^d \to \mathbb{R}$, define the the homogeneous Sobolev seminorm

$$\|h\|_{\dot{W}^{4,\infty}} := \max_{1 \leq j \leq 4} \sup_{x \in \mathbb{R}^d} \|D^j h(x)\|_{\text{op}}.$$

**Definition 4.1** (Fourth-order smooth distance)**.** For random variables $U, V$ taking values in $\mathbb{R}^d$, define

$$d_4(U, V) := \sup_{\|h\|_{\dot{W}^{4,\infty}} \leq 1} \left| \mathbb{E}\, h(U) - \mathbb{E}\, h(V) \right|.$$

This is the integral probability metric generated by the homogeneous unit ball of $\dot{W}^{4,\infty}$, equivalently smooth test functions with uniformly bounded derivatives of orders one through four. The smoothness assumption covers the probit, GeLU, and sine activations.

**Theorem 4.2** (Single-layer higher-order accuracy)**.** *Let $X = \mu_X + \lambda^{-1}\xi$, where $\xi \sim \mathcal{N}(0, \Sigma_X)$, and let $Y = f(X)$. Assume that $f : \mathbb{R}^n \to \mathbb{R}^d$ has bounded derivatives of orders $1, \ldots, 4$. Let $\gamma_Y \sim \mathcal{N}(\mathbb{E}\, Y, \operatorname{Cov} Y)$ be the moment-matched Gaussian. Then*

$$d_4(Y, \gamma_Y) = O(\lambda^{-4}).$$

*By contrast, the linearized Gaussian generically has $O(\lambda^{-2})$ error in the same smooth distance because it misses the second-order mean correction.*

The single-layer theorem is the local statement behind recursive moment matching. To propagate it through the network, we need the analogue of the Lipschitz property of Wasserstein distance. To propagate this result through the moment-matching recursion, we define the fourth-order composition constant of a $C^4$ map $g$: $C_4(g) := \sup_{\|h\|_{\dot{W}^{4,\infty}} \leq 1} \|h \circ g\|_{\dot{W}^{4,\infty}}$. By the chain rule and the Faà di Bruno formula applied to $g(x; A, b, C, d) = \sigma(Ax + b) + Cx + d$, $C_4(g)$ is bounded by a polynomial in $\|A\|$, $\|C\|$, and $\|\sigma\|_{\dot{W}^{4,\infty}}$.

**Theorem 4.3** (Recursive smooth-distance error bound)**.** *Let $Y^k_{\text{true}}$ and $Y^k$ be true and moment-matched distributions at layer $k$ in a multi-layer network. Assume that $\sigma$ has bounded derivatives of orders $1, \ldots, 4$. If $X$ has covariance $O(\lambda^{-2})$, then $d_4(Y^k_{\text{true}}, Y^k) = O(\lambda^{-4})$ for all $k \in [1 \ldots \ell]$.*

*Proof.* The proof is the triangle inequality for the integral probability metric $d_4$, followed by the composition bound

$$d_4(g(U), g(V)) \leq C_4(g) d_4(U, V).$$

The details are given in Appendix H. □

Theorems 4.2 and 4.3 formalize the main accuracy claim. Exact moment matching cancels the first two smooth-test terms at each layer, so its local Gaussianization error is higher order than the generic error of perturbative (linear and unscented) approximations that do not exactly match the true layer moments. Fig. 4.1 confirms that analytic moment matching converges quadratically, rather than linearly, in the input covariance. The recursion shows that this higher-order local accuracy is not merely a single-layer phenomenon: it is chained through the full network, with amplification controlled by explicit fourth-order composition constants depending on the weights and activation regularity.

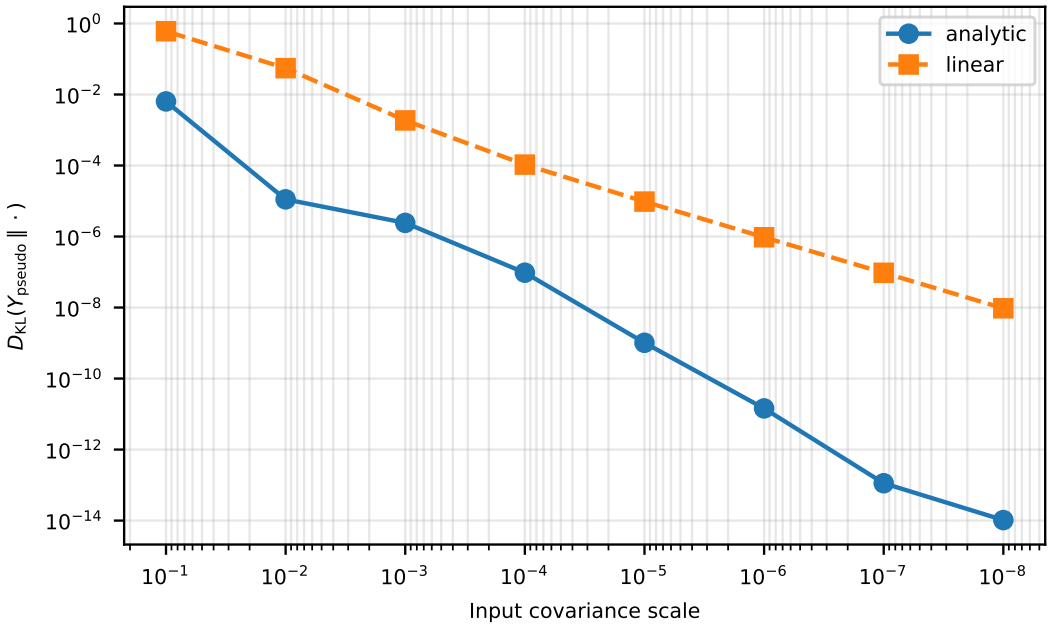



Figure 4.1: Empirical small-covariance convergence for a fixed deep sine network. Methodology is described in Appendix I.

## 5 Examples and applications

In numerical results, we obtain $Y_{\mathrm{true}}$ and $Y_{\mathrm{pseudo}}$ by quasi-Monte Carlo simulation. We report the KL divergence from $Y_{\mathrm{pseudo}}$ as a measure of how accurately our method computes means and covariances. Moreover, we report the Wasserstein distance from $Y_{\mathrm{true}}$ in order to assess whether our method's Gaussian Ansatz at the output layer is close to the true, non-Gaussian distribution. The Wasserstein distance is a metric on probability distributions whose induced topology is strictly stronger than weak convergence, and we use it because it can be computed efficiently in one dimension.

**Definition 5.1.** Let $X$ and $Y$ be random variables taking values $\mathbb{R}^n$. The Wasserstein distance $d_{\mathrm{W}}(X, Y)$ is

$$d_{\mathrm{W}}(X, Y) = \sup_{\|\nabla h\|_\infty \leq 1} \mathbb{E}(h(X) - h(Y)).$$

### 5.1 Random networks

We apply our method and other benchmarks to 36 different ensembles of random neural networks with more than one hidden layer. They are designed to cover a range of architectures, weights, and activations, and to stress-test the assumptions of layer-by-layer moment matching. We sample one neural network from each ensemble and evaluate the goodness of approximation of the output distribution for three input distributions. In each case we evaluate baselines (Appendix A) $Y_{\mathrm{true}}$, $Y_{\mathrm{pseudo}}$, $Y_{\mathrm{ana}}$, $Y_{\mathrm{mfa}}$, $Y_{\mathrm{lin}}$, $Y_{\mathrm{u95}}$, and $Y_{\mathrm{u02}}$, and compare distance to the true distribution $d_{\mathrm{W}}(\cdot, Y_{\mathrm{true}})$ and KL divergence from the pseudo-true distribution $d_{\mathrm{KL}}(\cdot, Y_{\mathrm{pseudo}})$.

Specifications of the 108 test cases and the Monte Carlo methodology are in Appendix L; the individual tables and distribution plots are in the accompanying *Supplementary Numerical Results*. For large input variance, the median atio of baseline to our method's KL divergence is about 33 for Unscented'95, 340 for linearization (over the 32 cases with finite results), and 2,000 for Unscented'02, as seen in Fig. 5.1. Even at small input variance, where linearization is justified by the delta method, our method has a lower KL divergence than linearization in all 32 cases with finite linearization results, as seen in Fig. 5.1. The medium-variance case is visualized in Appendix L.1.

Many examples, such as the small-variance test case with wide architecture, trained weights, and sine activation, exhibit the underdispersion of the mean-field approximation predicted in Example 3.

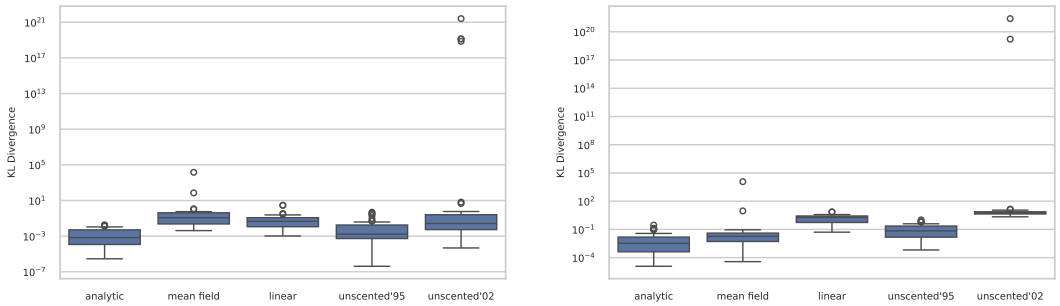



Figure 5.1: Comparison of goodness of approximation (lower KL divergence is better) for all random neural networks, grouped by approximation method, in the small (left) and large (right) input variance scenarios. Four non-finite results are omitted in each scenario.

It is noteworthy that even in the small-variance regime, where our method, linearization, and the unscented transformations are first-order asymptotically equivalent, our method is often still closest to the pseudo-true distribution. This confirms our theoretical claim that moment matching is a second-order accurate method, while linearization is only first-order.

### 5.2 Weight uncertainty: variational Bayesian neural networks

We apply our distributional approximation to variational inference for Bayesian neural networks. In this case, the input $x$ is deterministic and the weights $w$ are random with a prior distribution $p(w)$. The network predicts $y \sim p(y \mid w, x)$. The posterior distribution is approximated variationally as $q(w; \theta)$ by optimizing the evidence lower bound [Blundell et al., 2015]:

$$\theta^* = \arg\min_\theta \left\{ -\mathbb{E}_{w\sim q(w;\theta)} \log p(y \mid w, x) + D_{\mathrm{KL},w}(q(w;\theta) \mid p(w)) \right\}. \tag{2}$$

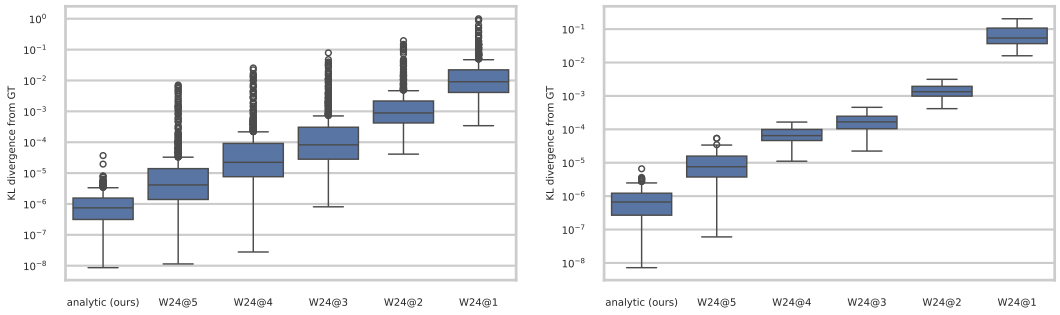
Our example uses the GeLU activation function and, like Wright et al. [2024, §4.3], a single hidden layer; for other details, see App. J.

Monte Carlo Variational Inference (MCVI), a stochastic gradient method that approximates $\mathbb{E}_{w\sim q(w;\theta)}$ with Monte Carlo samples, is the gold standard for training variational Bayes neural networks. In order to reduce the gradient variance and computational cost, Wu et al. [2019], Petersen et al. [2024], Wright et al. [2024], Rui Li and Trapp [2025] use deterministic moment-matching for training (to compute gradients of evidence lower bound), and for inference (generate the predictive distribution). While this setup evaluates the end-to-end learning process, it does not directly measure the accuracy of the underlying distributional approximation. In particular, a sufficiently expressive variational model may compensate for systematic errors in distribution propagation, rendering the overall inference procedure robust to approximation bias. However, strong end-to-end performance does not necessarily imply that the propagated distributions themselves are accurate. [1]

Because our interest is in the correctness of distributional approximation, our experiments differ from the works cited in the following ways:

1. We train the Bayesian neural network using MCVI with a large Monte Carlo batch size. After training has converged, the Monte Carlo predictions from this network are taken as ground truth mean and variance.
2. We test the Bayesian neural network using Monte Carlo on the variational posterior, as well as six different techniques to propagate distributions through activation functions: the power series of Wright et al. [2024], expanded to 1–5 terms, and our method.

[1] "Learning can still improve a bound on the log likelihood of the data even when the posterior distribution over hidden states is computed incorrectly"; see also the surrounding discussion in Frey and Hinton [1999, §1.2]. For a toy example, take $Z \sim \mathcal{N}(0, 1)$; if the uncertainty propagation formula were $aZ \sim \mathcal{N}(0, 2a)$, which is patently incorrect, the variational model could learn to approximate the target distribution $\mathcal{N}(0, \sigma^2)$ using the parameter $a = \sigma^2/2$. If the approximations were accurate, one would expect the different inference methods to be indistinguishable from each other and from MCVI, as they are in Wright et al. [2024, Table 3].




(a) Parkinsons telemonitoring

(b) Concrete compressive strength

Figure 5.2: KL divergence between ground-truth predictive distribution (by Monte Carlo) and approximations for two UCI regression datasets. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

3. The figure of merit is the KL divergence (lower is better) from the pseudo-true Gaussian distribution (via Monte Carlo) to each deterministic approximation.

We apply this method to eleven regression datasets from the UCI Machine Learning Repository. The dataset list and full results are in Appendix J; we highlight the Parkinsons telemonitoring and concrete compressive strength datasets here. As Fig. 5.2 shows, the power series expansion of Wright et al. [2024] shows a "dose response," i.e., it becomes more accurate as more terms are added. But the most accurate approximation to the Monte Carlo predictive distribution is attained using exact moment matching, our method. It is not exact because, for a Bayesian neural network with even a single hidden layer, Gaussian moment matching introduces additional error due to multiplication of random weights and layer inputs [Goulet et al., 2021, eqq. 3–6].

### 5.3 Stochastic activations

The 2024 Nobel Prize in Physics recognized the stochastic, biological inspiration for artificial neurons in machine learning [Davour]. Even though biological motivation is no longer front of mind, as a curiosity we analyze the output distribution of a neural network whose activations are random processes modeled after a stochastic neuron:

$$\tilde{\sigma}(x, U) = 2\mathbf{1}_{U<\Phi(x)} - 1, \tag{3}$$

where at each artificial neuron, $U \sim \mathrm{Uniform}(0, 1)$ is an independent random variable.

In Appendix K we derive a moment-matching approximation to the distribution of the output of a stochastic neural network. Applying this formula to 1 million samples from a stochastic version of the "deep" neural network of §5.1 with a constant zero input results in a normal distribution with a good agreement (Fig. K.1). Whether this line of inquiry deserves further methodological development is reserved for future work.

## 6 Limitations

A limitation of our moment matching derivation is that it does not apply to softmax (including attention), which appears to remain an open problem. Because the numerical examples are scoped to presenting our theory and derivations, we do not test on large problems or benchmark against other methodologies for applied machine learning.

## 7 Extensions

### 7.1 Residual connections across multiple layers

It is possible to represent ResNets with multi-layer skips within the formalism $\sigma(Ax + b) + Cx + d$, by using block matrices to pass the residual stream through each layer. Consider an $n$-deep residual

connection $\tilde{f}(x)$, defined by

$$\begin{aligned} \tilde{f}(x) &= x + f^n(x), & f^0(x) &= x, \\ f^k(x) &= \sigma\left(A^k f^{k-1}(x) + b^k\right), & k &\in [n]. \end{aligned}$$

This can be represented by enlarging the hidden layers to $\tilde{f}^k(x) := (f^k(x), x)$. For any vector $b_0$ satisfying $\sigma(b_0) = 0$ (e.g., the zero vector for sine or GeLU),

$$\begin{aligned} \tilde{f}^1(x) &= \sigma\left(\begin{pmatrix} A^1 \\ 0 \end{pmatrix} x + \begin{pmatrix} b^1 \\ b_0 \end{pmatrix}\right) + \begin{pmatrix} 0 \\ I \end{pmatrix} x, \\ \tilde{f}^k(x) &= \sigma\left(\begin{pmatrix} A^k & 0 \\ 0 & 0 \end{pmatrix} \tilde{f}^{k-1}(x) + \begin{pmatrix} b^k \\ b_0 \end{pmatrix}\right) + \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \tilde{f}^{k-1}(x), \qquad k \in [2 \dots n] \\ \tilde{f}(x) &= \begin{pmatrix} I & I \end{pmatrix} \tilde{f}^n(x). \end{aligned}$$

In this way, it is possible to moment-match across multi-layer skips by tracking the cross-covariances between the activation and the residual stream.

