Wide Bayesian neural networks have a simple weight posterior: theory and accelerated sampling

Jiri Hron [1 2] **Roman Novak** [1] **Jeffrey Pennington** [1] **Jascha Sohl-Dickstein** [1]

Abstract

We introduce *repriorisation*, a data-dependent reparameterisation which transforms a Bayesian neural network (BNN) posterior to a distribution whose KL divergence to the BNN prior vanishes as layer widths grow. The repriorisation map acts directly on parameters, and its analytic simplicity complements the known neural network Gaussian process (NNGP) behaviour of wide BNNs in function space. Exploiting the repriorisation, we develop a Markov chain Monte Carlo (MCMC) posterior sampling algorithm which mixes *faster* the wider the BNN. This contrasts with the typically poor performance of MCMC in high dimensions. We observe up to 50x higher effective sample size relative to no reparametrisation for both fully-connected and residual networks. Improvements are achieved at all widths, with the margin between reparametrised and standard BNNs growing with layer width.

1. Introduction

Bayesian neural networks (BNNs) have achieved significantly less success than their deterministic NN counterparts. Despite recent progress (e.g., Khan et al., 2018; Maddox et al., 2019; Osawa et al., 2019; Dusenberry et al., 2020; Daxberger et al., 2021; Izmailov et al., 2021), (a) our theoretical understanding of BNNs remains limited, and (b) practical applicability is hindered by high computational demands. We make progress on both these fronts.

Our work revolves around a reparametrisation $\theta = T(\phi)$ of the (flattened) NN weights $\theta \in \mathbb{R}^d$. Its form comes from a theorem in which we establish that the Kullback–Leibler (KL) divergence between the standard normal distribution $\mathcal{N}(0, I_d)$ and the reparametrised *posterior* $p(\phi \mid \mathcal{D}) = p(T(\phi) \mid \mathcal{D}) \left|\det \partial_\phi T(\phi)\right|$ converges to zero as the BNN layers grow wide.[1] This closes a gap in the theory of over-parametrised NNs by providing a rigorous characterisation of the BNN *parameter space* behaviour (Figure 1 and Section 2), and enables faster posterior sampling (Section 3).

In *function space*, wide BNN *priors* converge (weakly) to a so-called neural network Gaussian process (NNGP) limit (Matthews et al., 2018; Lee et al., 2018; Garriga-Alonso et al., 2019; Novak et al., 2019; Yang, 2019; Hron et al., 2020b), and the *posteriors* converge (weakly) to that of the corresponding NNGP (assuming the likelihood is a bounded continuous function of the NN outputs; Hron et al., 2020a). *Parameter space* behaviour is less understood (Hron et al., 2020a). The main exception is the work of Matthews et al. (2017) who showed that randomly initialising a NN, and then optimising only the last layer with respect to a mean squared error loss, is equivalent to drawing a sample from the *conditional last layer posterior*. This result holds for a Gaussian likelihood and zero observation noise. Our reparametrisation T can be seen as a non-zero noise generalisation of the underlying map from the initial to the optimised weights, and provides samples from the *joint posterior* over all layers in the infinite width limit (Theorem 2.1).

Since sampling from the 'KL-limit' $\mathcal{N}(0, I)$ is trivial relative to the notoriously complex BNN posterior, our theory motivates applying Markov chain Monte Carlo (MCMC) to the reparametrised density. To apply MCMC, two concerns must be addressed: (i) understanding if the KL-closeness to $\mathcal{N}(0, I_d)$ actually simplifies sampling, and (ii) computational efficiency. In Section 3.1, we partially answer (i) by proving that the gradient $\nabla_\phi \log p(\phi \mid \mathcal{D})$ concentrates around the log density gradient of $\mathcal{N}(0, I)$ in wide BNNs. For (ii), we propose a simple-to-implement and efficient way of computing both $T(\phi)$ and the corresponding density at the same time using Cholesky decomposition (Section 3.2).

Applying Langevin Monte Carlo (LMC), we empirically demonstrate up to 50x improved mixing speed, as measured by effective sample size (ESS). The phenomenon occurs both for residual and fully-connected networks (FCNs), and

[1]Google Research, USA [2]University of Cambridge, UK. Correspondence to: Jiri Hron <jh2084@cam.ac.uk>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]Case distinguishes distributions (P) and their densities (p). Densities are used in place of distributions where convenient.

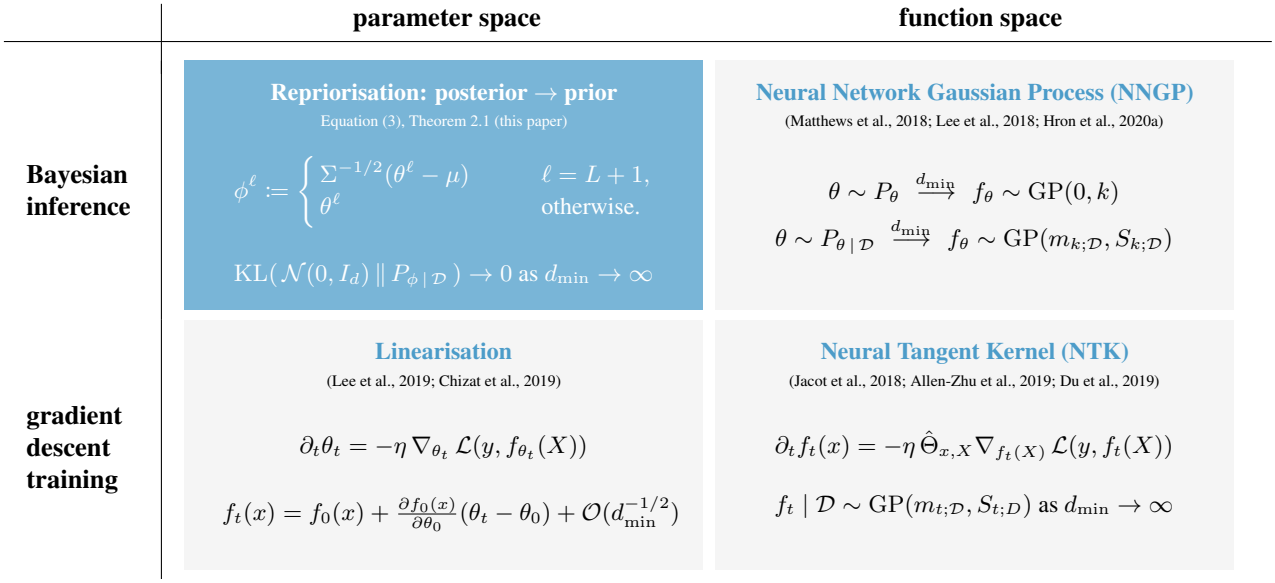

Figure 1. **As neural networks are made wide, their behaviour often becomes simple.** Past results examine wide NNs in either function space or parameter space, and either under gradient descent training or Bayesian inference. The central result for each condition is stated, using this paper's formalism where applicable. The behaviour of wide Bayesian NNs in parameter space is largely unexplored. We address this gap, by reparametrising the weight posterior so that its KL divergence from the prior $\mathcal{N}(0, I_d)$ vanishes with increasing width.

for a variety of dataset sizes and hyperparameter configurations. The improvement over no reparametrisation increases with layer width, but is observed even far from the NNGP regime. For example, we find a 10x improvement on `cifar-10` for a 3-hidden layer FCN with 1024 units per layer. However, for ResNet-20, a 10x improvement occurs only when the top-layer width d^L is similar to or larger than the number of observations, illustrating that gains are possible but not guaranteed outside of the NNGP regime.

1.1. Assumptions and notation

A BNN models a mapping from inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$ using a parametric function $f := f_\theta$. An example is an L-hidden layer fully-connected network $f_\theta = f^{L+1}$ with

$$f^\ell(x) = \frac{\sigma_W^\ell}{\sqrt{d^{\ell-1}}} h^{\ell-1}(x) W^\ell, \quad h^\ell(x) = \psi(f^\ell(x)), \quad (1)$$

with ψ the nonlinearity, $h^0(x) := x$, and $W^\ell \in \mathbb{R}^{d^{\ell-1} \times d^\ell}$ (bias terms are wrapped into W^ℓ by adding a constant entry dimension to $h^{\ell-1}(x)$). θ^ℓ will denote the flattened ℓ^{th} layer parameters, and $\theta := [\theta^\ell]_{\ell=1}^{L+1} \in \mathbb{R}^d$ their concatenation. Later on, the vector of readout weights θ^{L+1} will be of special importance, as will the **minimum hidden layer width** $\boldsymbol{d_{\min} := \min_{1 \le \ell \le L} d^\ell}$. While we will study how the behaviour of f and θ changes with layer width, we suppress this dependence in our notation to reduce clutter.