### 7.2 Kalman filtering

In Kalman filtering, a state estimation algorithm for dynamic systems, it is important to propagate the belief state $X_t \sim \mathcal{N}(\mu_t, \Sigma_t)$ through the recursion $X_{t+1} = f(X_t)$ and the measurement law $Y_t = h(X_t)$. In the measurement update step [Särkkä and Svensson, 2023, Chapter 8], a function $h$ is given, and the task is to compute the covariance between $X$ and $h(X)$. Moment matching results in an approximately Gaussian distribution for the pair $(X, h(X))$, whose covariance matrix's off-diagonal blocks contain the desired cross-covariance:

$$\operatorname{Cov}\begin{pmatrix} X \\ h(X) \end{pmatrix} = \begin{pmatrix} \operatorname{Cov} X & \operatorname{Cov}(X, h(X)) \\ \operatorname{Cov}(h(X), X) & \operatorname{Cov} h(X) \end{pmatrix}.$$

Moment matching is a promising numerical scheme for joint state and measurement moments, and has been developed in concurrent work [Kuang and Lin, 2026].

### 7.3 Misclassification under Gaussian perturbations

One application of evaluating $f(X)$ under Gaussian input is to study the robustness of the network $f$ to random perturbations. Suppose $f$ gives the class scores of a classifier and $i^*$ is the correct label. A competing class $i \neq i^*$ is misclassified with probability

$$\Pr\left(f_i(X) > f_{i^*}(X)\right) = \mathbb{E}\left(\text{Heaviside}\left(f_i(X) - f_{i^*}(X)\right)\right),$$

which can be solved using moment matching.

## 8 Novelty and significance

Our theoretical analysis shows that Gaussian moment matching is second-order accurate. Our moment matching method is the first to achieve this guarantee.

Prior works on uncertainty propagation in neural networks have relied on approximations for the first and second moments under Gaussian inputs, including independence assumptions, approximate activation functions, and truncated series expansions. In contrast, we derive exact, closed-form expressions for these moments for a range of widely used activation functions, enabling layer-wise moment propagation in deep networks without restrictive assumptions. Our moment matching methods support general multivariate Gaussian input distributions and residual network architectures. Empirically, we show that even beyond the guaranteed regime, the resulting method achieves orders-of-magnitude improvements in distributional accuracy over existing approaches. We further demonstrate its effectiveness in uncertainty-aware inference on real data and in deterministic inference for variational Bayesian neural networks.

## Reproducibility statement

In the supplementary material, executable Python scripts are in `demo/`. They reference libraries in `neural_uncertainty_propagation/` and generate output in `docs/manuscript/generated/`. `deep_sine_convergence.py` generates Fig. 4.1 and the results described in Appendix I. `test_case.py` generates the random-network tables and figures in the accompanying *Supplementary Numerical Results*. `test_case_summary.py` uses its CSV output to generate the KL summary boxplots. To reproduce §K, run `stochastic.py`. To reproduce §J, run `bayesian-network.py`.

## Acknowledgements

This work was supported by the National Science Foundation CAREER Program (Grant No. 2046292).

## References


Ahmed Hussen Abdelaziz, Shinji Watanabe, John R. Hershey, Emmanuel Vincent, and Dorothea Kolossa. Uncertainty propagation through deep neural networks. In *Interspeech 2015*, pages 3561–3565. ISCA, September 2015. doi:10.21437/Interspeech.2015-706. URL https://www.isca-archive.org/interspeech_2015/abdelaziz15_interspeech.html.

Abdullah Akgül, Manuel HauSSmann, and Melih Kandemir. Deterministic Uncertainty Propagation for Improved Model-Based Offline Reinforcement Learning, January 2025. URL http://arxiv.org/abs/2406.04088. arXiv:2406.04088 [cs].

Ramón Fernandez Astudillo and João Paulo Da Silva Neto. Propagation of uncertainty through multilayer perceptrons for robust automatic speech recognition. In *Interspeech 2011*, pages 461–464, ISCA, August 2011. ISCA. doi:10.21437/interspeech.2011-196. URL https://www.isca-archive.org/interspeech_2011/astudillo11_interspeech.html.

Max Little Athanasios Tsanas. Parkinsons Telemonitoring, 2009. URL https://archive.ics.uci.edu/dataset/189.

Richard Bergna, Stefan Depeweg, Sergio Calvo Ordoñez, Jonathan Plenk, Alvaro Cartea, and José Miguel Hernández-Lobato. Post-Hoc Uncertainty Quantification in Pre-Trained Neural Networks via Activation-Level Gaussian Processes. In *7th Symposium on Advances in Approximate Bayesian Inference Workshop Track*, 2025. URL https://openreview.net/forum?id=y78LfYhlNy.

Adel Bibi, Modar Alfadly, and Bernard Ghanem. Analytic Expressions for Probabilistic Moments of PL-DNN with Gaussian Input. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9099–9107, Salt Lake City, UT, June 2018. IEEE. doi:10.1109/cvpr.2018.00948. URL https://ieeexplore.ieee.org/document/8579046/.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Network. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France, July 2015. PMLR. URL https://proceedings.mlr.press/v37/blundell15.html.

Youngmin Cho and Lawrence Saul. Kernel Methods for Deep Learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/5751ec3e9a4feab575962e78e006250d-Paper.pdf.

Anna Davour. The Nobel Prize in Physics 2024 - Popular science background. URL https://www.nobelprize.org/prizes/physics/2024/popular-information/.

DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Milo Stanojevi, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem, 2020. URL http://github.com/google-deepmind.

Zvi Drezner and G. O. Wesolowsky. On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, 35(1-2):101–107, March 1990. ISSN 0094-9655. doi:10.1080/00949659008811236. URL https://doi.org/10.1080/00949659008811236. _eprint: https://doi.org/10.1080/00949659008811236.

Brendan J. Frey and Geoffrey E. Hinton. Variational Learning in Nonlinear Gaussian Belief Networks. *Neural Computation*, 11(1):193–213, January 1999. ISSN 0899-7667, 1530-888X. doi:10.1162/089976699300016872. URL https://direct.mit.edu/neco/article/11/1/193-213/6239.

James-A. Goulet, Luong Ha Nguyen, and Saeid Amiri. Tractable Approximate Gaussian Inference for Bayesian Neural Networks. *Journal of Machine Learning Research*, 22(251):1–23, 2021. URL http://jmlr.org/papers/v22/20-1009.html.

Insu Han, Amir Zandieh, Jaehoon Lee, Roman Novak, Lechao Xiao, and Amin Karbasi. Fast Neural Kernel Embeddings for General Activations. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 35657–35671. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/e7be1f4c6212c24919cd743512477c13-Paper-Conference.pdf.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, December 2015. doi:10.1109/ICCV.2015.123. URL https://ieeexplore.ieee.org/document/7410480/.

Marco F. Huber. Bayesian Perceptron: Towards fully Bayesian Neural Networks. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3179–3186, December 2020. doi:10.1109/CDC42340.2020.9303764. URL https://ieeexplore.ieee.org/abstract/document/9303764.

I-Cheng Yeh. Concrete Compressive Strength, 1998. URL https://archive.ics.uci.edu/dataset/165.

I-Cheng Yeh. Real Estate Valuation, 2018. URL https://archive.ics.uci.edu/dataset/477.

Shida Jiang, Junzhe Shi, and Scott Moura. A New Framework for Nonlinear Kalman Filters, February 2025. URL http://arxiv.org/abs/2407.05717. arXiv:2407.05717 [eess].

S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3): 477–482, March 2000. ISSN 0018-9286. doi:10.1109/9.847726. URL http://ieeexplore.ieee.org/document/847726/.

Simon J. Julier and Jeffrey K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Defense, Security, and Sensing*, 1997. URL https://api.semanticscholar.org/CorpusID:7937456.

S.J. Julier. The scaled unscented transformation. In *Proceedings of the 2002 American Control Conference*, volume 6, pages 4555–4559 vol.6, May 2002. doi:10.1109/ACC.2002.1025369. URL https://ieeexplore.ieee.org/document/1025369.

S.J. Julier and J.K. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401–422, March 2004. ISSN 0018-9219. doi:10.1109/JPROC.2003.823141. URL http://ieeexplore.ieee.org/document/1271397/.

S.J. Julier, J.K. Uhlmann, and H.F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of 1995 American Control Conference - ACC'95*, volume 3, pages 1628–1632, Seattle, WA, USA, 1995. American Autom Control Council. doi:10.1109/acc.1995.529783. URL http://ieeexplore.ieee.org/document/529783/.

Paul Jungmann, Julia Poray, and Akash Kumar. Analytical Uncertainty Propagation in Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2495–2508, February 2025. ISSN 2162-2388. doi:10.1109/TNNLS.2023.3347156. URL https://ieeexplore.ieee.org/document/10398277.

Patrick Kidger and Cristian Garcia. Equinox: neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming workshop at Neural Information Processing Systems 2021*, 2021.

Simon Kuang and Xinfan Lin. Assumed Density Filtering and Smoothing with Neural Network Surrogate Models. In *Proceedings of The 8th Annual Learning for Dynamics and Control Conference*, pages 1741–1790. PMLR, June 2026. URL https://proceedings.mlr.press/v331/kuang26b.html.

Lennart Ljung. unscentedKalmanFilter. In *System Identification Toolbox Reference*, pages 1–2372. The MathWorks, Inc., 2025.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Bengt Muthén. Moments of the censored and truncated bivariate normal distribution. *British Journal of Mathematical and Statistical Psychology*, 43(1):131–143, May 1990. ISSN 0007-1102, 2044-8317. doi:10.1111/j.2044-8317.1990.tb00930.x. URL https://bpspsychub.onlinelibrary.wiley.com/doi/10.1111/j.2044-8317.1990.tb00930.x.

Tobias Nagel and Marco F. Huber. Kalman-Bucy-Informed Neural Network for System Identification. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1503–1508, December 2022. doi:10.1109/CDC51059.2022.9993245. URL https://ieeexplore.ieee.org/document/9993245/.

D. B. Owen. A table of normal integrals: A table. *Communications in Statistics - Simulation and Computation*, 9(4):389–419, January 1980. ISSN 0361-0918, 1532-4141. doi:10.1080/03610918008812164. URL http://www.tandfonline.com/doi/abs/10.1080/03610918008812164.

Anbal Morais Paulo Cortez. Forest Fires, 2007. URL https://archive.ics.uci.edu/dataset/162.

Felix Petersen, Aashwin Mishra, Hilde Kuehne, Christian Borgelt, Oliver Deussen, and Mikhail Yurochkin. Uncertainty Quantification via Stable Distribution Propagation, February 2024. URL http://arxiv.org/abs/2402.08324. arXiv:2402.08324 [cs].

Heysem Kaya Pnar Tfekci. Combined Cycle Power Plant, 2014. URL https://archive.ics.uci.edu/dataset/294.

R. Quinlan. Auto MPG, 1993. URL https://archive.ics.uci.edu/dataset/9.

Michael Redmond. Communities and Crime, 2002. URL https://archive.ics.uci.edu/dataset/183.

Arno Solin Rui Li, Marcus Klasson and Martin Trapp. Streamlining Prediction in Bayesian Deep Learning. In *International Conference on Learning Representations (ICLR)*, 2025.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/53c04118df112c13a8c34b38343b9c10-Abstract.html.

Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*. Institute of Mathematical Statistics textbooks. Cambridge University Press, New York, second edition edition, 2023. ISBN 978-1-108-92664-5.

Yanbo Tang. A Note on Monte Carlo Integration in High Dimensions. *The American Statistician*, 78(3):290–296, July 2024. ISSN 0003-1305. doi:10.1080/00031305.2023.2267637. URL https://doi.org/10.1080/00031305.2023.2267637. _eprint: https://doi.org/10.1080/00031305.2023.2267637.

D. Pope Thomas Brooks. Airfoil Self-Noise, 1989. URL https://archive.ics.uci.edu/dataset/291.

Jessica S. Titensky, Hayden Jananthan, and Jeremy Kepner. Uncertainty Propagation in Deep Neural Networks Using Extended Kalman Filtering. In *2018 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pages 1–4, October 2018. doi:10.1109/URTC45901.2018.9244804. URL https://ieeexplore.ieee.org/abstract/document/9244804.

Karl Ulrich. Servo, 1986. URL https://archive.ics.uci.edu/dataset/87.

Unknown. Liver Disorders, 2016. URL https://archive.ics.uci.edu/dataset/60.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, Ilhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2. URL https://doi.org/10.1038/s41592-019-0686-2.

Philipp Wagner, Xinyang Wu, and Marco F. Huber. Kalman Bayesian Neural Networks for Closed-form Online Learning, November 2022. URL http://arxiv.org/abs/2110.00944. arXiv:2110.00944 [cs].

E.A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158, Lake Louise, Alta., Canada, 2000. IEEE. ISBN 978-0-7803-5800-3. doi:10.1109/ASSPCC.2000.882463. URL http://ieeexplore.ieee.org/document/882463/.

Quanzeng Wang, Yangling Zhou, Pejman Ghassemi, Dwith Chenna, Michelle Chen, Jon Casamento, Joshua Pfefer, and David Mcbride. Facial and oral temperature data from a large set of human subject volunteers. URL https://physionet.org/content/face-oral-temp-data/.

Oren Wright, Yorie Nakahira, and José M. F. Moura. An Analytic Solution to Covariance Propagation in Neural Networks. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pages 4087–4095. PMLR, April 2024. URL https://proceedings.mlr.press/v238/wright24a.html.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E. Turner, José Miguel Hernández-Lobato, and Alexander L. Gaunt. Deterministic Variational Inference for Robust Bayesian Neural Networks, March 2019. URL http://arxiv.org/abs/1810.03958. arXiv:1810.03958 [cs].

Wilson Wu, Victor Lecomte, Michael Winer, George Robinson, Jacob Hilton, and Paul Christiano. Estimating the expected output of wide random MLPs more efficiently than sampling, May 2026. URL http://arxiv.org/abs/2605.05179. arXiv:2605.05179 [cs.LG].

## Supplementary Material

## Contents


**1 Introduction** **1**

**2 Exact, closed-form analytical expressions for mean and covariance** **3**

2.1 Our analytic moment-matching method $Y_{\mathrm{ana}}$ . . . . . 3

2.2 Targets: ground truth $Y_{\mathrm{true}}$ and pseudo-truth $Y_{\mathrm{pseudo}}$ . . . . . 5

**3 Explicit examples** **5**

**4 Second-order accuracy of moment matching** **6**

**5 Examples and applications** **7**

5.1 Random networks . . . . . 7

5.2 Weight uncertainty: variational Bayesian neural networks . . . . . 8

5.3 Stochastic activations . . . . . 9

**6 Limitations** **9**

**7 Extensions** **9**

7.1 Residual connections across multiple layers . . . . . 9

7.2 Kalman filtering . . . . . 10

7.3 Misclassification under Gaussian perturbations . . . . . 10

**8 Novelty and significance** **10**

**A Baselines** **20**

A.1 The mean-field approximation $Y_{\mathrm{mfa}}$ . . . . . 20

A.2 The linear approximation $Y_{\mathrm{lin}}$ . . . . . 20

A.3 The unscented approximation(s) $Y_{\mathrm{u95}}$ and $Y_{\mathrm{u02}}$ . . . . . 20

**B Preliminaries: Gaussian integrals** **20**

**C Derivation of uncertainty propagation formulas for probit activation** **24**

**D Derivation of uncertainty propagation formulas for GeLU activation** **25**

**E Derivation of uncertainty propagation formulas for ReLU activation** **29**

**F Derivation of uncertainty propagation formulas for Heaviside activation** **31**

**G Derivation of uncertainty propagation formulas for sine activation** **32**

**H Theoretical Guarantees** **34**

H.1 Single-layer smooth-distance expansion . . . . . 34
H.2 Layerwise recursion in smooth distance . . . . . 35

**I Supplement to Fig. 4.1** **36**

**J Supplement to §5.2** **37**
J.1 Results . . . . . 38

**K Supplement to §5.3** **44**

**L Supplement to §5.1** **45**
L.1 Summaries . . . . . 47

**M Comments on computational complexity** **49**

## List of Figures


4.1 Empirical small-covariance convergence for a fixed deep sine network. Methodology is described in Appendix I. . . . . 7

5.1 Comparison of goodness of approximation (lower KL divergence is better) for all random neural networks, grouped by approximation method, in the small (left) and large (right) input variance scenarios. Four non-finite results are omitted in each scenario. . . . . 8

5.2 KL divergence between ground-truth predictive distribution (by Monte Carlo) and approximations for two UCI regression datasets. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 9

J.1 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the Servo dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 38

J.2 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the liver disorders dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 39

J.3 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the Auto MPG dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 39

J.4 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the real estate valuation dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 40

J.5 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the forest fires dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 40

J.6 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the infrared thermography temperature dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 41

J.7 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the concrete compressive strength dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 41

J.8 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the airfoil self-noise dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 42

J.9 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the communities and crime dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 42

J.10 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the Parkinsons telemonitoring dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 43

J.11 KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the combined cycle power plant dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion. . . . . 43

K.1 Output distribution of a stochastic neural network, pseudo-true Normal distribution ("pseudo"), and layer-by-layer moment-matched Normal distribution ("analytic"). . 44

L.1 Comparison of goodness of approximation (lower KL divergence is better) for all random neural networks, grouped by approximation method, in the small input variance scenario. . . . . 47

L.2 Comparison of goodness of approximation (lower KL divergence is better) for all random neural networks, grouped by approximation method, in the medium input variance scenario. . . . . 47

L.3 Comparison of goodness of approximation (lower KL divergence is better) for all random neural networks, grouped by approximation method, in the large input variance scenario. . . . . 48

# A Baselines

## A.1 The mean-field approximation $Y_{\mathrm{mfa}}$

Following Huber [2020], Wagner et al. [2022], Akgül et al. [2025], we define the mean-field analytic approximation by assuming that neurons in the same hidden layer are independent:

$$\begin{aligned}
Y_{\mathrm{mfa}} &= Y^{\ell} \\
Y^k &= \mathcal{N}(\mu^k, \Sigma^k) && \forall k \in \{1 \dots \ell\} \\
\mu^k &= \mathbb{E}\, g(Y^{k-1}; A^k, b^k, C^k, d^k) \\
\Sigma^k_{ij} &= \begin{cases} \left[\operatorname{Cov} g(Y^{k-1}; A^k, b^k, C^k, d^k)\right]_{ij}, & i = j \\ 0, & \text{else} \end{cases} && \text{(mean-field assumption)} \\
Y^0 &= X
\end{aligned}$$

## A.2 The linear approximation $Y_{\mathrm{lin}}$

Following Titensky et al. [2018], Nagel and Huber [2022], Petersen et al. [2024], Jungmann et al. [2025], we define the linear approximation as the asymptotically Normal output distribution (pursuant to the delta method) in the limit of small input variance:

$$Y_{\mathrm{lin}} = \mathcal{N}(f(\mathbb{E}\, X), \nabla f(\mathbb{E}\, X) \operatorname{Cov} X \nabla f(\mathbb{E}\, X)^{\intercal})$$

## A.3 The unscented approximation(s) $Y_{\mathrm{u95}}$ and $Y_{\mathrm{u02}}$

An unscented transformation is a quadrature rule for approximating a probability measure on $\mathbb{R}^n$ by $2n + 1$ point masses whose locations (called sigma points) and weights satisfy certain first- and second-order moment matching conditions. This technique was developed to improve upon linearization in nonlinear Kalman filtering [Julier et al., 1995, Julier and Uhlmann, 1997, Julier et al., 2000, Julier, 2002, Wan and Van Der Merwe, 2000, Julier and Uhlmann, 2004].