The factor $\sigma_W^\ell / \sqrt{d^{\ell-1}}$ in Equation (1) is more commonly part of the weight prior. We use this so-called 'NTK parametrisation' as it allows taking $\mathcal{N}(0, I)$ as the prior regardless of the network width, which simplifies our notation without changing the implied function space distribution. Our claims hold under the standard parametrisation as well, by making multiplication by $\sigma_W^\ell / \sqrt{d^{\ell-1}}$ a part of the reparametrisation.

We assume the final readout layer is linear, and that the likelihood is Gaussian $p(y \mid X, \theta) \propto \exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - f_\theta(x_i))^2\}$ with observation variance $\sigma^2 > 0$. In contrast, the rest of the network need not be an FCN, but can contain any layers for which NNGP behaviour is known, including convolutional and multi-head attention layers, and skip connections (see, e.g., Yang, 2020, for an overview).

2. Repriorisation: posterior → prior

2.1. Repriorisation of linear models

To begin, we consider a Bayesian linear model $y \mid \theta, x \sim \mathcal{N}(x^\top \theta, \sigma^2)$. For a standard normal prior $\theta \sim \mathcal{N}(0, I_d)$, the Bayesian posterior after observing n points, $\mathcal{D} := (X, y) \in \mathbb{R}^{n \times d} \times \mathbb{R}^n$, is a Gaussian $\theta \mid \mathcal{D} \sim \mathcal{N}(\mu, \Sigma)$ with

$$\Sigma = (I_d + \sigma^{-2} X^\top X)^{-1}, \text{ and } \mu = \sigma^{-2} \Sigma X^\top y. \quad (2)$$

We can thus *transform a posterior into a prior sample* using the data-dependent reparameterisation $\phi = T^{-1}(\theta) = \Sigma^{-1/2}(\theta - \mu) \sim \mathcal{N}(0, I_d)$ as illustrated in Figure 2.

2.2. Repriorisation of Bayesian neural networks

A similar insight applies to BNNs with the assumed Gaussian prior and likelihood. Specifically, the posterior of readout (top-layer) weights *conditioned* on the other parameters is $\theta^{L+1} \mid \theta^{\le L}, \mathcal{D} \sim \mathcal{N}(\mu, \Sigma)$ with μ and Σ as in

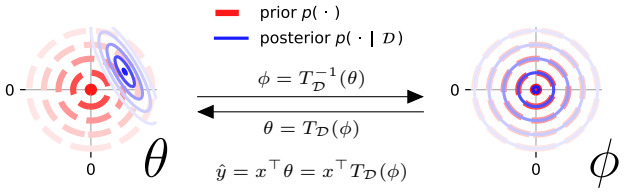

Figure 2. **Reparameterisation of Bayesian *linear* regression maps the posterior to the prior. Left:** Shows the parameter θ prior, and the posterior for a single datapoint $X = [0.9, 0.5]$, $y = [2]$, with $\sigma^2 = 0.1$. **Right:** After a data-dependent affine re-parameterisation, described in Section 2.1, the posterior becomes identical to the prior in terms of new parameters ϕ. MCMC sampling from the parameter posterior is easier in ϕ than in θ. A similar reparameterisation can be applied to deep BNNs.

Equation (2), except X is replaced by the scaled top-layer embeddings $\Psi \coloneqq {}^{\sigma_W^{L+1} h^L(X)}/_{\sqrt{d^L}} \in \mathbb{R}^{n \times d^L}$. Our reparametrisation $\phi = T^{-1}(\theta)$ can thus be defined analogously:[2]

$$\phi^\ell = \begin{cases} \Sigma^{-1/2}(\theta^\ell - \mu) & \ell = L+1, \\ \theta^\ell & \text{otherwise.} \end{cases} \tag{3}$$

This ensures $\phi^{L+1} \,|\, \phi^{\leq L}, \mathcal{D} \sim \mathcal{N}(0, I_{d^L})$ for any fixed value of $\phi^{\leq L} = \theta^{\leq L}$. We omit the dependence of μ, Σ on the dataset $\mathcal{D}$ and the pre-readout parameters $\theta^{\leq L}$ to reduce clutter, but emphasise that T depends on both.

Since T is a differentiable bijection, the full reparametrised density is $p(\phi \,|\, \mathcal{D}) = p(\theta \,|\, \mathcal{D}) \left|\det \partial_\phi \theta\right|$ where

$$\det(\partial_\phi \theta) = \det \begin{pmatrix} \Sigma^{1/2} & \frac{\partial \theta^{L+1}}{\partial \phi^{\leq L}} \\ 0 & I_{(d-d^L)} \end{pmatrix} = \sqrt{\det(\Sigma)}\,. \tag{4}$$

2.3. Interpreting the reparametrised distribution

To better understand the reparametrisation, note $p(\phi \,|\, \mathcal{D}) = p(\phi^{L+1} \,|\, \phi^{\leq L}, \mathcal{D})\, p(\phi^{\leq L} \,|\, \mathcal{D})$ where we already know the first term is equivalent to $\mathcal{N}(0, I_{d^L})$. Since $\phi^{\leq L} = \theta^{\leq L}$, the latter term is equal to the *marginal* posterior over $\theta^{\leq L}$

$$\begin{aligned} p(\phi^{\leq L} \,|\, \mathcal{D}) &\propto p(\theta^{\leq L})\; \mathbb{E}_{\theta^{L+1} \sim \mathcal{N}(0, I_{d^L})} \left[p(y \,|\, \theta, X)\right] \\ &\overset{\text{(i)}}{\propto} p(\theta^{\leq L}) \sqrt{\det \Sigma}\, \exp\{\tfrac{1}{2}\, y^\top (\sigma^2 I_n + \Psi\Psi^\top)^{-1} y\} \end{aligned} \tag{5}$$

where (i) comes from completing the square $\|\theta^{L+1} - \mu\|^2_{\Sigma^{-1}}$ ($\|v\|^2_A \coloneqq v^\top A v$), and applying the Woodbury identity.

Using the Weinstein–Aronszajn identity

$$\det(\Sigma) = \det(I_{d^L} + \sigma^{-2} \Psi^\top \Psi) \propto \det(\sigma^2 I_n + \Psi\Psi^\top)\,,$$

the readers familiar with the GP literature may recognise $p(\phi^{\leq L} \,|\, \mathcal{D})$ as $p(\theta^{\leq L})$ weighted by the marginal likelihood of a centred GP with the *empirical NNGP kernel* $\hat{K}_{\sigma^2} \coloneqq \sigma^2 I_n + \Psi\Psi^\top$ (Rasmussen & Williams, 2005). The marginal likelihood is often used for optimisation of kernel hyperparameters. Here the role of the 'hyperparameters' is played by $\theta^{\leq L}$, i.e., all but the readout weights.

[2] See Appendix E for discussion of alternative reparametrisations which modify parameter values in layers beyond readout.

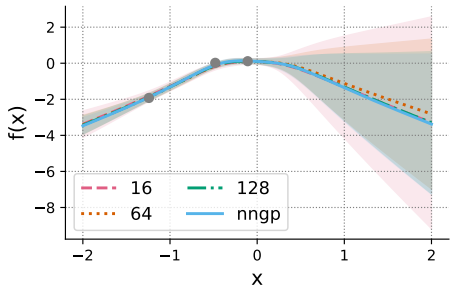

Figure 3. **The functions parametrised by $T(\phi)$, $\phi \sim \mathcal{N}(0, I)$, converge to the true posterior.** The distribution over functions $f_{T(\phi)}$ is shown for a 3-hidden layer FCN with 1D inputs and outputs, and 3 datapoints (dark circles). As layer width increases (legend), the BNN posterior converges to the NNGP posterior (Hron et al., 2020a), and so does $f_{T(\phi)}$ (our Proposition 2.2).

This relation to the empirical NNGP kernel is crucial in the next section where we exploit the known convergence of $\hat{K}_{\sigma^2}$ to a constant *independent* of $\theta^{\leq L}$ in wide BNNs, to prove that the reparametrised posterior converges to the prior distribution at large width. Figure 4 provides an informal argument motivating that same convergence, using the language of probabilistic graphical models (PGMs).

2.4. Asymptotic normality under reparametrisation

Our reparametrisation guarantees normality of the reparametrised posterior over ϕ^{L+1} for *any* width. Equation (5) suggests $\phi^{\leq L}$ may exhibit Gaussian behaviour as well in wide BNNs, since—outside of $p(\theta^{\leq L})$—its posterior depends on $\phi^{\leq L} = \theta^{\leq L}$ only through the empirical NNGP kernel which is known to converge a constant $K \in \mathbb{R}^{n \times n}$ as the layer width grows (see Yang, 2020, for an overview).

This is crucial in establishing our first major result: convergence of the KL divergence between the $\mathcal{N}(0, I_d)$ prior and the reparametrised posterior to zero as the number of hidden units in each layer goes to infinity.[3]

Theorem 2.1. *Let $P_{\phi \,|\, \mathcal{D}}$ be the reparametrised posterior distribution defined by the density $p(\phi \,|\, \mathcal{D})$. Assume the Gaussian prior and likelihood with $\sigma > 0$, and that $\hat{K} \xrightarrow{\mathbb{P}} K$ and $\mathbb{E}[\hat{K}] \to K$ under the prior as $d_{\min} \to \infty$.*

Then $\mathrm{KL}(\mathcal{N}(0, I_d) \,\|\, P_{\phi \,|\, \mathcal{D}}) \to 0$ *as* $d_{\min} \to \infty$.

The assumptions ($\hat{K} \xrightarrow{\mathbb{P}} K$, $\mathbb{E}[\hat{K}] \to K$, as $d_{\min} \to \infty$) hold for most common architectures, including FCNs, CNNs, and attention networks (Matthews et al., 2018; Garriga-Alonso et al., 2019; Hron et al., 2020b; Yang, 2020).

[3] Please refer to Appendix A for all proofs.

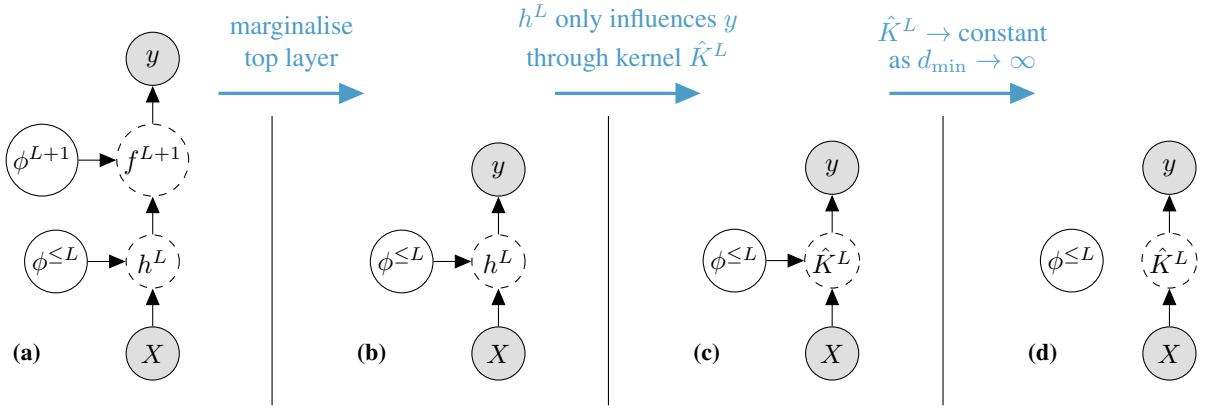

Figure 4. **Sketch motivating convergence of the reparametrised posterior to the prior.** **(a)** A probabilistic graphical model (PGM) describing the BNN *after reparameterisation*. Shaded circles correspond to observed variables. Dashed outlines indicate that the variable value is a deterministic function of its ancestors. The reparametrised posterior factorises as $p(\phi \mid \mathcal{D}) = p(\phi^{L+1} \mid \phi^{\leq L}, \mathcal{D})\, p(\phi^{\leq L} \mid \mathcal{D})$. **(b)** Our reparametrisation in Equation (3) makes $\phi^{L+1} \mid \phi^{\leq L}, \mathcal{D} \sim \mathcal{N}(0, I)$ for any $\phi^{\leq L}$. The marginal $p(\phi^{\leq L} \mid \mathcal{D}) = \int p(\phi \mid \mathcal{D})\, \mathrm{d}\phi^{L+1}$ can be evaluated analytically with a Gaussian likelihood (Equation (5)). **(c)** $p(\phi^{\leq L} \mid \mathcal{D})$ depends on the data only via the (empirical) NNGP kernel $\hat{K}^L = h^L(X)h^L(X)^\top / d^L$. This is due to the assumed linearity of the top layer, the Gaussian prior, and the identity $A\varepsilon \sim \mathcal{N}(0, AA^\top)$ for $\varepsilon \sim \mathcal{N}(0, I)$ and any fixed matrix A. **(d)** As layer width $d_{\min}$ goes to infinity, $\hat{K}^L$ converges to a constant independent of $\phi^{\leq L}$ (Section 2.4). Hence $p(\phi^{\leq L} \mid \mathcal{D}) \approx p(\phi^{\leq L}) = \mathcal{N}(0, I)$ by the PGM. Since both ϕ^{L+1} and $\phi^{\leq L}$ are approximately Gaussian, and ϕ^{L+1} is independent of $\phi^{\leq L}$, the joint posterior is also approximately Gaussian $p(\phi \mid \mathcal{D}) = p(\phi^{L+1} \mid \phi^{\leq L}, \mathcal{D}) p(\phi^{\leq L} \mid \mathcal{D}) \approx \mathcal{N}(0, I)$.

The fact that KL divergence does not increase under measurable transformations (e.g., Gray, 2011, corollary 5.2.2), combined with Theorem 2.1, implies that the KL divergence between the distribution of $T(\phi)$ with $\phi \sim \mathcal{N}(0, I_d)$ and the posterior $P_{\theta \mid \mathcal{D}}$ also converges to zero. This provides a simple way of approximating wide BNN posteriors as illustrated by Figure 3. We use the same argument to establish convergence in *function space* by showing measurability of the $\theta \mapsto f_\theta$ mapping w.r.t. a suitable σ-algebra.

Proposition 2.2. *Let $\phi \sim \mathcal{N}(0, I_d)$, $\theta \sim P_{\theta \mid \mathcal{D}}$, and denote the functions they parametrise by $f_{T(\phi)}$ and f_θ. Assume all nonlinearities are continuous, and each layer's outputs are jointly continuous in the layer parameters and inputs.*

Then $\mathrm{KL}(P_{f_{T(\phi)}} \,\|\, P_{f_\theta \mid \mathcal{D}}) \to 0$ *as* $d_{\min} \to \infty$*, where the* KL *is defined w.r.t. the usual product σ-algebra on $\mathbb{R}^{\mathcal{X}}$.*

Hron et al. (2020a) showed that for continuous bounded likelihoods, $P_{f_\theta \mid \mathcal{D}}$ converges (weakly) to the NNGP posterior whenever P_{f_θ} with $\theta \sim P_\theta$ converges to the NNGP prior. This implies that the $P_{f_{T(\phi)}}$ from Proposition 2.2 converges (weakly) to the NNGP posterior whenever $P_{f_\theta \mid \mathcal{D}}$ does: by Pinsker's inequality, we have convergence in total variation which implies convergence of expectations of all bounded measurable (incl. bounded continuous) functions.