There are (at least) two unscented transformations: a one-parameter family, which we refer to as Unscented'95; and a three-parameter family, which we refer to as Unscented'02.

- Unscented'95 [Julier et al., 1995, Julier and Uhlmann, 1997, Julier et al., 2000], used in Astudillo and Neto [2011], Abdelaziz et al. [2015], is a one-parameter family of unscented transformations, parameterized by a single shape hyperparameter $\kappa$.
- Unscented'02 [Julier, 2002, Wan and Van Der Merwe, 2000] is a three-parameter family of unscented transformations, parameterized by three shape hyperparameters $(\alpha, \beta, \kappa)$. This version, with default values tuned to be drastically more localized in $X$-space than Unscented'95, is usually called "the" unscented transformation in current filtering research [Jiang et al., 2025] and tooling [Ljung, 2025].

# B Preliminaries: Gaussian integrals

**Definition B.1.** The bivariate normal CDF $\Phi_2$ is defined by

$$\Phi_2(h, k; \rho) = \mathbb{P}\left[Z_1 \le h, Z_2 \le k\right],$$

where

$$\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right).$$

*Remark* B.2. While we consider it a valid "closed form" atom, the bivariate normal CDF can be a difficult transcendental function to evaluate. Many software packages such as SciPy [Wagner et al., 2022] implement the multivariate normal CDF by a (quasi-) Monte Carlo integration over $\mathbb{R}^n$, which is too expensive for our purposes. Furthermore, we frequently use the expression $\Phi(h, k; \rho) - \Phi(h, k; 0)$, which is vulnerable to cancellation error for extreme values of $h$, $k$, and $\rho$. To avoid this,

we compute the increment $\Delta\Phi_2(h,k;\rho) := \Phi_2(h,k;\rho) - \Phi_2(h,k;0)$ as the one-dimensional proper integral

$$\Delta\Phi_2(h,k;\rho) = \int_0^\rho \partial_{\rho'}\Phi_2(h,k;\rho')\,\mathrm{d}\rho',$$

using

$$\partial_\rho \Phi_2(h,k;\rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left(h^2+k^2-2\rho hk\right)},$$

a helpful identity found in Drezner and Wesolowsky [1990]. Substituting $\rho' = \sin\theta$ with $\theta = z\arcsin\rho$, the Jacobian $\cos\theta\,\mathrm{d}\theta$ cancels the factor $(1-\rho'^2)^{-1/2}$, giving

$$\begin{aligned}\Delta\Phi_2(h,k;\rho) &= \int_0^{\arcsin\rho} \frac{1}{2\pi}\exp\left[-\frac{h^2+k^2-2\sin\theta\,hk}{2\cos^2\theta}\right]\mathrm{d}\theta \\ &= \frac{\arcsin\rho}{2\pi}\int_0^1 \exp\left[-\frac{h^2+k^2-2\sin((\arcsin\rho)z)\,hk}{2\cos^2((\arcsin\rho)z)}\right]\mathrm{d}z.\end{aligned}$$

We apply 10-point Gauss–Legendre quadrature to this integrand, which is analytic on $[0,1]$ for every $|\rho| \le 1$.

**Definition B.3.** The bivariate Normal density is

$$\phi_2(h,k;\rho) = \frac{\partial^2}{\partial h\partial k}\,\Phi_2(h,k;\rho).$$

**Definition B.4.** The partial derivative of the joint CDF is

$$\Phi_{2;1}(h,k;\rho) = \frac{\partial}{\partial h}\,\Phi_2(h,k;\rho) \tag{4}$$

**Lemma B.5.** *The function $\Phi_{2;1}$ satisfies*

$$\Phi_{2;1}(h,k;\rho) = \phi(h)\Phi\left(\frac{k-\rho h}{\sqrt{1-\rho^2}}\right).$$

*Proof.* Letting $Z_1, Z_2$ be standard normal with correlation $\rho$, there is a probabilistic interpretation of $\Phi_{2;1}$:

$$\begin{aligned}\Phi_{2;1}(h,k;\rho) &= \underbrace{f_{Z_1}(h)}_{\text{marginal density}}\ \underbrace{\mathbb{P}\left(Z_2 \le k \mid Z_1 = h\right)}_{\text{conditional cdf}} \\ &= \phi(h)\Phi\left(\frac{k-\rho h}{\sqrt{1-\rho^2}}\right).\end{aligned}$$

□

**Lemma B.6** (Multivariate Stein's lemma)**.** *Let $(X_1,\dots,X_n)$ be a multivariate normal random vector. Then*

$$\mathbb{E}(X_1 - \mathbb{E}\,X_1)f(X_1,\dots,X_n) = \sum_{i=1}^n \mathrm{Cov}(X_1,X_i)\,\mathbb{E}\left[\partial_{X_i} f(X_1,\dots,X_n)\right]$$

**Lemma B.7** (Stein simplification of $L_\sigma$)**.** *Let $L_\sigma$ be defined as in Definition 2.3, and suppose that $\sigma$ is differentiable. Then*

$$L_\sigma(\mu_1;\nu_{11},\nu_{22},\nu_{12}) = \nu_{12}\,\mathbb{E}\,\sigma'(Z_1), \qquad Z_1 \sim \mathcal{N}(\mu_1,\nu_{11}).$$

*Proof.* Straightforward application of Lemma B.6. □

**Lemma B.8** (Univariate integrals)**.** *If $Z \sim \mathcal{N}(\mu, \nu)$, then*

$$\mathbb{E}\,\Phi(Z) = \Phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) \tag{5a}$$

$$\mathbb{E}\,\phi(Z) = \frac{1}{\sqrt{1+\nu}}\phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) \tag{5b}$$

$$\mathbb{E}\,Z\Phi(Z) = \frac{\mu}{(1+\nu)^{3/2}}\phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) \tag{5c}$$

*Proof of* (5a). Introducing an independent $Z \sim \mathcal{N}(0,1)$, we have

$$\begin{aligned}
\mathbb{E}\,\Phi(Z) &= \mathbb{E}\,\mathbb{P}\left[Z \le X \mid X\right] && (6)\\
&= \mathbb{P}\left[Z \le X\right] && \text{(by the law of total probability)}\\
&= \mathbb{P}\left[Z - X \le 0\right] && (7)
\end{aligned}$$

We conclude by noting that the random variable $Z - X$ has a Normal distribution with mean $-\mu$ and variance $1 + \nu$. □

*Proof of* (5b). We use $\phi = \Phi'$.

$$\begin{aligned}
\mathbb{E}\,\phi(Z) &= \mathbb{E}\,\frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0}\Phi(Z+t) && (8)\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0}\mathbb{E}\,\Phi(Z+t) && \text{(dominated convergence theorem)}\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0}\Phi\left(\frac{\mu+t}{\sqrt{1+\nu}}\right) && \text{(by (5a))}
\end{aligned}$$

□

*Proof of* (5c). We use the Gaussian ODE

$$\phi'(x) + x\phi(x) = 0. \tag{9}$$

Centering and applying Lemma B.6,

$$\begin{aligned}
\mathbb{E}\,Z\phi(Z) &= \mathbb{E}(Z-\mu)\phi(Z) + \mu\,\mathbb{E}\,\phi(Z) && \text{(centering)}\\
&= \nu\,\mathbb{E}\,\phi'(Z) + \mu\,\mathbb{E}\,\phi(Z) && \text{(Stein's)}\\
&= -\nu\,\mathbb{E}\,Z\Phi(Z) + \mu\,\mathbb{E}\,\phi(Z) && \text{(Gaussian ODE)}
\end{aligned}$$

Collecting like terms and solving for (I),

$$\begin{aligned}
(1+\nu)\,\mathbb{E}\,Z\phi(Z) &= \mu\,\mathbb{E}\,\phi(Z) && (10)\\
\mathbb{E}\,Z\phi(Z) &= \frac{\mu}{1+\nu}\,\mathbb{E}\,\phi(Z) && (11)\\
&= \frac{\mu}{(1+\nu)^{3/2}}\phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) && \text{(by (5b))}
\end{aligned}$$

□

**Lemma B.9** (Bivariate integrals)**.** *Let*

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \nu_{11} & \nu_{12} \\ \nu_{12} & \nu_{22} \end{pmatrix}\right).$$

*Then*

$$\mathbb{E}\,\Phi(X_1)\Phi(X_2) = \Phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \tag{12a}$$

$$\mathbb{E}\,\phi(X_1)\Phi(X_2) = \frac{1}{\sqrt{1+\nu_{11}}}\Phi_{2;1}\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \tag{12b}$$

$$\mathbb{E}\,\phi(X_1)\phi(X_2) = \frac{1}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \tag{12c}$$

$$\mathbb{E}\,\phi'(X_1)\Phi(X_2) = \frac{-\mu_1}{1+\nu_{11}}\,\mathbb{E}\,\phi(X_1)\Phi(X_2) + \frac{-\nu_{12}}{1+\nu_{11}}\,\mathbb{E}\,\phi(X_1)\phi(X_2) \tag{12d}$$

*Proof of* (12a). Introduce independent $Z_1, Z_2 \sim \mathcal{N}(0,1)$. Then

$$\begin{aligned}
\mathbb{E}\,\Phi(X_1)\Phi(X_2) &= \mathbb{E}\,\mathbb{P}\left[Z_1 \le X_1 \mid X_1\right]\mathbb{P}\left[Z_2 \le X_2 \mid X_2\right] && (13)\\
&= \mathbb{E}\,\mathbb{P}\left[Z_1 \le X_1, Z_2 \le X_2 \mid X_1, X_2\right] && \text{(independence)}\\
&= \mathbb{P}\left[Z_1 \le X_1, Z_2 \le X_2\right] && \text{(by the law of total probability)}\\
&= \mathbb{P}\left[Z_1 - X_1 \le 0, Z_2 - X_2 \le 0\right]. && (14)
\end{aligned}$$

We conclude by using the fact that $(Z_1 - X_1, Z_2 - X_2)$ is jointly Normal with distribution

$$\begin{pmatrix} Z_1 - X_1 \\ Z_2 - X_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} -\mu_1 \\ -\mu_2 \end{pmatrix}, \begin{pmatrix} 1+\nu_{11} & \nu_{12} \\ \nu_{12} & 1+\nu_{22} \end{pmatrix}\right). \tag{15}$$

□

*Proof of* (12b). Using $\phi = \Phi'$, we have

$$\begin{aligned}
&\mathbb{E}\,\phi(X_1)\Phi(X_2)\\
&\quad= \mathbb{E}\,\frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0}\Phi(X_1+t)\Phi(X_2) && (16)\\
&\quad= \frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0}\mathbb{E}\,\Phi(X_1+t)\Phi(X_2) && \text{(dominated convergence theorem)}\\
&\quad= \frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0}\mathbb{P}\left(Z_1 - X_1 - t \le 0, Z_2 - X_2 \le 0\right) && \text{(introducing } Z_1, Z_2 \sim \mathcal{N}(0,1))\\
&\quad= \frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0}\Phi_2\left(\frac{\mu_1+t}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) && (17)\\
&\quad= \frac{1}{\sqrt{1+\nu_{11}}}\Phi_{2;1}\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) && (18)
\end{aligned}$$

□

*Proof of* (12c). Using $\phi = \Phi'$ in both terms,

$$\mathbb{E}\,\phi(X_1)\phi(X_2) = \mathbb{E}\,\frac{\partial^2}{\partial t \partial s}\bigg|_{t=0,s=0} \Phi(X_1+t)\Phi(X_2+s) \tag{19}$$

$$= \frac{\partial^2}{\partial t \partial s}\bigg|_{t=0,s=0} \mathbb{E}\,\Phi(X_1+t)\Phi(X_2+s) \qquad \text{(dominated convergence theorem)}$$

$$= \frac{\partial^2}{\partial t \partial s}\bigg|_{t=0,s=0} \mathbb{P}\left(Z_1 - X_1 - t \le 0, Z_2 - X_2 - s \le 0\right)$$

(introducing $Z_1, Z_2 \sim \mathcal{N}(0,1)$)

$$= \frac{\partial^2}{\partial t \partial s}\bigg|_{t=0,s=0} \Phi_2\left(\frac{\mu_1+t}{\sqrt{1+\nu_{11}}}, \frac{\mu_2+s}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \tag{20}$$

$$= \frac{1}{\sqrt{1+\nu_{11}}\sqrt{1+\nu_{22}}}\phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right)$$

(pdf is derivative of cdf)

□

*Proof of* (12d). We use the Gaussian ODE

$$\phi'(x) + x\phi(x) = 0. \tag{21}$$

Using this fact,

$$\mathbb{E}\,\phi'(X_1)\Phi(X_2) = -\,\mathbb{E}\,X_1\phi(X_1)\Phi(X_2) \tag{22}$$

$$= -\mu_1\,\mathbb{E}\,\phi(X_1)\Phi(X_2) - \mathbb{E}(X_1-\mu_1)\phi(X_1)\Phi(X_2) \qquad \text{(centering)}$$

$$= -\mu_1\,\mathbb{E}\,\phi(X_1)\Phi(X_2) - \nu_{11}\underbrace{\mathbb{E}\,\phi'(X_1)\Phi(X_2)}_{\text{same as LHS}} - \nu_{12}\,\mathbb{E}\,\phi(X_1)\phi(X_2)$$

(Lemma B.6)

Collecting like terms and solving for $\mathbb{E}\,\phi'(X_1)\Phi(X_2)$,,

$$\mathbb{E}\,\phi'(X_1)\Phi(X_2) = \frac{-\mu_1}{1+\nu_{11}}\,\mathbb{E}\,\phi(X_1)\Phi(X_2) + \frac{-\nu_{12}}{1+\nu_{11}}\,\mathbb{E}\,\phi(X_1)\phi(X_2). \tag{23}$$

□

# C Derivation of uncertainty propagation formulas for probit activation

In this appendix, we derive the $M_\sigma$, $K_\sigma$, and $L_\sigma$ functions (Def. 2.3) for the normal CDF activation function

$$\sigma(x) = \sqrt{\frac{2}{\pi}}\int_0^x e^{-\frac{1}{2}u^2}du = 2\Phi(x) - 1, \quad \Phi(x) = \mathbb{P}_{Z\sim\mathcal{N}(0,1)}(Z \le x) \tag{24}$$

**Lemma C.1.** *Let $M_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (24)*.*

$$M_\sigma(\mu;\nu) = \sigma\left(\frac{\mu}{\sqrt{1+\nu}}\right)$$

*Proof.* Let $X \sim \mathcal{N}(\mu,\nu)$.

$$\mathbb{E}\,\sigma(X) = -1 + 2\,\mathbb{E}\,\Phi\left(X\right) \tag{25}$$

This is a direct application of Lemma B.8, (5a). □

**Lemma C.2.** *Let $K_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (24)*.*

$$K(\mu_1, \mu_2; \nu_{11}, \nu_{22}, \nu_{12}) = 4\ \Phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \rho'\right)\Bigg|_{\rho'=0}^{\rho'=\frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}},$$

*where $\Phi_2$ is the bivariate normal CDF.*

*Proof.* The covariance may be expressed as

$$\begin{aligned}\operatorname{Cov}(\sigma(X_1), \sigma(X_2)) &= 4 \operatorname{Cov}(\Phi(X_1), \Phi(X_2)) &(26)\\ &= 4\,\mathbb{E}\,\Phi(X_1)\Phi(X_2) - 4\,\mathbb{E}\,\Phi(X_1)\,\mathbb{E}\,\Phi(X_2) &(27)\end{aligned}$$

Now apply Lemma B.9 to the first term. □

**Lemma C.3.** *Let $L_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (24)*. Then*

$$L_\sigma(\mu_1; \nu_{11}, \nu_{22}, \nu_{12}) = 2\frac{\nu_{12}}{\sqrt{1+\nu_{11}}}\phi\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}\right).$$

*Proof.* Using Lemma B.7, we have

$$\begin{aligned}L_\sigma(\mu_1; \nu_{11}, \nu_{22}, \nu_{12}) &= \nu_{12}\,\mathbb{E}\,\sigma'(X_1) &(28)\\ &= 2\nu_{12}\,\mathbb{E}\,\phi(X_1) &(29)\end{aligned}$$

where $\phi = \Phi'$. By Lemma B.8, applied to $X_1 \sim \mathcal{N}(\mu_1, \nu_{11})$,

$$\mathbb{E}\,\phi(X_1) = \frac{1}{\sqrt{1+\nu_{11}}}\phi\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}\right). \tag{30}$$