Our Theorem 2.1 may moreover be seen as an appealing answer to the issue of finding a useful notion of 'convergence' in parameter space of an *increasing* dimension from (Hron et al., 2020a, section 4). As the authors themselves point out, their approach of embedding the weights θ in $\mathbb{R}^{\mathbb{N}}$, and studying weak convergence w.r.t. its usual product σ-algebra, 'tells us little about behaviour of *finite* BNNs'. This is most clearly visible in their proposition 2, which establishes asymptotic reversion of the parameter space posterior to the *prior* (without any reparametrisation), which we know does not induce the correct function space limit.

We note that our Theorem 2.1 does not contradict the Hron et al.'s result exactly because their definition of convergence essentially only captures the *marginal* behaviour of finite weight subsets, whereas our KL divergence approach characterises the *joint* behaviour of all the weights. Indeed, our next proposition shows that reversion to the prior does not occur under our stronger notion of convergence.

Proposition 2.3. *Under the assumptions of Theorem 2.1*

$$\mathrm{KL}(\,\mathcal{N}(0, I_d) \,\|\, P_{\theta \mid \mathcal{D}}\,) \\ \to \tfrac{1}{2}[\sigma^{-2}\mathrm{Tr}(K) + \|y\|^2_{\sigma^{-2}I_n - K^{-1}_{\sigma^2}} - \log\det(\sigma^{-2}K_{\sigma^2})]\,,$$

as $d_{\min} \to \infty$. The limit is equal to the KL divergence between the NNGP prior and posterior in function space, defined w.r.t. the usual product σ-algebra on $\mathbb{R}^{\mathcal{X}}$.

3. Faster mixing via reprioritisation

We now have a reparametrisation which *provably* makes the reparametrised posterior close to the standard normal prior $\mathcal{N}(0, I_d)$. Since sampling from the BNN posterior is notoriously difficult, the closeness to $\mathcal{N}(0, I_d)$ suggests sampling from $P_{\phi \mid \mathcal{D}}$ may be significantly easier.

However closeness in KL divergence does not guarantee that the gradients of $p(\phi \mid \mathcal{D})$ are as well behaved as those of the

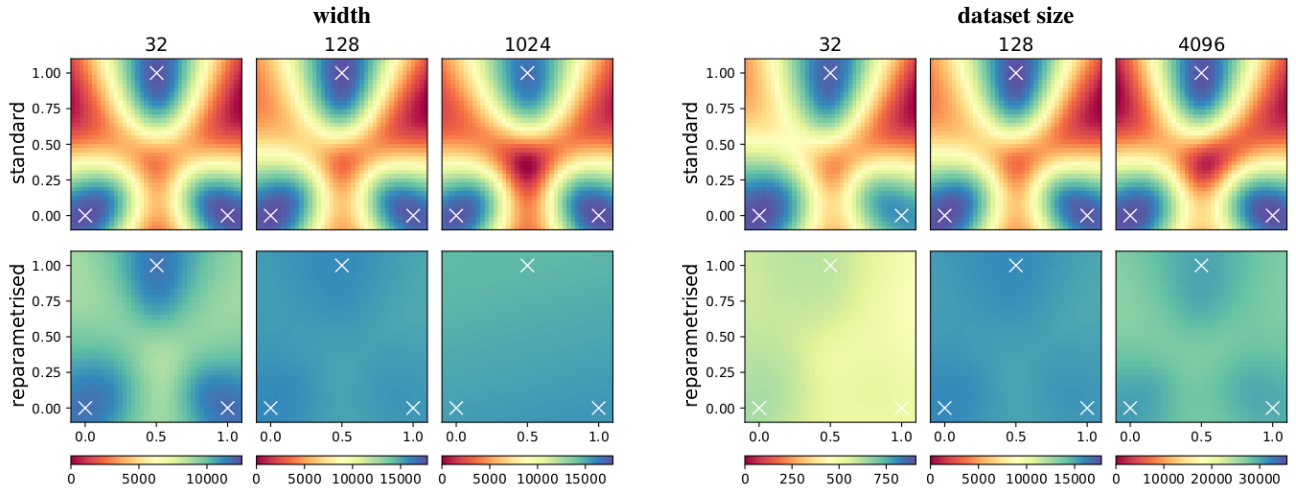

Figure 5. **The posterior energy landscape is smoother after reprioritisation.** Plots show 2D slices of the log posterior on `cifar-10` in terms of the original parameters θ (top row) and the reparametrised $\phi = T^{-1}(\theta)$ (bottom row), for a 1-hidden layer FCN. The 2D slices are obtained by spherical linear interpolation between three parameter values found by gradient descent training on CIFAR-10 after random initialisation; norms may differ, and are therefore interpolated linearly. **Left:** Columns represent varying hidden layer width. The number of observations is fixed at $n = 128$. The vanishing structure of the reparametrised posterior as width increases, making mixing between the normally separated modes easier. **Right:** Columns represent varying dataset size, width is fixed at $d^1 = 128$. As the number of datapoints becomes large compared to d^1, the reparametrised posterior gradually grows less smooth.

standard normal distribution, which is crucial for gradient-guided methods like Langevin Monte Carlo (LMC). In Section 3.1, we provide partial reassurance by proving that the ℓ^2-norm of the differences between the two gradients vanishes in probability as the layers grow wide. Section 3.2 then provides a simple-to-implement way of computing both the reparametrised $\theta = T(\phi)$ and the corresponding density $p(\phi \mid \mathcal{D})$ at a fraction of cost of the naive implementation.

3.1. Convergence of density gradients

Our second major result proves that the *posterior* mass of the region where $\nabla_\phi \log p(\phi \mid \mathcal{D})$ significantly differs from the gradient of the $\mathcal{N}(0, I)$ density vanishes in wide BNNs.

Proposition 3.1. *Let $\Delta_\phi := -\phi - \nabla_\phi \log p(\phi \mid \mathcal{D})$ where $-\phi$ is the gradient of the log density of $\mathcal{N}(0, I)$. Assume the conditions in Theorem 2.1, and continuous nonlinearities.*

Then $P_{\phi \mid \mathcal{D}}(\|\Delta_\phi\|_2 > \epsilon) \to 0$ as $d_{\min} \to \infty$, $\forall \epsilon > 0$.

Two more technical assumptions are in Appendix A.4. Albeit useful, Proposition 3.1 does not ensure gradient-guided samplers like LMC will not dawdle in regions with ill-behaved gradients. This behaviour did not occur in our experiments, but the caveat is worth remembering.

To provide a rough quantitative intuition for the scale of the speed up, we show our reparametrisation enables $\sqrt{n}/\sigma$-times higher LMC stepsize relative to no reparametrisation in *linear* models, without any drop in the integrator accuracy (see Appendix D).

3.2. Practical implementation

To apply LMC, we need an efficient way of computing both $\theta = T(\phi)$ and the corresponding gradient of $\log p(\phi \mid \mathcal{D})$. Since our largest experiments require millions of steps, the naive approach of computing each of μ, $\Sigma^{1/2}$, and $\det(\Sigma^{1/2})$ separately—see Equations (3) and (4)— for the cost of *three* $\mathcal{O}[(d^L)^3]$ operations is worth improving upon. We propose a three-times more efficient alternative.

The trick is to first compute the Cholesky decomposition $U^\top U = \sigma^2 I_{d^L} + \Psi^\top \Psi$, and then reuse U in computation of all three terms. This can be done by observing $\Sigma = \sigma^2 U^{-1}(U^{-1})^\top$, which implies $\mathbb{V}[\sigma U^{-1}\varepsilon] = \Sigma$ for $\varepsilon \sim \mathcal{N}(0, I)$. We can thus use the *alternative* reparametrisation

$$\theta^{L+1} = U^{-1}[(U^\top)^{-1}\Psi^\top y + \sigma\phi^{L+1}],$$

which uses the fact $\mu = (\sigma^2 I + \Psi^\top\Psi)^{-1}\Psi^\top y$ together with the above definition of U. The determinant can then be computed essentially for free since

$$\det(\partial_\phi \theta) \overset{\text{(i)}}{=} \det(\sigma U^{-1}) \overset{\text{(ii)}}{=} \frac{\sigma^{d^L}}{\det(U)} \overset{\text{(iii)}}{=} \frac{\sigma^{d^L}}{\prod_i U_{ii}},$$

where (i) is as in Equation (4), (ii) uses standard determinant identities, and (iii) exploits that U is triangular.

Because U is upper-triangular, $U^{-1}v$ (resp. $(U^\top)^{-1}v$) can be efficiently computed by back (resp. forward) substitution for any v (Voevodin & Kuznetsov, 1984). The sequential application of forward and backward substitution is known as the Cholesky solver algorithm (Cholesky, 1924). Since

we would need to apply the Cholesky solver to compute $\mu = (\sigma^2 I + \Psi^\top \Psi)^{-1} \Psi^\top y$ anyway, we obtain both the reparametrisation the density at essentially the same cost.

Figure 6 demonstrates our approach introduces little wall-time clock overhead on modern accelerators relative to no reparametrisation when combined with LMC. Since the per-step computational overhead is never larger than 1.5x in Figure 6 and we observe $\geq$10x faster mixing in Section 4, the cost is more than offset by the sampling speed up.

The above ('feature-space') reparametrisation scales as $\mathcal{O}[(d^L)^3]$ in computation and $\mathcal{O}[(d^L)^2]$ in memory. This is typically much better than exact (NN)GP inference which scales as $\mathcal{O}(n^3)$ in computation and $\mathcal{O}(n^2)$ in memory. To enable future study of very wide BNNs ($d^L \gg n$), we provide an alternative 'data-space' formulation with $\mathcal{O}(n^3)$ computation and $\mathcal{O}(n^2)$ memory scaling in Appendix B.

Interpolating between parametrisations

We can swap the likelihood variance σ^2 for a general regulariser $\lambda > 0$ in the definition of $U^\top U := \lambda I_{d^L} + \Psi^\top \Psi$. Clearly, $T \to \mathrm{Id}$ (identity mapping) as $\lambda \to \infty$. This allows interpolation between our proposed reparametrisation at $\lambda = \sigma^2$, and the standard parametrisation at $\lambda = \infty$, which may be useful when far from the NNGP regime. Proper tuning of the hyperparameter λ thus ensures the algorithm never performs worse than under the standard parametrisation.

4. Experiments

Armed with theoretical understanding and an efficient implementation, we now demonstrate that our reparametrisation can dramatically improve BNN posterior sampling, as quantified by per-step effective sample size (ESS; Ripley, 1989), and the $\hat{R}$ statistics (Gelman & Rubin, 1992), over a range of architectural choices and dataset sizes.

4.1. Setup

4.1.1. $\hat{R}$ diagnostic

$\hat{R}$ is a standard tool for testing non-convergence of MCMC samplers. It takes M collections of samples $\{z_{mi}\}_{i=1}^S \subset \mathbb{R}$ produced by *independent* chains, and computes

$$\hat{R}^2 = \frac{\hat{\mathbb{E}}\big[\hat{\mathbb{V}}(z_{mi} \,|\, m)\big] + \hat{\mathbb{V}}\big[\hat{\mathbb{E}}(z_{mi} \,|\, m)\big]}{\hat{\mathbb{E}}\big[\hat{\mathbb{V}}(z_{mi} \,|\, m)\big]}, \qquad (6)$$

where $\hat{\mathbb{E}}$ and $\hat{\mathbb{V}}$ compute expectation and variance w.r.t. the empirical distribution which assigns a $\frac{1}{MS}$ weight to each z_{mi} (conditioning on m yields within-chain statistics). The numerator is just $\hat{\mathbb{V}}(z_{mi})$ by the law of total variance, where $\hat{\mathbb{V}}\big[\hat{\mathbb{E}}(z_{mi} \,|\, m)\big]$ vanishes when all chains sample from the same distribution. Values significantly larger than one thus indicate non-convergence. We follow Izmailov et al. (2021) in reporting $\hat{R}^2$ (denoted by $\hat{R}$ in their paper), instead of the square root of the above quantity. For multi-dimensional random variables, we estimate $\hat{R}^2$ for each dimension, and report the resulting distribution over dimensions.

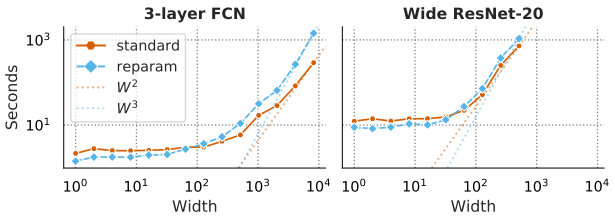

Figure 6. **Sampling with reprioritisation is comparable to the baseline in wall-clock time.** While our reparametrisation cost is cubic in top layer width d^L (Section 3.2), it is notable only for very large FCNs (**left**), and even larger CNNs (**right**) where the quadratic cost of the forward pass dominates due to a large multiplicative constant. See Appendix B for an alternative formulation which scales only quadratically with d^L but cubically with n.

4.1.2. Effective sample size

ESS is a measure of sampling speed. It estimates lag k autocorrelations $\hat{\rho}_k := \hat{\mathbb{C}}(z_{mi}, z_{m(i+k)}) / \hat{\mathbb{V}}(z_{mi})$ for $k = 1, \ldots, S-1$ ($\hat{\mathbb{C}}$ is the empirical covariance), and computes

$$\widehat{\mathrm{ESS}} := \frac{S}{1 + 2\sum_{k=1}^{S-1}(1 - \frac{k}{S})\hat{\rho}_k}, \qquad (7)$$

The per-step ESS is then $\widehat{\mathrm{ESS}}/S$. Since $\hat{\rho}_k$ estimates for high k are based on only $M(S-k)$ samples, we denoise by stopping the sum in Equation (7) at the first $\hat{\rho}_k < 0$, which is the default in Tensorflow Probability's `effective_sample_size` (Dillon et al., 2017).

Equation (7) is based on the $\mathbb{V}(\bar{z}_S) = \mathbb{V}(z_1)/S$ identity for (population) variance of the empirical mean $\bar{z}_S$ of S *i.i.d.* variables z_i Since MCMC chains produce dependent samples, $\widehat{\mathrm{ESS}}$ estimates the number which would satisfy the above equality when substituted for S under the assumption of stationarity. Because we need to estimate ESS in the very high-dimensional parameter space, we randomly choose *100 one-dimensional subspaces*, project the samples, estimate ESS in each of them, and report their distribution.

4.1.3. Data and hyperparameters

The underdamped LMC sampler (Rossky et al., 1978) and `cifar-10` dataset (Krizhevsky et al., 2009) are used in all experiments. We use regression on one-hot encoding of the labels shifted by $-\frac{1}{10}$ so that the label mean of each example is zero, matching the BNN prior (extension to multi-dimensional outputs described in Appendix C). Gaussian likelihood with $\sigma = 0.1$ is used throughout, which we selected to ensure that (a) positive and negative labels are separated by at least 4σ, but (b) σ is not too small to give

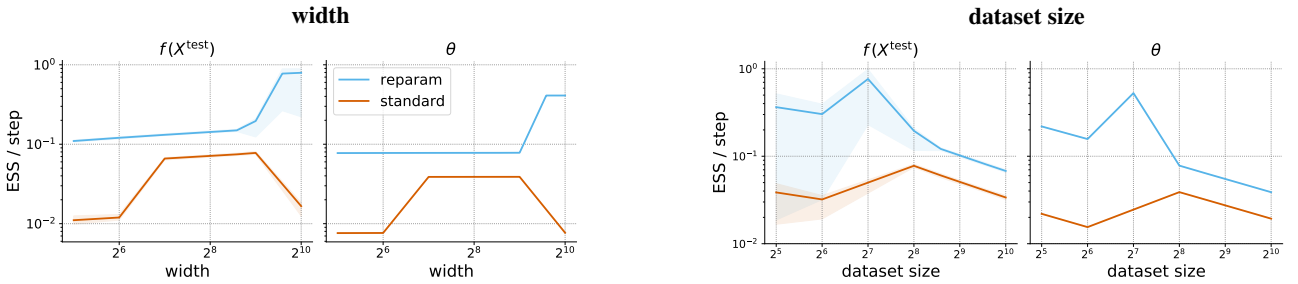

Figure 7. **Reprioritisation makes mixing speed faster, and the performance benefits increase as the width to dataset size ratio increases.** Per-step ESS as a function of layer width (left) and dataset size (right) for a 3-hidden layer FCN. The line indicates the mean, and the bands the *minimum* and *maximum*, ESS over the 100 projection subspaces (see Section 4.1.2). In all cases, reparametrisation achieves significantly better mixing speed (note the log-log scale of the axes). `3M` samples collected for each configuration. Hyperparameters were tuned for each configuration separately (see Section 4.2.1). **Left:** Dataset size fixed at $n = 256$. The benefit of reparametrisation increases with width, especially so when width becomes larger than dataset size where reparametrisation yields 50x higher ESS. **Right:** Width fixed at 512. Note that reparameterisation leads to a higher ESS for all configurations.

ourselves an unfair advantage based on the linear case intuition (Appendix D), or cause numerical issues. Sample thinning is applied in all experiments, and ESS and $\hat{R}$ are always computed based on the thinned sample. However, Figures 8 and 9 visualise ESS evolution as a function of the number of LMC steps which is equal the *unthinned* sample size.