□

# D Derivation of uncertainty propagation formulas for GeLU activation

In this appendix, we derive the $M_\sigma$, $K_\sigma$ and $L_\sigma$ functions (Def. 2.3) for the GeLU activation function

$$\sigma(x) = x\Phi(x). \tag{31}$$

**Lemma D.1.** *Let $M_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (31)*. Then*

$$M_\sigma(\mu; \nu) = \frac{\nu}{\sqrt{1+\nu}}\phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) + \mu\Phi\left(\frac{\mu}{\sqrt{1+\nu}}\right).$$

*Proof.* Let $X \sim \mathcal{N}(\mu, \nu)$ and $Z \sim \mathcal{N}(0, 1)$. Centering $X$,

$$\begin{aligned}M_\sigma(\mu; \nu) = \mathbb{E}\,X\Phi(X) &= \mathbb{E}(X-\mu)\Phi(X) + \mu\,\mathbb{E}\,\Phi(X)\\ &= \mathbb{E}(X-\mu)\Phi(X) + \mu\Phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) && \text{(using Lemma B.8, (5a))}\\ &= \nu\,\mathbb{E}\,\phi(X) + \mu\Phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) && \text{(using Lemma B.6)}\\ &= \frac{\nu}{\sqrt{1+\nu}}\phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) + \mu\Phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) && \text{(using Lemma B.8, (5b))}\end{aligned}$$

□

**Lemma D.2.** *Let $K_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (31)*. Then*

$$
\begin{aligned}
&K_\sigma(\mu_1,\mu_2;\nu_{11},\nu_{22},\nu_{12}) \\
&= \left(\mu_1\nu_{12}+\mu_2\nu_{11}-\frac{\mu_1\nu_{12}\nu_{11}}{1+\nu_{11}}\right)\frac{1}{\sqrt{1+\nu_{11}}}\Phi_{2;1}\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}},\frac{\mu_2}{\sqrt{1+\nu_{22}}};\frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&+\left(\mu_2\nu_{12}+\mu_1\nu_{22}-\frac{\mu_2\nu_{12}\nu_{22}}{1+\nu_{22}}\right)\frac{1}{\sqrt{1+\nu_{22}}}\Phi_{2;1}\left(\frac{\mu_2}{\sqrt{1+\nu_{22}}},\frac{\mu_1}{\sqrt{1+\nu_{11}}};\frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&\quad+\frac{\nu_{11}\nu_{22}+\nu_{12}^2\left(1-\frac{\nu_{11}}{1+\nu_{11}}-\frac{\nu_{22}}{1+\nu_{22}}\right)}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}},\frac{\mu_2}{\sqrt{1+\nu_{22}}};\frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&\qquad+(\mu_1\mu_2+\nu_{12})\,\Phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}},\frac{\mu_2}{\sqrt{1+\nu_{22}}};\frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&\qquad\qquad-M_\sigma(\mu_1;\nu_{11})M_\sigma(\mu_2;\nu_{22})
\end{aligned}
$$

*Proof.* Let

$$
\begin{pmatrix}X_1\\X_2\end{pmatrix}\sim\mathcal{N}\left(\begin{pmatrix}\mu_1\\\mu_2\end{pmatrix},\begin{pmatrix}\nu_{11}&\nu_{12}\\\nu_{12}&\nu_{22}\end{pmatrix}\right). \tag{32}
$$

Our strategy is to use the formula

$$
\operatorname{Cov}(\sigma(X_1),\sigma(X_2))=\mathbb{E}\,\sigma(X_1)\sigma(X_2)-\mathbb{E}\,\sigma(X_1)\,\mathbb{E}\,\sigma(X_2)
$$

and use Lemma D.1 for the second term. For the cross term, we first center the random variables:

$$
\begin{aligned}
\mathbb{E}\,X_1\Phi(X_1)X_2\Phi(X_2)=&\underbrace{\mathbb{E}(X_1-\mu_1)\Phi(X_1)(X_2-\mu_2)\Phi(X_2)}_{\text{(I)}}\\
&+\underbrace{\mathbb{E}\,\mu_1\Phi(X_1)(X_2-\mu_2)\Phi(X_2)}_{\text{(II)}}\\
&+\underbrace{\mathbb{E}(X_1-\mu_1)\Phi(X_1)\mu_2\Phi(X_2)}_{\text{(III)}}\\
&+\underbrace{\mathbb{E}\,\mu_1\Phi(X_1)\mu_2\Phi(X_2)}_{\text{(IV)}}
\end{aligned}\tag{33}
$$

For term (I) of (33),

$$
\begin{aligned}
\text{(I)}&=\mathbb{E}(X_1-\mu_1)\Phi(X_1)(X_2-\mu_2)\Phi(X_2) && (34)\\
&=\sum_{i=1}^{2}\operatorname{Cov}(X_1,X_i)\,\mathbb{E}\,\partial_{X_i}\left(\Phi(X_1)(X_2-\mu_2)\Phi(X_2)\right) && \text{(Lemma B.6)}\\
&=\underbrace{\nu_{11}\,\mathbb{E}\,\phi(X_1)(X_2-\mu_2)\Phi(X_2)}_{\text{(I.a)}}+\nu_{12}\,\mathbb{E}\,\Phi(X_1)\Phi(X_2)+\underbrace{\nu_{12}\,\mathbb{E}\,\Phi(X_1)(X_2-\mu_2)\phi(X_2)}_{\text{(I.b)}} && (35)
\end{aligned}
$$

We apply Lemma B.6 to (I.a) and (I.b) to get

$$
\begin{aligned}
\text{(I.a)}+\text{(I.b)}&=\nu_{11}\,\mathbb{E}\,\phi(X_1)(X_2-\mu_2)\Phi(X_2)+\nu_{12}\,\mathbb{E}\,\Phi(X_1)(X_2-\mu_2)\phi(X_2) && (36)\\
&=\nu_{11}\,\mathbb{E}\left[\nu_{12}\phi'(X_1)\Phi(X_2)+\nu_{22}\phi(X_1)\phi(X_2)\right]\\
&\quad+\nu_{12}\,\mathbb{E}\left[\nu_{12}\phi(X_1)\phi(X_2)+\nu_{22}\Phi(X_1)\phi'(X_2)\right] && (37)\\
&=\nu_{12}\nu_{11}\underbrace{\mathbb{E}\,\phi'(X_1)\Phi(X_2)}_{\text{(I.c)}}+\nu_{12}\nu_{22}\underbrace{\mathbb{E}\,\Phi(X_1)\phi'(X_2)}_{\text{(I.d)}}\\
&\quad+\left(\nu_{11}\nu_{22}+\nu_{12}^2\right)\mathbb{E}\,\phi(X_1)\phi(X_2) && (38)
\end{aligned}
$$

By Lemma B.9, (12d), we have

$$\text{(I.c)} = \frac{-\mu_1}{1+\nu_{11}}\,\mathbb{E}\,\phi(X_1)\Phi(X_2) + \frac{-\nu_{12}}{1+\nu_{11}}\,\mathbb{E}\,\phi(X_1)\phi(X_2)$$
$$\text{(I.d)} = \frac{-\mu_2}{1+\nu_{22}}\,\mathbb{E}\,\Phi(X_1)\phi(X_2) + \frac{-\nu_{12}}{1+\nu_{22}}\,\mathbb{E}\,\phi(X_1)\phi(X_2)$$

Combining these last two equations with (35) and (38), (I) becomes

$$\begin{aligned}\text{(I)} = \nu_{12}\nu_{11}&\left(\frac{-\mu_1}{1+\nu_{11}}\,\mathbb{E}\,\phi(X_1)\Phi(X_2) + \frac{-\nu_{12}}{1+\nu_{11}}\,\mathbb{E}\,\phi(X_1)\phi(X_2)\right)\\ &+ \nu_{12}\nu_{22}\left(\frac{-\mu_2}{1+\nu_{22}}\,\mathbb{E}\,\Phi(X_1)\phi(X_2) + \frac{-\nu_{12}}{1+\nu_{22}}\,\mathbb{E}\,\phi(X_1)\phi(X_2)\right)\\ &+ \left(\nu_{11}\nu_{22}+\nu_{12}^2\right)\mathbb{E}\,\phi(X_1)\phi(X_2)\\ &+ \nu_{12}\,\mathbb{E}\,\Phi(X_1)\Phi(X_2)\end{aligned} \tag{39}$$

which simplifies to

$$\begin{aligned}\text{(I)} = \frac{-\mu_1\nu_{12}\nu_{11}}{1+\nu_{11}}\,\mathbb{E}\,\phi(X_1)\Phi(X_2) &+ \frac{-\mu_2\nu_{12}\nu_{22}}{1+\nu_{22}}\,\mathbb{E}\,\Phi(X_1)\phi(X_2)\\ &+ \left[\nu_{11}\nu_{22}+\nu_{12}^2\left(1-\frac{\nu_{11}}{1+\nu_{11}}-\frac{\nu_{22}}{1+\nu_{22}}\right)\right]\mathbb{E}\,\phi(X_1)\phi(X_2)\\ &+ \nu_{12}\,\mathbb{E}\,\Phi(X_1)\Phi(X_2)\end{aligned} \tag{40}$$

For terms (II-III) of (33), we apply Lemma B.6 to get

$$\text{(II)} = \mathbb{E}\,\mu_1\Phi(X_1)(X_2-\mu_2)\Phi(X_2) = \mu_1\left(\nu_{12}\,\mathbb{E}\,\phi(X_1)\Phi(X_2) + \nu_{22}\,\mathbb{E}\,\Phi(X_1)\phi(X_2)\right)$$
$$\text{(III)} = \mathbb{E}(X_1-\mu_1)\Phi(X_1)\mu_2\Phi(X_2) = \mu_2\left(\nu_{11}\,\mathbb{E}\,\phi(X_1)\Phi(X_2) + \nu_{12}\,\mathbb{E}\,\Phi(X_1)\phi(X_2)\right)$$

Now (33) becomes:

$$\begin{aligned}&\mathbb{E}\,X_1\Phi(X_1)X_2\Phi(X_2)\\ &\quad= \left(\mu_1\nu_{12}+\mu_2\nu_{11}-\frac{\mu_1\nu_{12}\nu_{11}}{1+\nu_{11}}\right)\underbrace{\mathbb{E}\,\phi(X_1)\Phi(X_2)}_{\text{(V.a)}}\\ &\quad+ \left(\mu_2\nu_{12}+\mu_1\nu_{22}-\frac{\mu_2\nu_{12}\nu_{22}}{1+\nu_{22}}\right)\underbrace{\mathbb{E}\,\Phi(X_1)\phi(X_2)}_{\text{(V.b)}}\\ &\quad+ \left[\nu_{11}\nu_{22}+\nu_{12}^2\left(1-\frac{\nu_{11}}{1+\nu_{11}}-\frac{\nu_{22}}{1+\nu_{22}}\right)\right]\underbrace{\mathbb{E}\,\phi(X_1)\phi(X_2)}_{\text{(V.c)}}\\ &\quad+ (\mu_1\mu_2+\nu_{12})\underbrace{\mathbb{E}\,\Phi(X_1)\Phi(X_2)}_{\text{(V.d)}}\end{aligned} \tag{41}$$

By Lemma B.9, (12b),

$$\text{(V.a)} = \frac{1}{\sqrt{1+\nu_{11}}}\Phi_{2;1}\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \tag{42}$$

and

$$\text{(V.b)} = \frac{1}{\sqrt{1+\nu_{22}}}\Phi_{2;1}\left(\frac{\mu_2}{\sqrt{1+\nu_{22}}}, \frac{\mu_1}{\sqrt{1+\nu_{11}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right). \tag{43}$$

By Lemma B.9, (12c),

$$\text{(V.c)} = \frac{1}{\sqrt{1+\nu_{11}}\sqrt{1+\nu_{22}}}\phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right). \tag{44}$$

By Lemma B.9, (12a),

$$\text{(V.d)} = \Phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \tag{45}$$

The final form of (41) is

$$\begin{aligned}
&\mathbb{E}\, X_1\Phi(X_1)X_2\Phi(X_2) \\
&= \left(\mu_1\nu_{12} + \mu_2\nu_{11} - \frac{\mu_1\nu_{12}\nu_{11}}{1+\nu_{11}}\right)\frac{1}{\sqrt{1+\nu_{11}}}\Phi_{2;1}\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&+ \left(\mu_2\nu_{12} + \mu_1\nu_{22} - \frac{\mu_2\nu_{12}\nu_{22}}{1+\nu_{22}}\right)\frac{1}{\sqrt{1+\nu_{22}}}\Phi_{2;1}\left(\frac{\mu_2}{\sqrt{1+\nu_{22}}}, \frac{\mu_1}{\sqrt{1+\nu_{11}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&\quad + \frac{\nu_{11}\nu_{22} + \nu_{12}^2\left(1 - \frac{\nu_{11}}{1+\nu_{11}} - \frac{\nu_{22}}{1+\nu_{22}}\right)}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right) \\
&\qquad + (\mu_1\mu_2 + \nu_{12})\,\Phi_2\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}, \frac{\mu_2}{\sqrt{1+\nu_{22}}}; \frac{\nu_{12}}{\sqrt{(1+\nu_{11})(1+\nu_{22})}}\right)
\end{aligned} \tag{46}$$

The conclusion follows from subtracting $M(\mu_1;\nu_{11})M(\mu_2;\nu_{22})$. □

**Lemma D.3.** *Let $L_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (31)*. Then*

$$L_\sigma(\mu_1;\nu_{11},\nu_{22},\nu_{12}) = \nu_{12}\frac{\mu_1}{(1+\nu_{11})^{3/2}}\phi\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}\right) + \nu_{12}\Phi\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}\right). \tag{47}$$

*Proof.* By Lemma B.7,

$$\begin{aligned}
L_\sigma(\mu_1;\nu_{11},\nu_{22},\nu_{12}) &= \nu_{12}\,\mathbb{E}\,\sigma'(X_1), && X_1 \sim \mathcal{N}(\mu_1,\nu_{11}) \\
&= \nu_{12}\underbrace{\mathbb{E}\,X_1\phi(X_1)}_{\text{(I)}} + \nu_{12}\underbrace{\mathbb{E}\,\Phi(X_1)}_{\text{(II)}} && \text{(product rule)}
\end{aligned}$$

By Lemma B.8, (5c)

$$\text{(I)} = \frac{\mu_1}{(1+\nu_{11})^{3/2}}\phi\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}\right). \tag{48}$$

Term (II) is given by

$$\text{(II)} = \mathbb{E}\,\Phi(X_1) = \Phi\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}\right) \tag{49}$$

Applying terms (I)–(II),

$$L_\sigma(\mu_1;\nu_{11},\nu_{22},\nu_{12}) = \nu_{12}\frac{\mu_1}{(1+\nu_{11})^{3/2}}\phi\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}\right) + \nu_{12}\Phi\left(\frac{\mu_1}{\sqrt{1+\nu_{11}}}\right) \tag{50}$$

□

## E Derivation of uncertainty propagation formulas for ReLU activation

In this appendix, we derive the uncertainty propagation formulas for ReLU activation

$$\sigma(x) = \max(0, x) \tag{51}$$

using the identity

$$\mathrm{ReLU}(x) = \lim_{\alpha\to\infty} \alpha^{-1} \,\mathrm{GeLU}(\alpha x). \tag{52}$$

See Muthén [1990] for another way to derive $M$ and $K$.