As common, we omit the Metropolis-Hastings correction of LMC (Neal, 1993), and instead tune hyperparameters so that average acceptance probability after burn-in stays above 98%. We observed significant impact of numerical errors on both the proposal and the calculation of the acceptance probability; we use `float32` which offers significant speed advantages over `float64`, but can still suffer 2-5 percentage point changes in acceptance probability compared to `float64` computation with the same candidate sample.

Our experiments are implemented in JAX (Bradbury et al., 2018), and rely on the Neural Tangents library (Novak et al., 2020) for both implementation of the NN model logic, and evaluation of the NNGP predictions (e.g., in Figure 3).[4]

4.2. Results

In Figure 5 we visualize 2D slices through the weight posterior, for the original and reparameterized network for a 1-hidden layer FCN. The log posterior is far smoother after reparametrization, and its relative smoothness improves as network width is increased relative to dataset size. As we quantify in the following sections, this enables dramatic improvements in sampling efficiency.

4.2.1. Dependence on width and dataset size

In Figure 7, we explore the dependence of mixing speed of the LMC sampler on network width and dataset size as measured by the per-step ESS. We use a single chain run of `3M` steps with a `20K` burn-in per each width/training set size. The *results are computed on a sample thinned by a factor of 25*. We use a FCN with three equal width hidden layers and GELU nonlinearities (Hendrycks & Gimpel, 2016), and the weight scaling described in Section 1.1. The weight and bias scaling factors are set to $\sigma_W^2 = 2$, $\sigma_b^2 = 0.01$ everywhere except in the readout layer where $\sigma_W^2 = 1$ to achieve approximately unit average variance of the network predictions under the prior (at initialisation). For the width experiments (Figure 7, left), the dataset size was fixed at $n = 256$, and the layer width varied over $\{32, 64, 128, 384, 512, 768, 1024\}$.

For each width and both parametrisations, we *separately* tuned the LMC stepsize and damping factor to maximise the ESS of θ while satisfying the $> 98\%$ mean acceptance probability criterion. For the reparametrised sampler, we also tried to tune the regularisation parameter λ (see Section 3.2), but found that the value dictated by our wide BNN theory ($\lambda = \sigma^2$) worked best even for the smallest width models. The distribution of ESS over the random projections of θ and $f(X^{\text{test}})$ on `5K` test points is shown, with its mean, minimum, and maximum values.

Our reparametrisation achieves significantly faster mixing in all conditions (note the log-log scale). The improvement becomes larger as the BNN grows wider (left), achieving near-perfect values for the widest BNNs ($10^0 = 1$ is maximum) where the width ($> 2^8$) is larger than the dataset ($256 = 2^8$), with a 50x speed up at width 1024.

For the dataset size plots (right), the performance is again best when the ratio of width to the number of observations is the smallest (here on the left). The ESS of both $f(X^{\text{test}})$ and θ decays with dataset size as expected, since the posterior becomes more complicated, and the width to dataset size ratio dips below one, i.e., we are leaving the NNGP regime. The reparametrisation however maintains higher ESS even outside of the conditions where our theory holds.

[4]Code: github.com/google/wide_bnn_sampling.

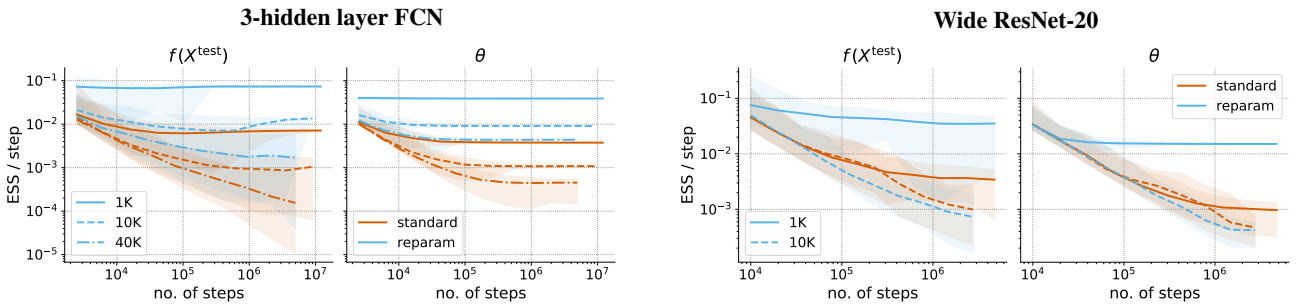

Figure 8. **Reprioritisation results in faster mixing across architectures.** Per-step ESS estimates are plotted as a function of the number of sampler steps for three random subsets of `cifar-10` of size $n = 1024, 10240, 40960$. The line indicates the mean, and the bands the *minimum* and *maximum*, ESS over the 100 projection subspaces (see Section 4.1.2). The initial downward trend of the curves is due to underestimation of long range autocorrelations at earlier steps where sample size is too small. **Left:** Results for a FCN with three width 1024 hidden layers. The non-reparametrised BNNs mix around 10x slower for all three dataset sizes, and up to 200x slower if the worst mixing projections (bottom of shaded regions) are compared to each other for each case. **Right:** Results for a normaliser-free Wide ResNet-20 with 128 and 512 units (channels) in the narrowest and widest layers (see Section 4.2.2). For $n =$ `1K`, reparameterisation performs more than 10x better. For $n =$ `10K`, both chains have very similar ESS / step values. This could be because the dataset size is now two orders of magnitude larger than the width, a regime in which our theory does not apply. It could also be because, even after 3 million sampling steps, the chains are so far from equilibrium and the ESS estimates are dominated by the initial transient effects.

4.2.2. Dependence on architecture

While our theory does not cover the cases where the ratio of dataset size to width is large, the consistent advantage of reparametrisation even in this regime is intriguing (Figure 7, right). In this section, we investigate the dependence of this phenomenon on the architecture and dataset size.

For the architecture, we compare the FCN from Section 4.2.1 with a normaliser-free Wide ResNet-20 with 128, 256, and 512 channels arranged from bottom to the top layer in the usual three groups of residual layer blocks (He et al., 2016; Zagoruyko & Komodakis, 2016). GELU is used for both.

As in (Izmailov et al., 2021), we remove batch normalisation since it makes predictions depend on the mini-batch, which interferes with Bayesian interpretation. We however use a different alternative in its place, namely the proposal of Shao et al. (2020), which replaces each residual sum $f_k^{\text{skip}}(x) + f_k^{\text{resid}}(x)$ by $\sqrt{k-1+c/k+c}\, f_k^{\text{skip}}(x) + \sqrt{1/k+c}\, f_k^{\text{resid}}(x)$ where k is the index of the skip connection, and c is a hyper-parameter we set to 1. Since (Izmailov et al., 2021) is the closest comparison for this section, we note the authors use a categorical instead of a Gaussian likelihood[5], and an i.i.d. $\mathcal{N}(0, \alpha)$ prior for all parameters, but *without* scaling the weights by $\sigma_W^\ell/\sqrt{d^\ell}$ as we do (see Section 1.1).

For the scaling factors, FCN is as in Section 4.2.1, whereas the ResNet uses $\sigma_W^2 = 2.2$ and $\sigma_b^2 = 0$, except in the last layer where $\sigma_W^2 = 1$ and $\sigma_b^2 = 0.01$ is again employed to ensure reasonable scale of the network predictions under the prior. As a sanity check, using this prior as initialiser, and optimising with Adam, this normaliser-free ResNet achieves a decent 95.6% validation accuracy on full `cifar-10`. We tuned the LMC stepsize and damping factor separately for every configuration to maximise ESS of θ while satisfying $> 98\%$ mean acceptance rate after burn-in. For the reparametrised chain, the regulariser λ was σ^2 and $n\sigma^2$ respectively for the FCN and the ResNet, with $\sigma^2 = 0.01$ the output variance. The *sample thinning factor is 100*.

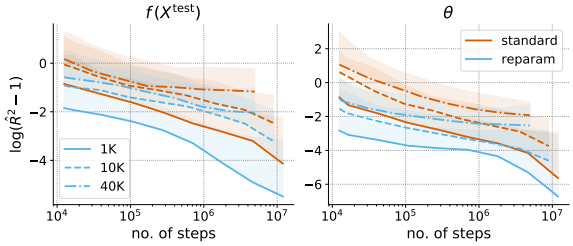

Figure 9. **Convergence of $\hat{R}$ with varying dataset size.** Complements the left side of Figure 8. Line represents mean, band the maximum value. Estimates based on three independent chains. Note the log-log scale, and that $\hat{R}^2$ is plotted relative to its unit lower bound. High $\hat{R}^2$ indicate non-convergence so lower is better.

As can be seen in Figure 8, the mixing speed advantage of reparametrisation persists far from the NNGP regime in the case of the FCN (left), where even a factor of forty in dataset to width ratio did not erase the surprisingly consistent 10x higher (average) ESS. This observation is supported by Figure 9, in which $\hat{R}^2$ is already near one (optimum) when the standard parametrisation still exhibits $\mathcal{O}(10^2)$ values.

For the Wide ResNet (right), we observed a similar $\geq$10x benefit from reparameterisation at `1K` datapoints. For the $n =$`10K` case, no benefit was observed. This may be be-

[5]Our reparameterization also enables accelerated sampling for categorical likelihood – see Appendix F.

cause both chains are far from equilibrium even after 3M steps, and the per-step ESS measurement reflects initial transients. It may also be because the dataset size (10,240) was much larger than smallest channel count (128), exceeding by two orders of magnitude the scale where the model is expected to approximate the NNGP limit.

5. Other related work

For ensembles of infinitely-wide NNs trained with gradient descent, Shwartz-Ziv & Alemi (2020) find the KL divergence between the posterior and the prior to diverge linearly with training time, an interesting departure from the finite value we find for Bayesian inference in Proposition 2.3. The information content of infinitely-wide NNs was analysed by Bernstein & Yue (2021) using the NNGP posterior, leading to non-vacuous PAC-Bayes generalisation bounds, adding to an active line of work focusing on generalisation bounds for overparametrised models (see, e.g., Dziugaite & Roy, 2017; Valle-Perez et al., 2018; Vakili et al., 2021).

If some hidden layers are narrow and remain finite, the resulting model is a type of Deep Gaussian process (Damianou & Lawrence, 2013; Agrawal et al., 2020) or Deep kernel process (Aitchison et al., 2021). Aitchison (2020) argues that such models may exhibit improved performance relative to uniformly wide BNNs owing to a form of representation learning. While we do not investigate such questions in this work, the favourable computational properties of our LMC algorithm could facilitate such analyses in the future.

Moving away from the infinite width limit, Yaida (2020) and Roberts et al. (2021) show how to compute the Bayesian prior and posterior systematically in powers of $1/\text{width}$, with the leading-order term given by the NNGP, though the subleading corrections pose computational challenges. For some architectures, a Gaussian prior in parameter space leads to closed-form expressions for the prior in function space, even at finite width (Zavatone-Veth & Pehlevan, 2021). Less is known about the posterior for finite BNNs, since computational considerations often require approximate inference, whose quality in real-world problems has recently been investigated in (Izmailov et al., 2021).

Our work is partially inspired by a thread of papers which accelerate MCMC sampling by applying an invertible transformation to a distribution's variables, and then running sampling chains in the transformed space (e.g., El Moselhy & Marzouk, 2012; Marzouk & Parno, 2014; Parno, 2015; Marzouk et al., 2016; Titsias, 2017; Hoffman et al., 2018).

6. Conclusion

We introduced *repriorisation*, a reparameterisation that (1) provides a rigorous characterisation of wide BNN parameter space, and (2) enables a more efficient BNN sampling algorithm which mixes *faster* as the network is made wider.

Our theory shows BNN posteriors exhibit non-negligible interactions between parameters in different layers, even at large width. In contrast, many popular BNN posterior approximations (e.g., mean-field methods) assume independence between parameters in different layers. Beyond MCMC, algorithms which incorporate between-layer interactions—from Laplace (Tierney & Kadane, 1986) to more recent ones like (Ober & Aitchison, 2021)—are therefore better positioned to capture the true BNN posterior behaviour. In fairness, approximation fidelity and downstream performance are not the same thing though, as clearly demonstrated by the success of non-Bayesian approaches.

Our sampling results parallel the interplay between optimisation and model size in deterministic NNs. In particular, first-order methods used to be considered ill-suited for the non-convex loss landscapes of NNs, yet experimental and later theoretical results showed they can be highly effective, especially for large NNs (e.g., Allen-Zhu et al., 2019; Du et al., 2019). Similarly, MCMC theory tells us that sampling is often harder in high dimensions, yet our results show that a simple reparametrisation exploiting the particular structure of wide BNN posteriors makes sampling much easier.

While the gap between deterministic and Bayesian NNs remains considerable, we hope our work enables further progress for practical large-scale BNNs.

References

Agrawal, D., Papamarkou, T., and Hinkle, J. Wide neural networks with bottlenecks are deep Gaussian processes. *JMLR*, 2020.

Aitchison, L. Why bigger is not always better: on finite and infinite neural networks. In *ICML*, 2020.

Aitchison, L., Yang, A., and Ober, S. W. Deep kernel processes. In *ICML*, 2021.

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *ICML*, 2019.

Bernstein, J. and Yue, Y. Computing the information content of trained neural networks. *arXiv*, 2021.

Billingsley, P. *Convergence of probability measures*. Wiley, second edition, 1999.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. *Handbook of Markov chain Monte Carlo*. CRC press, 2011.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. *NeurIPS*, 2019.

Cholesky, A.-L. Note sur une méthode de résolution des équations normales provenant de l'application de la méthode des moindres carrés a un système d'équations linéaires en nombre inférieur a celui des inconnues. — application de la méthode a la résolution d'un système defini d'équations linéaires. *Bulletin géodésique*, 1924.

Damianou, A. and Lawrence, N. D. Deep Gaussian processes. In *AISTATS*, 2013.

Daxberger, E., Nalisnick, E., Allingham, J. U., Antoran, J., and Hernandez-Lobato, J. M. Bayesian deep learning via subnetwork inference. In *ICML*, 2021.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. Tensorflow distributions. *arXiv*, 2017.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *ICML*, 2019.

Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. Efficient and scalable Bayesian neural nets with rank-1 factors. In *ICML*, 2020.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *UAI*, 2017.

El Moselhy, T. A. and Marzouk, Y. M. Bayesian inference with optimal maps. *Journal of Computational Physics*, 2012.

Garriga-Alonso, A., Rasmussen, C. E., and Aitchison, L. Deep convolutional networks as shallow Gaussian processes. In *ICLR*, 2019.

Gelman, A. and Rubin, D. B. Inference from iterative simulation using multiple sequences. *Statistical Science*, 1992.

Gray, R. M. *Entropy and Information Theory*. Springer, second edition, 2011.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (GELUs). *arXiv*, 2016.

Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., and Vasudevan, S. NeuTra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport. *AABI*, 2018.

Hron, J., Bahri, Y., Novak, R., Pennington, J., and Sohl-Dickstein, J. Exact posterior distributions of wide Bayesian neural networks. *UDL*, 2020a.

Hron, J., Bahri, Y., Sohl-Dickstein, J., and Novak, R. Infinite attention: NNGP and NTK for deep attention networks. In *ICML*, 2020b.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. G. What are Bayesian neural network posteriors really like? In *ICML*, 2021.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *NeurIPS*, 2018.