**Lemma E.1.** *Let $M_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (51)*. Then*

$$M_\sigma(\mu;\nu) = \sqrt{\nu}\phi\left(\frac{\mu}{\sqrt{\nu}}\right) + \mu\Phi\left(\frac{\mu}{\sqrt{\nu}}\right).$$

*Proof.* By the dominated convergence theorem,

$$\begin{aligned}
M_\sigma(\mu;\nu) &= \lim_{\alpha\to\infty} \alpha^{-1}\,\mathbb{E}\, M_{\mathrm{GeLU}}(\alpha\mu;\alpha^2\nu) && (53)\\
&= \lim_{\alpha\to\infty} \alpha^{-1}\left[\frac{\alpha^2\nu}{\sqrt{1+\alpha^2\nu}}\phi\left(\frac{\alpha\mu}{\sqrt{1+\alpha^2\nu}}\right) + \mu\Phi\left(\frac{\alpha\mu}{\sqrt{1+\alpha^2\nu}}\right)\right] && \text{(by Lemma D.1)}\\
&= \sqrt{\nu}\phi\left(\frac{\mu}{\sqrt{\nu}}\right) + \mu\Phi\left(\frac{\mu}{\sqrt{\nu}}\right) && (54)
\end{aligned}$$

□

**Lemma E.2.** *Let $K_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (51)*. Then*

$$\begin{aligned}
K_\sigma(\mu_1,\mu_2;\nu_{11},\nu_{22},\nu_{12}) = {}& \mu_2\sqrt{\nu_{11}}\Phi_{2;1}\left(\frac{\mu_1}{\sqrt{\nu_{11}}}, \frac{\mu_2}{\sqrt{\nu_{22}}}; \frac{\nu_{12}}{\sqrt{\nu_{11}\nu_{22}}}\right)\\
&+ \mu_1\sqrt{\nu_{22}}\Phi_{2;1}\left(\frac{\mu_2}{\sqrt{\nu_{22}}}, \frac{\mu_1}{\sqrt{\nu_{11}}}; \frac{\nu_{12}}{\sqrt{\nu_{11}\nu_{22}}}\right)\\
&+ \left(\sqrt{\nu_{11}\nu_{22}} - \frac{\nu_{12}^2}{\sqrt{\nu_{11}\nu_{22}}}\right)\phi_2\left(\frac{\mu_1}{\sqrt{\nu_{11}}}, \frac{\mu_2}{\sqrt{\nu_{22}}}; \frac{\nu_{12}}{\sqrt{\nu_{11}\nu_{22}}}\right)\\
&+ (\mu_1\mu_2 + \nu_{12})\,\Phi_2\left(\frac{\mu_1}{\sqrt{\nu_{11}}}, \frac{\mu_2}{\sqrt{\nu_{22}}}; \frac{\nu_{12}}{\sqrt{\nu_{11}\nu_{22}}}\right)\\
&- M_\sigma(\mu_1;\nu_{11})M_\sigma(\mu_2;\nu_{22})
\end{aligned}$$

*Proof.* By the dominated convergence theorem,

$$\begin{aligned}
\mathbb{E}\,\sigma(X_1)\sigma(X_2) &= \mathbb{E}\lim_{\alpha\to\infty} \alpha^{-2}\,\mathrm{GeLU}(\alpha X_1)\,\mathrm{GeLU}(\alpha X_2) && (55)\\
&= \lim_{\alpha\to\infty} \alpha^{-2}\,\mathbb{E}\,\mathrm{GeLU}(\alpha X_1)\,\mathrm{GeLU}(\alpha X_2) && (56)
\end{aligned}$$

To compute $\alpha^{-2}\,\mathbb{E}\,\mathrm{GeLU}(\alpha X_1)\,\mathrm{GeLU}(\alpha X_2)$, we use (46) (Proof of Lemma D.2) while scaling $\mu$s by $\alpha$ and $\nu$s by $\alpha^2$.

$$
\begin{aligned}
&\alpha^{-2}\,\mathbb{E}\,\mathrm{GeLU}(\alpha X_1)\,\mathrm{GeLU}(\alpha X_2) \\
&\quad= \frac{\alpha\mu_1\nu_{12} + \alpha\mu_2\nu_{11} - \alpha^3\frac{\mu_1\nu_{12}\nu_{11}}{1+\alpha^2\nu_{11}}}{\sqrt{1+\alpha^2\nu_{11}}} \\
&\qquad \Phi_{2;1}\left(\frac{\alpha\mu_1}{\sqrt{1+\alpha^2\nu_{11}}}, \frac{\alpha\mu_2}{\sqrt{1+\alpha^2\nu_{22}}}; \frac{\alpha^2\nu_{12}}{\sqrt{(1+\alpha^2\nu_{11})(1+\alpha^2\nu_{22})}}\right) \\
&\qquad + \frac{\alpha\mu_2\nu_{12} + \alpha\mu_1\nu_{22} - \alpha^3\frac{\mu_2\nu_{12}\nu_{22}}{1+\alpha^2\nu_{22}}}{\sqrt{1+\alpha^2\nu_{22}}} \\
&\qquad \Phi_{2;1}\left(\frac{\alpha\mu_2}{\sqrt{1+\alpha^2\nu_{22}}}, \frac{\alpha\mu_1}{\sqrt{1+\alpha^2\nu_{11}}}; \frac{\alpha^2\nu_{12}}{\sqrt{(1+\alpha^2\nu_{11})(1+\alpha^2\nu_{22})}}\right) \\
&\qquad + \frac{\alpha^2\nu_{11}\nu_{22} + \alpha^2\nu_{12}^2\left(1 - \frac{\alpha^2\nu_{11}}{1+\alpha^2\nu_{11}} - \frac{\alpha^2\nu_{22}}{1+\alpha^2\nu_{22}}\right)}{\sqrt{(1+\alpha^2\nu_{11})(1+\alpha^2\nu_{22})}} \\
&\qquad \phi_2\left(\frac{\alpha\mu_1}{\sqrt{1+\alpha^2\nu_{11}}}, \frac{\alpha\mu_2}{\sqrt{1+\alpha^2\nu_{22}}}; \frac{\alpha^2\nu_{12}}{\sqrt{(1+\alpha^2\nu_{11})(1+\alpha^2\nu_{22})}}\right) \\
&\qquad + (\mu_1\mu_2 + \nu_{12})\,\Phi_2\left(\frac{\alpha\mu_1}{\sqrt{1+\alpha^2\nu_{11}}}, \frac{\alpha\mu_2}{\sqrt{1+\alpha^2\nu_{22}}}; \frac{\alpha^2\nu_{12}}{\sqrt{(1+\alpha^2\nu_{11})(1+\alpha^2\nu_{22})}}\right)
\end{aligned} \tag{57}
$$

Taking $\alpha \to \infty$,

$$
\begin{aligned}
\mathbb{E}\,\sigma(X_1)\sigma(X_2) &= \lim_{\alpha\to\infty} \alpha^{-2}\,\mathbb{E}\,\mathrm{GeLU}(\alpha X_1)\,\mathrm{GeLU}(\alpha X_2) \\
&= \mu_2\sqrt{\nu_{11}}\Phi_{2;1}\left(\frac{\mu_1}{\sqrt{\nu_{11}}}, \frac{\mu_2}{\sqrt{\nu_{22}}}; \frac{\nu_{12}}{\sqrt{\nu_{11}\nu_{22}}}\right) \\
&\quad + \mu_1\sqrt{\nu_{22}}\Phi_{2;1}\left(\frac{\mu_2}{\sqrt{\nu_{22}}}, \frac{\mu_1}{\sqrt{\nu_{11}}}; \frac{\nu_{12}}{\sqrt{\nu_{11}\nu_{22}}}\right) \\
&\quad + \left(\sqrt{\nu_{11}\nu_{22}} - \frac{\nu_{12}^2}{\sqrt{\nu_{11}\nu_{22}}}\right)\phi_2\left(\frac{\mu_1}{\sqrt{\nu_{11}}}, \frac{\mu_2}{\sqrt{\nu_{22}}}; \frac{\nu_{12}}{\sqrt{\nu_{11}\nu_{22}}}\right) \\
&\quad + (\mu_1\mu_2 + \nu_{12})\,\Phi_2\left(\frac{\mu_1}{\sqrt{\nu_{11}}}, \frac{\mu_2}{\sqrt{\nu_{22}}}; \frac{\nu_{12}}{\sqrt{\nu_{11}\nu_{22}}}\right)
\end{aligned} \tag{58}
$$

The conclusion follows after subtracting $\mathbb{E}\,\sigma(X_1)\,\mathbb{E}\,\sigma(X_2)$. □

**Lemma E.3.** *Let $L_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (51)*. Then*

$$
L_\sigma(\mu_1; \nu_{11}, \nu_{22}, \nu_{12}) = \nu_{12}\Phi\left(\frac{\mu_1}{\sqrt{\nu_{11}}}\right).
$$

*Proof.* By the dominated convergence theorem,

$$
\begin{aligned}
&L_\sigma(\mu_1; \nu_{11}, \nu_{22}, \nu_{12}) \\
&\quad= \lim_{\alpha\to\infty} \alpha^{-2} L_{\mathrm{GeLU}}(\alpha\mu_1; \alpha^2\nu_{11}, \alpha^2\nu_{22}, \alpha^2\nu_{12}) && (59) \\
&\quad= \lim_{\alpha\to\infty} \alpha^{-2}\left[\alpha^2\nu_{12}\frac{\alpha\mu_1}{(1+\alpha^2\nu_{11})^{3/2}}\phi\left(\frac{\alpha\mu_1}{\sqrt{1+\alpha^2\nu_{11}}}\right) + \alpha^2\nu_{12}\Phi\left(\frac{\alpha\mu_1}{\sqrt{1+\alpha^2\nu_{11}}}\right)\right] \\
&\qquad\qquad \text{(by Lemma D.3)} \\
&\quad= \nu_{12}\Phi\left(\frac{\mu_1}{\sqrt{\nu_{11}}}\right) && (60)
\end{aligned}
$$

□

# F Derivation of uncertainty propagation formulas for Heaviside activation

In this appendix, we state the $M_\sigma$, $K_\sigma$ and $L_\sigma$ functions (Def. 2.3) for the Heaviside activation function

$$\sigma(x) = \mathbf{1}_{x>0} \tag{61}$$

$$= \lim_{\alpha\to\infty} \left(1/2 + 1/2\hat{\sigma}(\alpha x)\right) \tag{62}$$

where $\hat{\sigma}$ is the odd probit sigmoid defined as in (51). The idea of the proofs is to take dominated limits of the results in Appendix. C, similar to Appendix E's treatment of Appendix D, so we omit them.

**Lemma F.1.** *Let $\sigma$ be the Heaviside function as in* (61)*. Then the functions defined in Def. 2.3 are*

$$M_\sigma(\mu;\nu) = \Phi\left(\frac{\mu}{\sqrt{\nu}}\right),$$

$$K_\sigma(\mu_1,\mu_2;\nu_{11},\nu_{22},\nu_{12}) = \Phi_2\left(\frac{\mu_1}{\sqrt{\nu_{11}}},\frac{\mu_2}{\sqrt{\nu_{22}}};\rho'\right)\Bigg|_{\rho'=0}^{\rho'=\frac{\nu_{12}}{\sqrt{\nu_{11}\nu_{22}}}}, \quad \textit{and}$$

$$L_\sigma(\mu_1;\nu_{11},\nu_{22},\nu_{12}) = \frac{\nu_{12}}{\sqrt{\nu_{11}}}\phi\left(\frac{\mu_1}{\sqrt{\nu_{11}}}\right).$$

## G Derivation of uncertainty propagation formulas for sine activation

In this appendix, we derive the $M_\sigma$, $K_\sigma$ and $L_\sigma$ functions (Def. 2.3) for the sinusoidal activation function

$$\sigma(x) = \sin(x). \tag{63}$$

We begin by recalling the identities that if $Z \sim \mathcal{N}(\mu, \nu)$,

$$\mathbb{E}\sin(Z) = e^{-\nu/2}\sin(\mu) \tag{64}$$

$$\mathbb{E}\cos(Z) = e^{-\nu/2}\cos(\mu) \tag{65}$$

These formulas, which can be derived from the characteristic function of the normal distribution, allow moments of $\sin(Z)$ and $\cos(Z)$ to be computed exactly. There is no need, as in Sitzmann et al. [2020, Lemma 1.6] to use analytic approximations in this step.

From (64) immediately follows:

**Lemma G.1.** *Let $M_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (63)*. Then*

$$M_\sigma(\mu; \nu) = e^{-\nu/2}\sin(\mu).$$

**Lemma G.2.** *Let $K_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (63)*. Then*

$$\begin{aligned} K_\sigma(\mu_1, \mu_2; \nu_{11}, \nu_{22}, \nu_{12}) = &\frac{1}{2}\left[e^{\nu^* + \nu_{12}} - e^{\nu^*}\right]\cos(\mu_1 - \mu_2) \\ &- \frac{1}{2}\left[e^{\nu^* - \nu_{12}} - e^{\nu^*}\right]\cos(\mu_1 + \mu_2), \end{aligned}$$

*where*

$$\nu^* = -\frac{\nu_{11} + \nu_{22}}{2}.$$

*Proof.* Let

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \nu_{11} & \nu_{12} \\ \nu_{12} & \nu_{22} \end{pmatrix}\right). \tag{66}$$

Then by inserting (63), we have

$$\mathrm{Cov}(\sigma(X_1), \sigma(X_2)) = \mathbb{E}\sin(X_1)\sin(X_2) - \mathbb{E}\sin(X_1)\,\mathbb{E}\sin(X_2). \tag{67}$$

Using some trigonometric identities and (65), the first term of (67) becomes

$$\mathbb{E}\sin(X_1)\sin(X_2) = \frac{1}{2}\,\mathbb{E}\cos(\underbrace{X_1 - X_2}_{\mathcal{N}(\mu_1 - \mu_2, \nu_{11} + \nu_{22} - 2\nu_{12})}) - \frac{1}{2}\,\mathbb{E}\cos(\underbrace{X_1 + X_2}_{\mathcal{N}(\mu_1 + \mu_2, \nu_{11} + \nu_{22} + 2\nu_{12})}) \tag{68}$$

$$\begin{aligned} &= \frac{1}{2}\exp\left(-\frac{\nu_{11} + \nu_{22}}{2} + \nu_{12}\right)\cos(\mu_1 - \mu_2) \\ &\quad - \frac{1}{2}\exp\left(-\frac{\nu_{11} + \nu_{22}}{2} - \nu_{12}\right)\cos(\mu_1 + \mu_2) \end{aligned} \tag{69}$$

The second term of (67) becomes

$$\begin{aligned} \mathbb{E}\sin(X_1)\,\mathbb{E}\sin(X_2) = &\frac{1}{2}\exp\left(-\frac{\nu_{11} + \nu_{22}}{2}\right)\cos(\mu_1 - \mu_2) \\ &- \frac{1}{2}\exp\left(-\frac{\nu_{11} + \nu_{22}}{2}\right)\cos(\mu_1 + \mu_2) \end{aligned} \tag{70}$$

The result follows from collecting like terms. □

**Lemma G.3.** *Let $L_\sigma$ be defined as in Def. 2.3, with $\sigma$ as in* (63)*. Then*

$$L_\sigma(\mu_1, \mu_2; \nu_{11}, \nu_{22}, \nu_{12}) = \nu_{12}e^{-\nu_{11}/2}\cos(\mu_1).$$

*Proof.* Let

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \nu_{11} & \nu_{12} \\ \nu_{12} & \nu_{22} \end{pmatrix} \right). \tag{71}$$

Using Lemma B.7, we have

$$\begin{aligned} L_\sigma(\mu_1, \mu_2; \nu_{11}, \nu_{22}, \nu_{12}) &= \nu_{12}\, \mathbb{E}\, \sigma'(X_1) && (72) \\ &= \nu_{12}\, \mathbb{E} \cos(X_1) && (73) \\ &= \nu_{12} e^{-\nu_{11}/2} \cos(\mu_1) && (74) \end{aligned}$$

by (65). □

## H Theoretical Guarantees

This section proves the smooth-distance bounds stated in §4. We first prove the single-layer higher-order expansion, and then prove that the smooth distance satisfies the layerwise recursion.

### H.1 Single-layer smooth-distance expansion

Let $X = \mu_X + \lambda^{-1}\xi$, where $\xi \sim \mathcal{N}(0, \Sigma_X)$, and set $\epsilon = \lambda^{-1}$. Let $Y = f(X)$, $\mu_Y = \mathbb{E}\, Y$, $\Sigma_Y = \operatorname{Cov} Y$, and $\gamma_Y \sim \mathcal{N}(\mu_Y, \Sigma_Y)$. For $h$ satisfying $\|h\|_{\dot{W}^{4,\infty}} \le 1$, Taylor expansion around $\mu_Y$ gives

$$h(y) = \underbrace{h(\mu_Y)}_{\text{(I)}} + \underbrace{Dh(\mu_Y)[y-\mu_Y]}_{\text{(II)}} + \underbrace{\frac{1}{2}D^2h(\mu_Y)[y-\mu_Y, y-\mu_Y]}_{\text{(III)}}$$
$$+ \underbrace{\frac{1}{6}D^3h(\mu_Y)[y-\mu_Y, y-\mu_Y, y-\mu_Y]}_{\text{(IV)}} + \underbrace{R_4(y)}_{\text{(V)}},$$

with $|R_4(y)| \lesssim \|y-\mu_Y\|^4$. Term (I) is deterministic and therefore cancels when subtracting $\mathbb{E}\, h(\gamma_Y)$ from $\mathbb{E}\, h(Y)$. Term (II) cancels because $Y$ and $\gamma_Y$ have the same mean $\mu_Y$. Term (III) cancels because $Y$ and $\gamma_Y$ have the same covariance $\Sigma_Y$. Term (V) is controlled by the fourth central moments:

$$\mathbb{E}\,\|Y-\mu_Y\|^4 + \mathbb{E}\,\|\gamma_Y-\mu_Y\|^4 = O(\epsilon^4),$$

because $Y - \mu_Y = O_p(\epsilon)$ and $\Sigma_Y = O(\epsilon^2)$. It remains to identify the size of term (IV), the cubic term.

Taylor expansion of $f$ around $\mu_X$ gives

$$Y = \underbrace{f(\mu_X)}_{\text{(A)}} + \underbrace{\epsilon J\xi}_{\text{(B)}} + \underbrace{\epsilon^2 Q(\xi)}_{\text{(C)}} + \underbrace{O_p(\epsilon^3)}_{\text{(D)}},$$

where $J = \nabla f(\mu_X)^\intercal$ and $Q(\xi) = \frac{1}{2}\nabla^2 f(\mu_X)[\xi,\xi]$. Taking expectations gives

$$\mu_Y = \underbrace{f(\mu_X)}_{\text{(A)}} + \epsilon^2\, \mathbb{E}\, Q(\xi) + O(\epsilon^4),$$

because the centered Gaussian term (B) has mean zero and the cubic-order term in (D) is odd to leading order. Consequently, subtracting $\mu_Y$ cancels term (A) and centers term (C):

$$Y - \mu_Y = \underbrace{\epsilon J\xi}_{\text{(B)}} + \underbrace{\epsilon^2\left(Q(\xi) - \mathbb{E}\, Q(\xi)\right)}_{\text{(C)}-\mathbb{E}(\text{C})} + \underbrace{O_p(\epsilon^3)}_{\text{(D)}}.$$

The cubic contribution in term (IV) is linear in $\mathbb{E}(Y-\mu_Y)^{\otimes 3}$. Using the previous display,

$$(Y-\mu_Y)^{\otimes 3} = \underbrace{(\text{B})^{\otimes 3}}_{\epsilon^3} + \underbrace{\operatorname{Sym}\Big((\text{B})^{\otimes 2} \otimes \big((\text{C}) - \mathbb{E}(\text{C})\big)\Big)}_{\epsilon^4} + O_p(\epsilon^5),$$

where Sym denotes the sum over the three placements of the $(\text{C}) - \mathbb{E}(\text{C})$ factor. The first term has expectation zero because $(\text{B}) = \epsilon J\xi$ is centered Gaussian. The displayed symmetrized term is therefore the first possible nonzero contribution: it contains two factors of (B), each of size $O(\epsilon)$, and one factor of $(\text{C}) - \mathbb{E}(\text{C})$, of size $O(\epsilon^2)$. All remaining terms contain either one factor of $(\text{D}) = O_p(\epsilon^3)$ or at least two factors of $(\text{C}) - \mathbb{E}(\text{C})$, and hence are $O_p(\epsilon^5)$ or smaller. Thus

$$\mathbb{E}(Y-\mu_Y)^{\otimes 3} = O(\epsilon^4).$$

Since $\gamma_Y$ is Gaussian, its third central moment is zero. Thus term (IV) contributes only $O(\epsilon^4)$ to $\mathbb{E}\, h(Y) - \mathbb{E}\, h(\gamma_Y)$. Therefore, uniformly over $\|h\|_{\dot{W}^{4,\infty}} \le 1$,

$$\left|\mathbb{E}\, h(Y) - \mathbb{E}\, h(\gamma_Y)\right| = O(\epsilon^4) = O(\lambda^{-4}).$$

Taking the supremum over the homogeneous unit ball of $\dot{W}^{4,\infty}$ proves Theorem 4.2.

The same expansion also explains the comparison with linearization. The linearized Gaussian $G_{\mathrm{lin}}$ has mean $f(\mu_X)$, whereas

$$\mu_Y = f(\mu_X) + \epsilon^2 \, \mathbb{E}\, Q(\xi) + O(\epsilon^4).$$

For test functions with gradient aligned with $\mathbb{E}\, Q(\xi)$, this missing second-order mean correction produces a generic $O(\epsilon^2) = O(\lambda^{-2})$ smooth-distance error. Standard fixed sigma-point approximations that do not exactly match the true layer mean and covariance have the same obstruction in this expansion.

### H.2 Layerwise recursion in smooth distance

We recall the layer-by-layer Normal approximation resulting in $Y_{\mathrm{ana}}$ (henceforth, $Y$) alongside the exact neural network formula resulting in $Y_{\mathrm{true}}$:

$$Y = Y^{\ell} \qquad\qquad Y_{\mathrm{true}} = Y^{\ell}_{\mathrm{true}} \tag{75a}$$

$$Y^k = \mathrm{N}\, g^k(Y^{k-1}), \qquad\qquad Y^k_{\mathrm{true}} = g^k(Y^{k-1}_{\mathrm{true}}), \qquad k \in \{1 \dots \ell\}, \tag{75b}$$

$$Y^0 = X \qquad\qquad Y^0_{\mathrm{true}} = X, \tag{75c}$$

where $g^k$ is the function

$$g^k(x) = g(x; A^k, b^k, C^k, d^k). \tag{75d}$$

The smooth distance is an integral probability metric, so for any random variables $U, V, W$,

$$d_4(U, W) \le d_4(U, V) + d_4(V, W).$$

Moreover, for any $C^4$ map $g$,

$$\begin{aligned} d_4(g(U), g(V)) &= \sup_{\|h\|_{\dot{W}^{4,\infty}} \le 1} \left| \mathbb{E}\, h(g(U)) - \mathbb{E}\, h(g(V)) \right| \\ &\le C_4(g) d_4(U, V), \end{aligned}$$

where

$$C_4(g) = \sup_{\|h\|_{\dot{W}^{4,\infty}} \le 1} \|h \circ g\|_{\dot{W}^{4,\infty}}.$$

For a layer $g(x; A, b, C, d) = \sigma(Ax + b) + Cx + d$, the Faà di Bruno formula gives

$$C_4(g) \le P_4\left( \|A\|, \|C\|, \|\sigma'\|_\infty, \dots, \|\sigma^{(4)}\|_\infty \right),$$

for a polynomial $P_4$ depending only on the input and output dimensions.

Define

$$\Delta_4^k := d_4(Y^k_{\mathrm{true}}, Y^k), \qquad\qquad E_{4,k} := d_4(g^k(Y^{k-1}), Y^k). \tag{76}$$

Then the basic triangle inequality is

$$\begin{aligned} \Delta_4^k &= d_4(g^k(Y^{k-1}_{\mathrm{true}}), Y^k) && \text{(definition of hidden layer (75b))} \\ &\le d_4(g^k(Y^{k-1}_{\mathrm{true}}), g^k(Y^{k-1})) + d_4(g^k(Y^{k-1}), Y^k) && \text{(triangle inequality)} \\ &\le C_4(g^k) d_4(Y^{k-1}_{\mathrm{true}}, Y^{k-1}) + d_4(g^k(Y^{k-1}), Y^k) && \text{(composition property of } d_4\text{)} \\ &\le C_4(g^k)\Delta_4^{k-1} + E_{4,k}. && (77) \end{aligned}$$

If $X$ is Gaussian, then $\Delta_4^0 = 0$. The forcing term $E_{4,k} = d_4(g^k(Y^{k-1}), Y^k)$ is exactly the single-layer Gaussianization error for the Normal input $Y^{k-1}$. In the low-variance regime, Theorem 4.2 gives $E_{4,k} = O(\lambda^{-4})$. Therefore

$$\Delta_4^k \le C_4(g^k)\Delta_4^{k-1} + E_{4,k}. \tag{78}$$

Unrolling this recursion and using $\Delta_4^0 = 0$ yields

$$\Delta_4^{\ell} \le \sum_{k=1}^{\ell} \left( \prod_{r=k+1}^{\ell} C_4(g^r) \right) E_{4,k},$$

which proves Theorem 4.3.

# I Supplement to Fig. 4.1

The convergence experiment in Fig. 4.1 is generated by `demo/deep_sine_convergence.py`. The purpose is to visualize the small-input-covariance regime in a fixed nonlinear network where both exact Gaussian moment matching and linearization can be compared against a quasi-Monte Carlo pseudo-truth.

The network is a randomly initialized deep sine network generated by the same `build_network` utility used in the random-network experiments. Specifically, the script fixes the JAX random seed to 1, uses input dimension 3, chooses the `deep` architecture, and uses the sine activation. For a covariance scale $c > 0$, the input distribution is

$$X_c \sim \mathcal{N}(0, cI_3).$$

In the full experiment, the covariance scales are logarithmically spaced as

$$c \in \mathrm{geomspace}(10^{-1}, 10^{-8}, 8).$$

The script also includes a faster diagnostic mode using 5 covariance scales from $10^{-1}$ to $10^{-5}$.

For each covariance scale, one quasi-Monte Carlo run is performed. The full experiment uses $2^{20}$ samples; the fast diagnostic mode uses $2^{12}$ samples. These samples are propagated through the deterministic neural network to produce samples from $Y_{\mathrm{true}} = f(X_c)$. The pseudo-true Gaussian is then

$$Y_{\mathrm{pseudo}} = \mathcal{N}\left(\widehat{\mathbb{E}}Y_{\mathrm{true}}, \widehat{\mathrm{Var}}Y_{\mathrm{true}}\right),$$

where the mean and variance are estimated from the quasi-Monte Carlo output samples.

The two approximations compared are

- $Y_{\mathrm{ana}}$, produced by analytic layerwise Gaussian moment matching, and
- $Y_{\mathrm{lin}}$, produced by linearized covariance propagation.

For each method, the reported error is

$$D_{\mathrm{KL}}(Y_{\mathrm{pseudo}} \parallel Y_{\mathrm{method}}).$$

The raw results, sorted summary table, and figure are written to `docs/manuscript/generated/convergence/`.

## J Supplement to §5.2

We expand on each of the terms in the evidence lower bound (2), repeated below.

$$\theta^* = \arg\min_\theta \left\{ -\mathbb{E}_{w\sim q(w;\theta)} \log p(y \mid w, x) + D_{\mathrm{KL},w}(q(w;\theta) \mid p(w)) \right\}.$$

Our neural network $f(x; w)$ consists of a single hidden layer. The weights $w$ encompass:

- $A$, $b$: pre-activation weights,
- $C$, $d$: post-activation weights, as well as
- $\lambda$: (homoscedastic) output log precision.

The log likelihood of $w$ is Normal:

$$\log p(y \mid w, x) = \frac{1}{2}\left(-\lambda + e^\lambda \left[y - f(x; w)\right]^2\right) + \frac{1}{2}\log\left[2\pi\right].$$

The prior distribution $p(w)$ is the Kaiming initialization of independent Normal distributions, zero mean in each matrix entry and variance equal to $2/\text{fan-in}$. Like Wu et al. [2019], Petersen et al. [2024], Wright et al. [2024], the variational distribution $q(w;\theta)$ is a Gaussian matrix with independent entries, parameterized by means and log precisions.

We train using the Adam optimizer with a learning rate of 0.1 for 10 000 iterations. Each gradient estimate averages 10 stochastic ELBO estimates, each drawing one random sample from the variational distribution $q(w;\theta)$ per training instance.

The implementation uses Equinox and JAX [Kidger and Garcia, 2021, DeepMind et al., 2020]. Each network takes about 2 minutes on an Nvidia T1200 GPU.

In reporting, we draw one million random variates from the variational posterior distribution for up to 1000 instances from the test set.

The datasets are

- Servo [Ulrich, 1986] (CC BY 4.0)
- Liver disorders [Unknown, 2016]
- Auto MPG [R. Quinlan, 1993] (CC BY 4.0)
- Real estate valuation [I-Cheng Yeh, 2018] (CC BY 4.0)
- Forest fires [Paulo Cortez, 2007] (CC BY 4.0)
- Infrared thermography temperature Wang et al. (CC0)
- Combined cycle power plant [Pnar Tfekci, 2014] (CC BY 4.0)
- Concrete compressive strength [I-Cheng Yeh, 1998] (CC BY 4.0)
- Airfoil self-noise [Thomas Brooks, 1989] (CC BY 4.0)
- Communities and crime [Redmond, 2002] (CC BY 4.0)
- Parkinsons telemonitoring [Athanasios Tsanas, 2009] (CC BY 4.0)

**Impact/ethics statement** Some of these datasets, served on the UC Irvine Machine Learning Repository, are based on observing humans. This choice of datasets follows Rui Li and Trapp [2025]. While we defer to UCI's custodianship and ICLR 2025's review process in deeming them ethically appropriate for methodology research, we are open to evidence to the contrary.

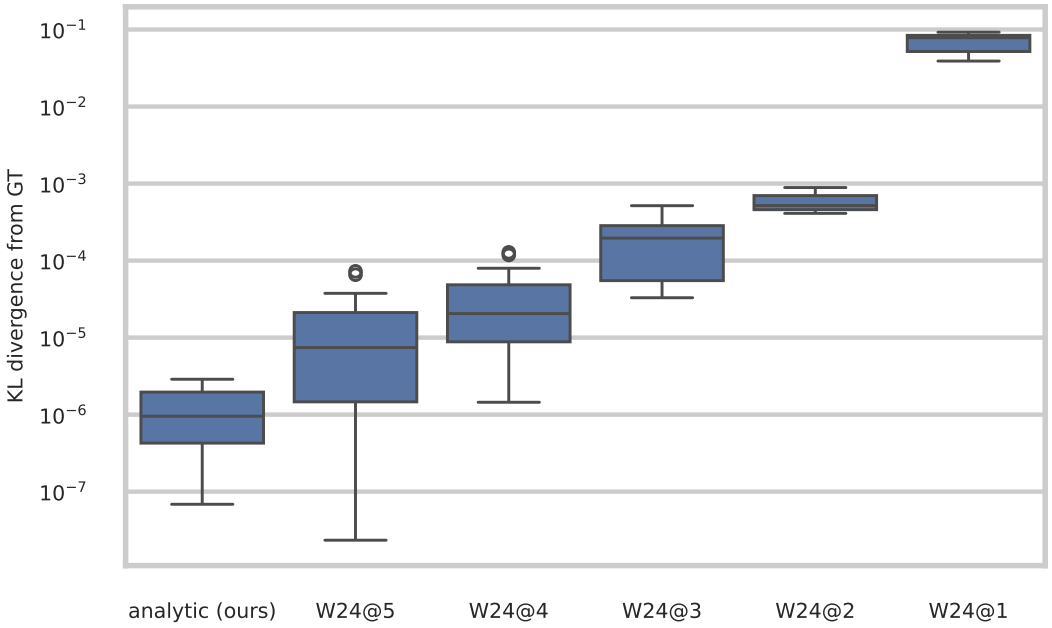



Figure J.1: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the Servo dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

### J.1 Results

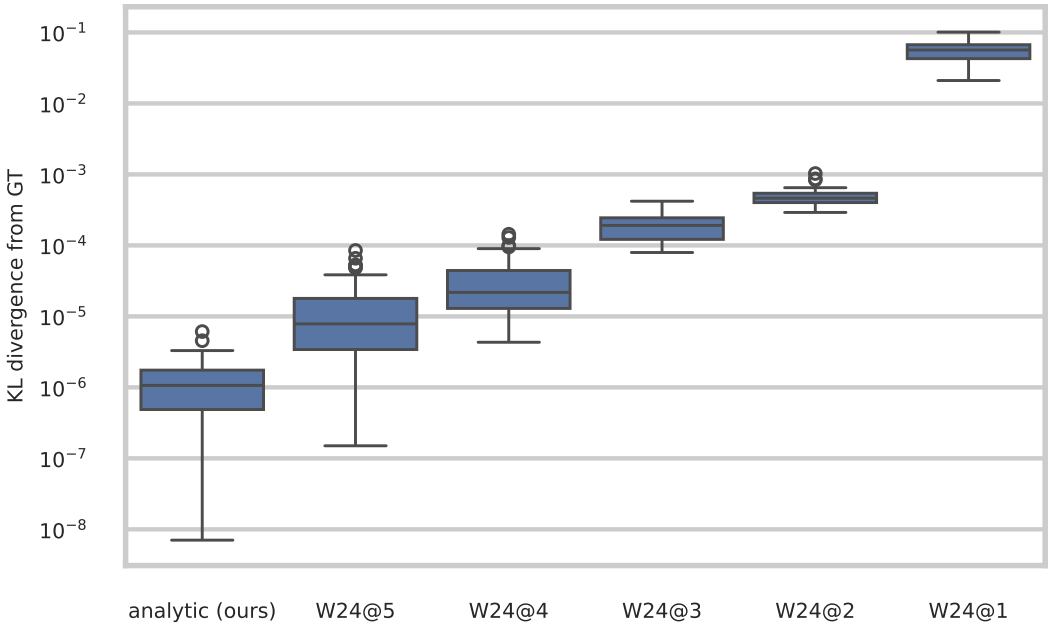



Figure J.2: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the liver disorders dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

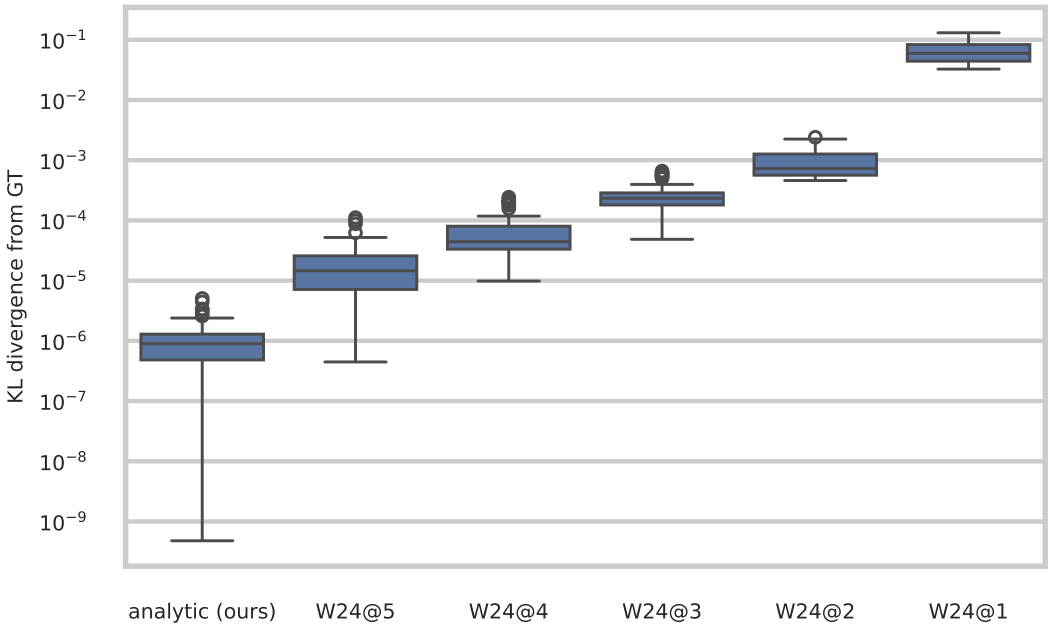



Figure J.3: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the Auto MPG dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

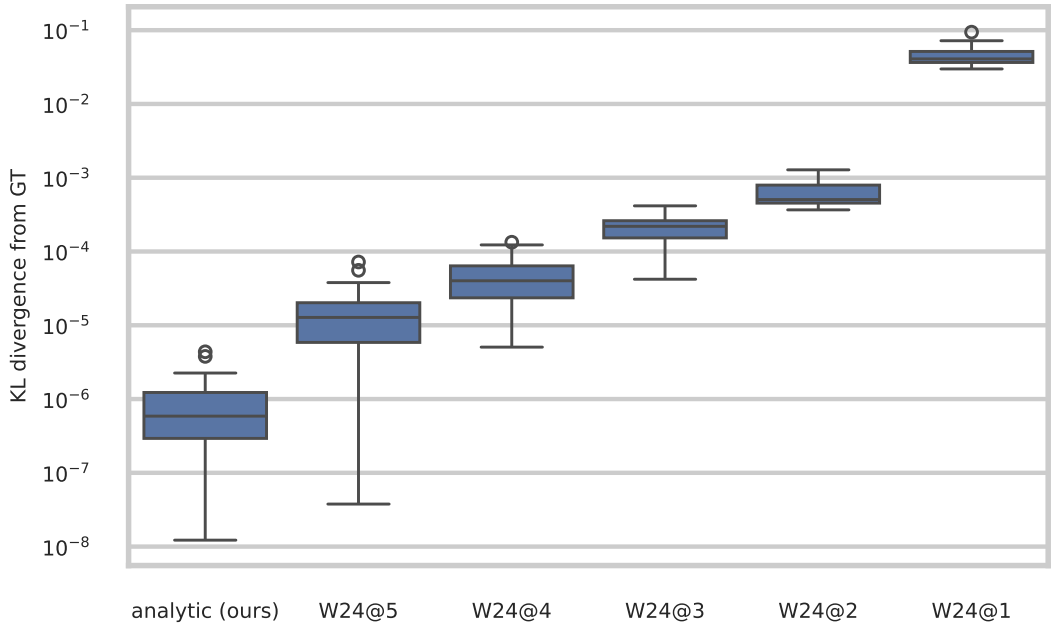



Figure J.4: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the real estate valuation dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

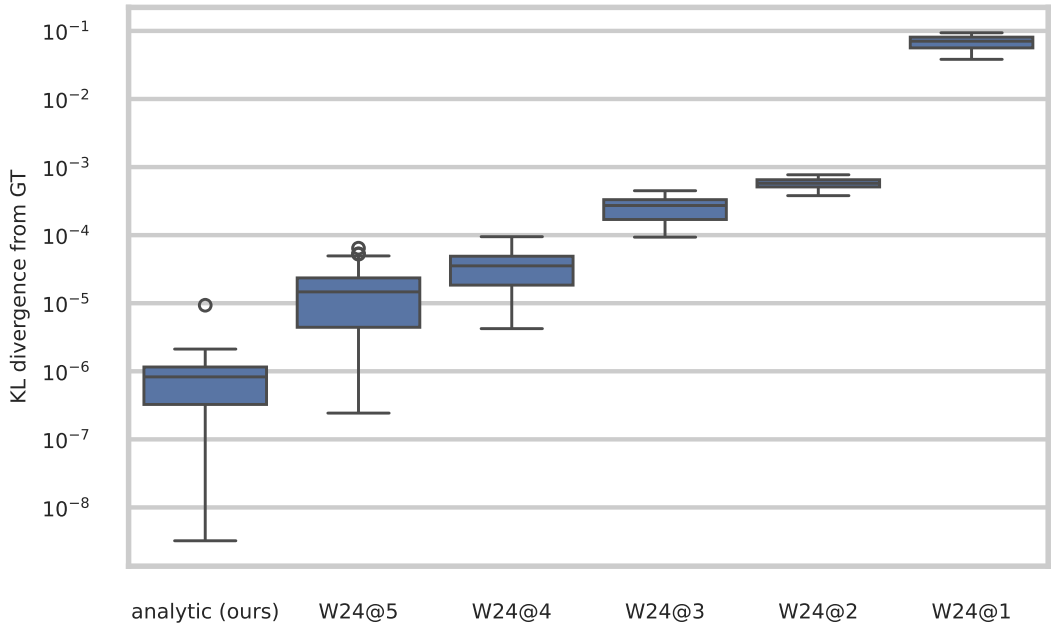



Figure J.5: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the forest fires dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

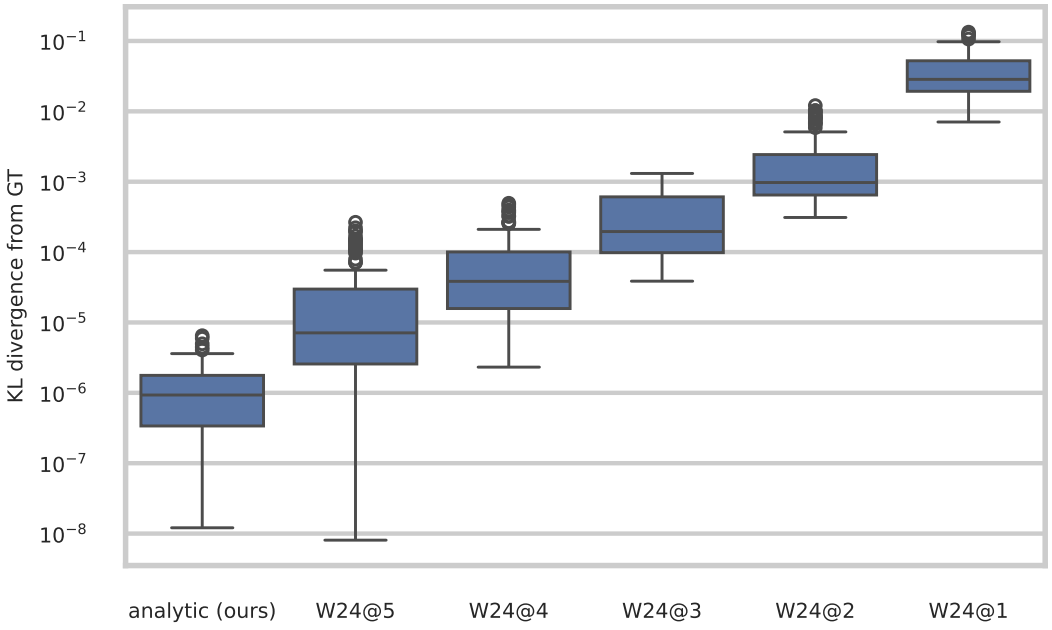



Figure J.6: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the infrared thermography temperature dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

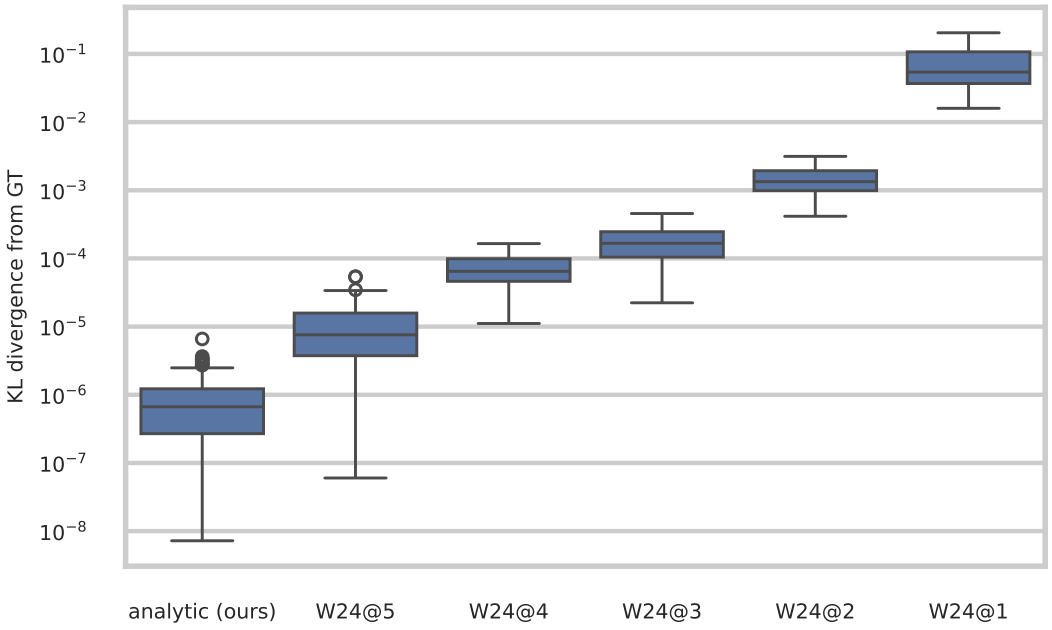



Figure J.7: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the concrete compressive strength dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

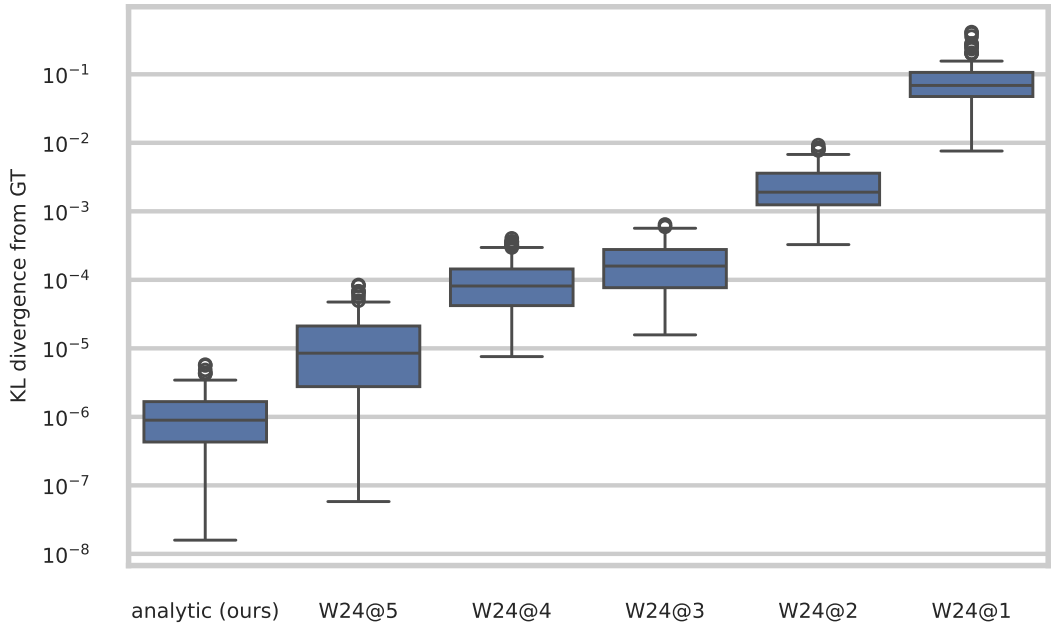



Figure J.8: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the airfoil self-noise dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

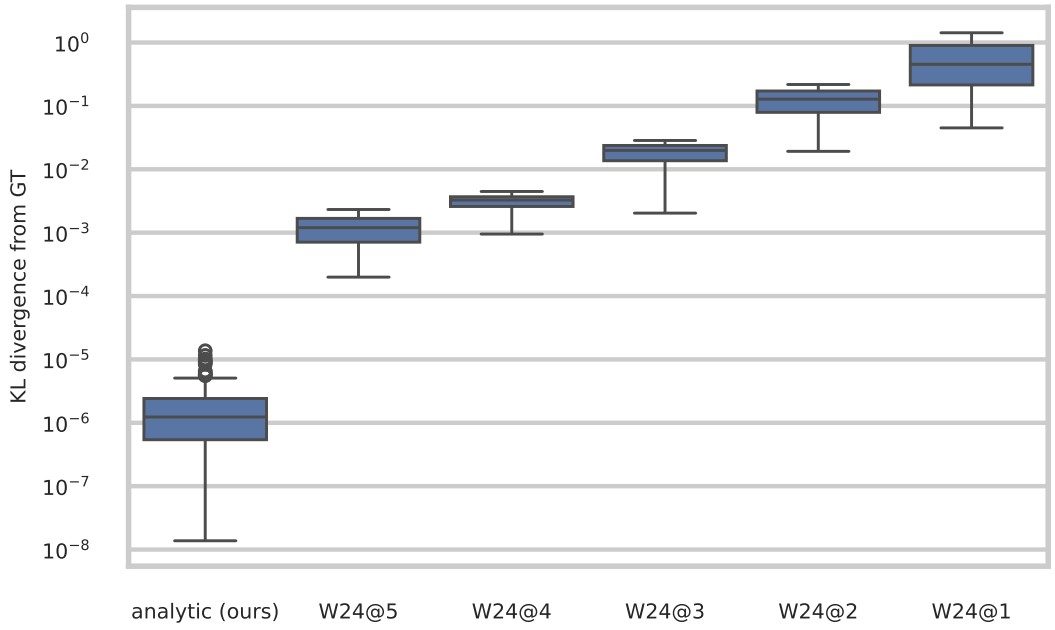



Figure J.9: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the communities and crime dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

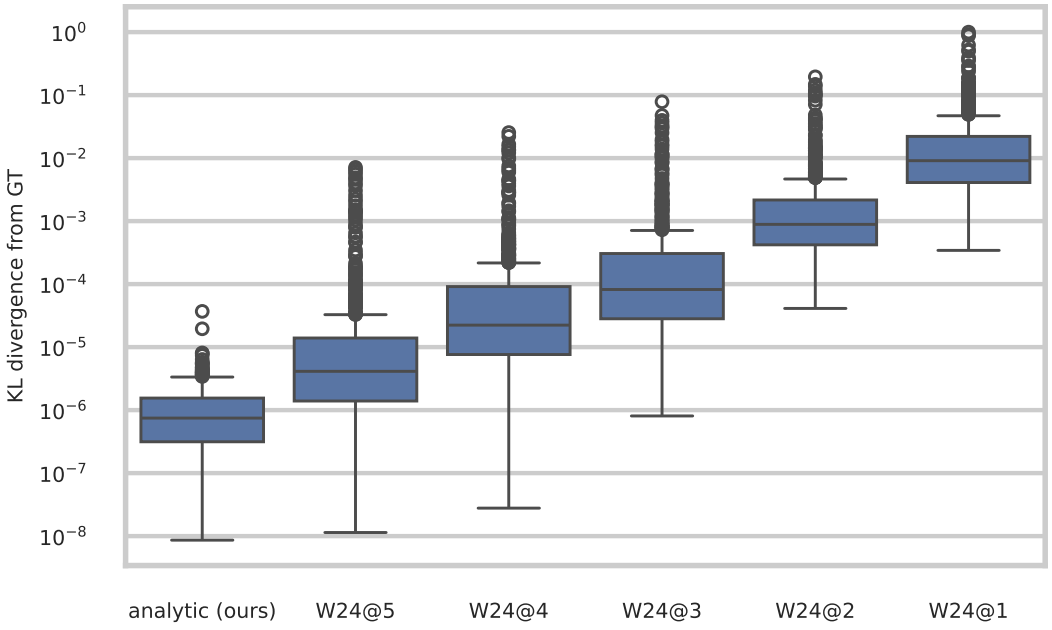



Figure J.10: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the Parkinsons telemonitoring dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

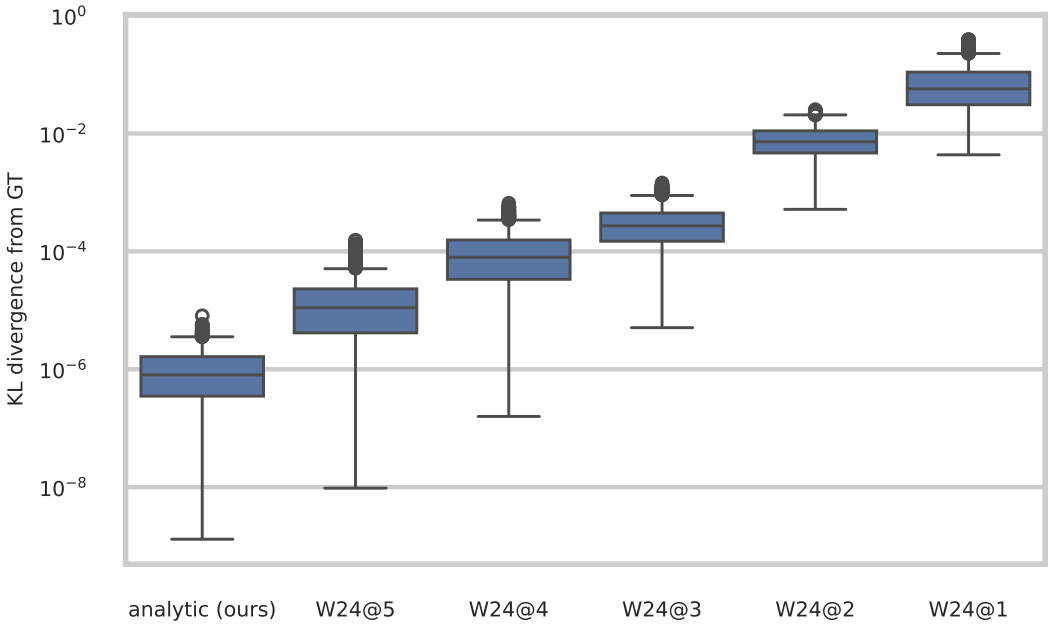



Figure J.11: KL divergence between ground truth predictive distribution (by Monte Carlo) and approximations for the combined cycle power plant dataset. W24@$k$ means Wright et al. [2024] with $k$ terms in the series expansion.

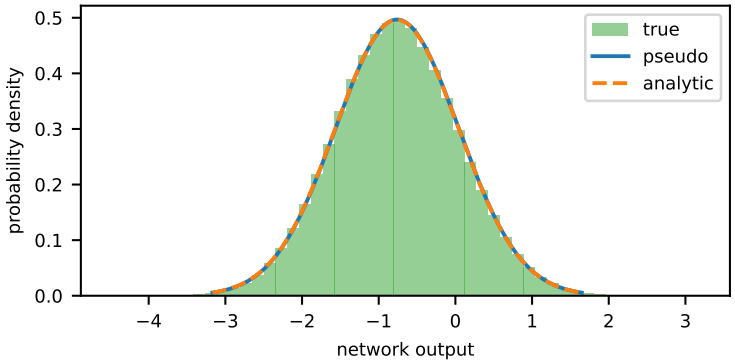



Figure K.1: Output distribution of a stochastic neural network, pseudo-true Normal distribution ("pseudo"), and layer-by-layer moment-matched Normal distribution ("analytic").

## K Supplement to §5.3

The random variable $\mathbf{1}_{U<\Phi(x)}$ has a Bernoulli distribution with parameter $\Phi(x)$, which contributes an additional diagonal variance term to Lemma 2.4 as we see from applying the Law of Total Covariance:

$$\begin{aligned}\left(\operatorname{Cov} g_{\tilde{\sigma}}(X;A,b,C,d)\right)_{i,j} &= \left(\operatorname{Cov}\mathbb{E}\left[g_{\tilde{\sigma}}(X;A,b,C,d)\mid X\right]\right)_{i,j} \\ &\quad + \left(\mathbb{E}\operatorname{Cov}\left[g_{\tilde{\sigma}}(X;A,b,C,d)\mid X\right]\right)_{i,j} \qquad (79)\end{aligned}$$

$$\left(\operatorname{Cov}\mathbb{E}\left[g_{\tilde{\sigma}}(X;A,b,C,d)\mid X\right]\right)_{i,j} = \left(\operatorname{Cov} g_{\sigma}(X;A,b,C,d)\right)_{i,j} \qquad (80)$$

since $\mathbb{E}\left(\tilde{\sigma}(X,U)\mid X\right) = \sigma(X)$ and

$$\begin{aligned}\left(\mathbb{E}\operatorname{Cov}\left[g_{\tilde{\sigma}}(X;A,b,C,d)\mid X\right]\right)_{i,j} &= \delta_{ij}\,\mathbb{E}\operatorname{Var}\left[\left(g_{\tilde{\sigma}}(X;A,b,C,d)\right)_i \mid X\right] \qquad (81)\\ &= 4\delta_{ij}\,\mathbb{E}\,\Phi(\xi_i)\left(1-\Phi(\xi_i)\right) \qquad (82)\end{aligned}$$

where $\xi_i \sim \mathcal{N}(\mu_i, \nu_{ii})$. This can be expressed using Owen's T function,

$$\mathbb{E}\,\Phi(\xi_i)\left(1-\Phi(\xi_i)\right) = 2T\left(\frac{\mu_i}{\sqrt{1+\nu_{ii}}}, \frac{1}{\sqrt{1+2\nu_{ii}}}\right), \qquad (83)$$

which satisfies [Owen, 1980, equation 20,010.4]

$$\int \Phi(\mu+\sqrt{\nu}z)^2\phi(z)dz = \Phi\left(\frac{\mu}{\sqrt{1+\nu}}\right) - 2T\left(\frac{\mu}{\sqrt{1+\nu}}, \frac{1}{\sqrt{1+2\nu}}\right).$$

## L Supplement to §5.1

We apply our method and other benchmarks to the random-network ensembles generated by `demo/test_case.py`. The script iterates over 36 ensembles:

- network architecture $\in \{\text{wide}, \text{deep}\}$
- weights $\in \{\text{initialized}, \text{trained}\}$
- activation function $\in \{$probit, probit residual, sine, sine residual, GeLU, GeLU residual, ReLU, ReLU residual, Heaviside, Heaviside residual$\}$.

(The trained Heaviside and Heaviside-residual ensembles are skipped.) From each ensemble, we sample one neural network and evaluate the goodness of approximation of the output distribution for input distributions:

- variance $\in \{\text{small}, \text{medium}, \text{large}\}$.

In each case we compare the distributions of:

- $Y_{\text{true}}$, the true distribution
- $Y_{\text{pseudo}}$, the pseudo-true Gaussian distribution
- $Y_{\text{ana}}$, the analytic layer-wise Gaussian approximation (our method)
- $Y_{\text{mfa}}$, mean-field: applying our method for moment propagation and setting off-diagonal layer covariances to zero
- $Y_{\text{lin}}$, linearization-based moment propagation
- $Y_{\text{u}'95}$, unscented transform of the whole network using $\alpha = 1, \beta = 0, \kappa = 0$
- $Y_{\text{u}'02}$, unscented transform of the whole network using $\alpha = 0.001, \beta = 2, \kappa = 0$

**Ensembles of neural networks**

This section specifies the 36 ensembles of random neural networks and the three ensembles of inputs used to produce the 108 test cases in the accompanying *Supplementary Numerical Results*. The neural networks in this example are parameterized as in Def. 2.1. Let $d_{\text{hidden}}$ and $w_{\text{hidden}}$ be the depth and width of the hidden layers, respectively. A random neural network $f : \mathbb{R}^3 \to \mathbb{R}$ is specified with $d_{\text{hidden}} + 1$ layers. The output layer is linear, with weights $C^{d_{\text{hidden}}+1}$ and biases $d^{d_{\text{hidden}}+1}$. The first hidden layer has $C^1 = 0_{w_{\text{hidden}} \times 3}$, $d^1 = 0_{w_{\text{hidden}}}$. The four degrees of freedom in the test cases are:

**architecture**
- if architecture = wide, then $d_{\text{hidden}} = 5$, $w_{\text{hidden}} = 400$;
- if architecture = deep, then $d_{\text{hidden}} = 20$, $w_{\text{hidden}} = 100$.

**weights**
- if weights = initialized:
  - $A$ matrices are initialized with i.i.d. Gaussian entries having mean zero and variance equal to 2 divided by the number of columns
  - if activation $\in \{$probit, probit residual$\}$, then $b$ vectors are initialized with independently sampled entries from $\mathcal{N}(0, 1)$
  - if activation $\in \{$sine, sine residual$\}$, then $b$ vectors are initialized with independently sampled entries from $\mathcal{U}(-\pi, \pi)$
  - if activation $\in \{$GeLU, GeLU residual, ReLU, ReLU residual, Heaviside, Heaviside residual$\}$, then $b$ vectors are initialized with independently sampled entries from $\mathcal{N}(0, 1)$
  - if the activation is non-residual, then hidden-layer $C$ matrices are initialized to the zero matrix.
  - if the activation is residual, then square hidden-layer $C$ matrices after the first hidden layer are initialized to the identity matrix, and all other hidden-layer $C$ matrices are initialized to the zero matrix.
  - the output-layer $C$ matrix is initialized with i.i.d. Gaussian entries having mean zero and variance equal to the reciprocal of the number of columns.
  - $d$ vectors are initialized to the zero vector.

- if weights = trained and activation $\notin \{\text{Heaviside}, \text{Heaviside residual}\}$, after initialization as above, the neural network is trained to minimize the mean squared error loss on a pseudorandomly generated dataset of ten $(x, y)$ samples drawn from $\mathcal{N}(0, 1)$. Training consists of using the AdamW optimizer [Loshchilov and Hutter, 2019] with a learning rate of $10^{-6}$ for 30,000 iterations and until the loss is less than $10^{-8}$ (whichever is later). Implementation is due to Optax, DeepMind et al. [2020].

**activation function**

- if activation $\in \{\text{probit}, \text{probit residual}\}$, then $\sigma(x) = 2\Phi(x) - 1$ where $\Phi$ is the cumulative distribution function of the standard normal distribution
- if activation $\in \{\text{sine}, \text{sine residual}\}$, then $\sigma(x) = \sin(x)$
- if activation $\in \{\text{GeLU}, \text{GeLU residual}\}$, then $\sigma(x) = x\Phi(x)$.
- if activation $\in \{\text{ReLU}, \text{ReLU residual}\}$, then $\sigma(x) = \max(0, x)$.
- if activation $\in \{\text{Heaviside}, \text{Heaviside residual}\}$, then $\sigma(x) = \mathbf{1}_{\{x \geq 0\}}$

**variance**

- if variance = small, then $X \sim \mathcal{N}(0, 10^{-2} I)$
- if variance = medium, then $X \sim \mathcal{N}(0, I)$
- if variance = large, then $X \sim \mathcal{N}(0, 10^2 I)$

**Simulation and reporting**

The only stochastic uncertainty in these numerical results comes from estimating $Y_{\text{true}} = f(X)$ and $Y_{\text{pseudo}}$. For each test case, `demo/test_case.py` uses twenty independently scrambled quasi-Monte Carlo realizations of $N = 2^{16}$ samples [Virtanen et al., 2020]. The pseudo-true Gaussian $Y_{\text{pseudo}}$ is recomputed from each realization using the sample mean and covariance. Tables report statistical uncertainty in the form mean $\pm$ standard error across the twenty realizations. Distribution plots pool the twenty realizations and display the resulting $Y_{\text{true}}$ sample with a histogram.

The reported Wasserstein statistic between a Normal distribution and $Y_{\text{true}}$ is computed by

$$d_{\text{W}}(\mathcal{N}(\mu, \sigma^2), Y_{\text{true}}) \approx s_Y^{-1/2} \frac{1}{N} \sum_{i=1}^{N} \left| y_{(i)} - Q\left(\frac{i - 1/2}{N}\right) \right|$$

where $\{y_{(i)}\}_{i=1}^{N}$ are the quasi-Monte Carlo samples of $Y_{\text{true}}$ sorted in ascending order, $Q$ is the quantile function of $\mathcal{N}(\mu, \sigma^2)$, and $s_Y$ is the sample standard deviation of the same realization. When plotting distributions, we show $Y_{\text{true}}$ using a histogram with 50 bins and overlay the pseudo-true, analytic, mean-field, linear, and two unscented Gaussian densities.

The horizontal axes are scaled to include the 0.5th (99.5th) percentiles of the quasi-Monte Carlo samples of $Y_{\text{true}}$, or the mean minus (plus) three standard deviations of $Y_{\text{pseudo}}$, whichever is smaller (greater).

The entire suite of 108 test cases is laptop-scale; including initialization, training, quasi-Monte Carlo, and reporting, a representative run takes 1–2 hours on an Ubuntu system with an 11th Gen Intelő Core i7-11850H CPU and 32GB of RAM.

### L.1 Summaries

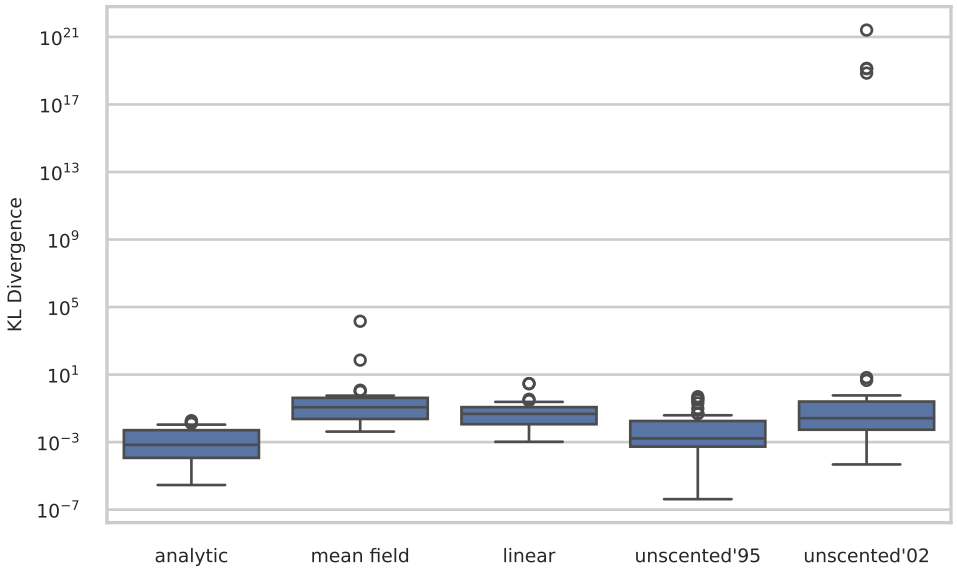



Figure L.1: Comparison of goodness of approximation (lower KL divergence is better) for all random neural networks, grouped by approximation method, in the small input variance scenario.

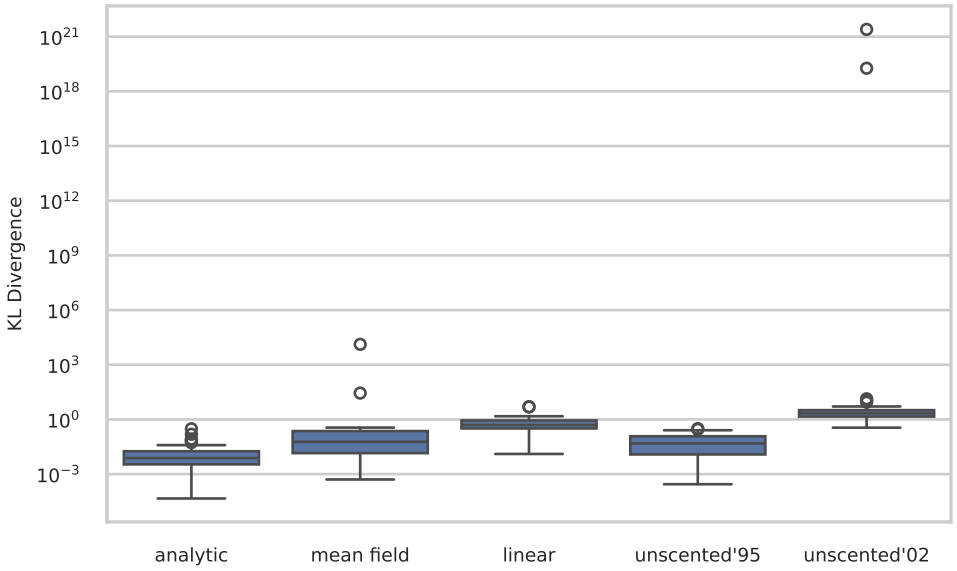



Figure L.2: Comparison of goodness of approximation (lower KL divergence is better) for all random neural networks, grouped by approximation method, in the medium input variance scenario.

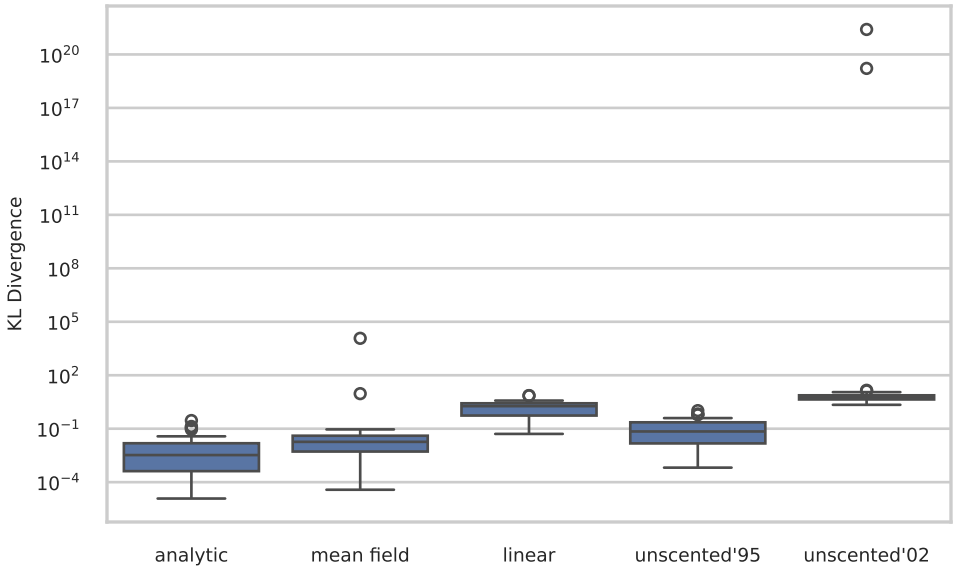



Figure L.3: Comparison of goodness of approximation (lower KL divergence is better) for all random neural networks, grouped by approximation method, in the large input variance scenario.

## M Comments on computational complexity

The cost of moment matching, like that of linearized covariance propagation, is cubic in the number of hidden neurons due to the need to evaluate "sandwich" covariances such as $A\Sigma A^\intercal$. [2] By contrast, the computational complexity of evaluating a neural network is quadratic in the number of hidden neurons. Thus, theoretically, moment matching is one polynomial order of magnitude slower than point prediction, on the same order as linearization. For neural networks with high input dimension, moment matching is much faster than Monte Carlo, which scales exponentially with dimension.

We performed a simple experiment using Jupyter Notebook's `%%timeit` to compare the computational cost of moment matching and Monte Carlo. We compare our method with the ground truth method, which runs quasi-Monte Carlo (QMC) on $2^{16}$ samples. We use GeLU because it has the most complicated covariance expression (see Appendix D).

On "wide residual" networks (defined in Appendix L):

- Our method takes 88.4 ms $\pm$ 5.96 ms.
- QMC takes 1.79 s $\pm$ 83 ms.

On "deep residual" networks (defined in Appendix L):

- Our method takes 23.7 ms $\pm$ 957 $\mu$s.
- QMC takes 878 ms $\pm$ 61.8 ms.

[2] Contrary to occasional claims that the cost of propagating a covariance matrix scales *quadratically* with the number of hidden neurons [Akgül et al., 2025, §5].

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The theoretical convergence claim in particular is supported by an experiment.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The main limitations of this work are that 1) it is theoretical rather than applied, and 2) it leaves softmax-type activations as an open problem.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Proofs are in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We give detailed instructions as well as Python code.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide code in the supplement.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: This is specified in the methodology appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Where applicable, we report uncertainty over Monte Carlo or quasi-Monte Carlo repetitions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments are laptop-scale, and compute resources are described in the appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research does not involve human-subjects research or new data collection.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We believe that the technical contribution of this paper does not have immediate consequences for society.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: We don't use pre-trained language models, image generators, or scraped datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: This information is stated in the methodological appendix where the asset is introduced.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Assets include a large collection of experimental results. We explain how they are generated and provide code to reproduce them.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.