Khan, M., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In *ICML*, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images, 2009.

Lee, J., Sohl-dickstein, J., Pennington, J., Novak, R., Schoenholz, S., and Bahri, Y. Deep neural networks as Gaussian processes. In *ICLR*, 2018.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *NeurIPS*, 2019.

Levenberg, K. A method for the solution of certain nonlinear problems in least squares. *Quarterly of applied mathematics*, 1944.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. In *NeurIPS*, 2019.

Marquardt, D. W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 1963.

Marzouk, Y. and Parno, M. Transport map-accelerated Markov chain Monte Carlo for Bayesian parameter inference. In *AGU Fall Meeting Abstracts*, 2014.

Marzouk, Y., Moselhy, T., Parno, M., and Spantini, A. An introduction to sampling via measure transport. *arXiv*, 2016.

Matthews, A., Hensman, J., Turner, R., and Ghahramani, Z. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *AISTATS*, 2016.

Matthews, A., Hron, J., Turner, R., and Ghahramani, Z. Sample-then-optimize posterior sampling for Bayesian linear models. In *AABI*, 2017.

Matthews, A., Hron, J., Rowland, M., Turner, R. E., and Ghahramani, Z. Gaussian process behaviour in wide deep neural networks. In *ICLR*, 2018.

Neal, R. M. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, University of Toronto, 1993.

Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-dickstein, J. Bayesian deep convolutional networks with many channels are Gaussian processes. In *ICLR*, 2019.

Novak, R., Xiao, L., Hron, J., Lee, J., Alemi, A. A., Sohl-Dickstein, J., and Schoenholz, S. S. Neural Tangents: Fast and easy infinite neural networks in Python. In *ICLR*, 2020. URL https://github.com/google/neural-tangents.

Novak, R., Sohl-Dickstein, J., and Schoenholz, S. S. Fast finite width neural tangent kernel. In *ICML*, 2022.

Ober, S. W. and Aitchison, L. Global inducing point variational posteriors for Bayesian neural networks and deep Gaussian processes. In *ICML*, 2021.

Osawa, K., Swaroop, S., Khan, M. E. E., Jain, A., Eschenhagen, R., Turner, R. E., and Yokota, R. Practical deep learning with Bayesian principles. In *NeurIPS*, 2019.

Parno, M. D. *Transport maps for accelerated Bayesian computation*. PhD thesis, MIT, 2015.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

Ripley, B. Stochastic simulation. *Statistical Papers*, 1989.

Roberts, D. A., Yaida, S., and Hanin, B. The principles of deep learning theory. *arXiv*, 2021.

Rossky, P. J., Doll, J. D., and Friedman, H. L. Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 1978.

Shao, J., Hu, K., Wang, C., Xue, X., and Raj, B. Is normalization indispensable for training deep neural network? In *NeurIPS*, 2020.

Shwartz-Ziv, R. and Alemi, A. A. Information in infinite ensembles of infinitely-wide neural networks. In *AABI*, 2020.

Tierney, L. and Kadane, J. B. Accurate approximations for posterior moments and marginal densities. *Journal of the American statistical association*, 1986.

Titsias, M. K. Learning model reparametrizations: Implicit variational inference by fitting MCMC distributions. *arXiv*, 2017.

Vakili, S., Bromberg, M., Garcia, J., Shiu, D.-s., and Bernacchia, A. Uniform generalization bounds for overparameterized neural networks. *arXiv*, 2021.

Valle-Perez, G., Camargo, C. Q., and Louis, A. A. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *ICLR*, 2018.

Voevodin, V. and Kuznetsov, Y. Matrices and computations. *Nauka*, 1984.

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

Yaida, S. Non-Gaussian processes and neural networks at finite widths. In *Mathematical and Scientific Machine Learning*, 2020.

Yang, G. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv*, 2019.

Yang, G. Tensor programs III: Neural matrix laws. *arXiv*, 2020.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*, 2016.

Zavatone-Veth, J. and Pehlevan, C. Exact marginal prior distributions of finite Bayesian neural networks. *NeurIPS*, 2021.

A. Proofs

Here and in the main body of the paper, all statements of convergence in probability or almost sure (a.s.) convergence with $d_{\min} \to \infty$ necessitate construction of an underlying probability space where the random variables *for network of all widths* live. Since verifying that the assumptions we use are satisfied for a majority of NN architectures can be obtained by referring to the Tensor Program work of Yang, we adopt his probability space construction (e.g., Yang, 2020, appendix L).

Throughout we use the following definitions: $\lesssim$ is $\leq$ up to a universal positive (i.e., $a \lesssim b$ means $\exists c > 0$ s.t. $a \leq c\, b$), $d^{\leq L} \coloneqq d - d^L = \dim(\phi^{\leq L})$, $\langle r, s\rangle = r^\top s$ is the dot product, $\hat{K}_{\sigma^2} \coloneqq \sigma^2 I_n + \hat{K}$, and $K_{\sigma^2} \coloneqq \sigma^2 I_n + K$.

A.1. Theorem 2.1

Theorem 2.1. *Let $P_{\phi \mid \mathcal{D}}$ be the reparametrised posterior distribution defined by the density $p(\phi \mid \mathcal{D})$. Assume the Gaussian prior and likelihood with $\sigma > 0$, and that $\hat{K} \xrightarrow{\mathbb{P}} K$ and $\mathbb{E}[\hat{K}] \to K$ under the prior as $d_{\min} \to \infty$.*

Then $\mathrm{KL}(\mathcal{N}(0, I_d) \,\|\, P_{\phi \mid \mathcal{D}}) \to 0$ *as* $d_{\min} \to \infty$.

Proof of Theorem 2.1. By the conditional KL divergence identity

$$\mathrm{KL}(\mathcal{N}(0, I_d) \,\|\, P_{\phi \mid \mathcal{D}}) = \mathrm{KL}(\mathcal{N}(0, I_{d^{\leq L}}) \,\|\, P_{\phi^{\leq L} \mid \mathcal{D}}) + \mathbb{E}_{\phi^{\leq L} \sim \mathcal{N}(0, I_{d^{\leq L}})}[\mathrm{KL}(\mathcal{N}(0, I_{d^L}) \,\|\, P_{\phi^{L+1} \mid \phi^{\leq L}, \mathcal{D}})]\,,$$

where the latter term is zero by the argument right after Equation (3). Recalling $\phi^{\leq L} = \theta^{\leq L}$ and $p(\theta^{\leq L}) \sim \mathcal{N}(0, I_{d^{\leq L}})$

$$\mathrm{KL}(\mathcal{N}(0, I_{d^{\leq L}}) \,\|\, P_{\phi^{\leq L} \mid \mathcal{D}}) = \log Z + \tfrac{1}{2}\, \mathbb{E}_{\phi^{\leq L} \sim \mathcal{N}(0, I)} \left[\|y\|^2_{\hat{K}^{-1}_{\sigma^2}} + \log\det(\hat{K}_{\sigma^2}) \right],\,,$$

where we substituted Equation (5) renormalised by $Z \coloneqq \mathbb{E}_{\phi^{\leq L} \sim \mathcal{N}(0, I_{d^{\leq L}})} \exp\{-\frac{1}{2}[\|y\|^2_{\hat{K}^{-1}_{\sigma^2}} + \log\det(\hat{K}_{\sigma^2})]\}$. Note that

$$\hat{K} \mapsto \exp\{-\tfrac{1}{2}[\|y\|^2_{\hat{K}^{-1}_{\sigma^2}} + \log\det(\hat{K}_{\sigma^2})]\}\,,$$

is a bounded function of $\hat{K}$ by the positive semidefiniteness of $\Psi\Psi^\top$. It is also continuous in $\hat{K}$: for the determinant use, e.g., the Leibniz formula; the matrix inverse is continuous on the space of invertible matrices by continuity of the determinant and of $A \mapsto \mathrm{adj}\, A$ (sufficient here since $\hat{K}_{\sigma^2} = \sigma^2 I_n + \hat{K}$ where $\hat{K}$ is positive semidefinite). Since the measure w.r.t. which we integrate is the prior $\mathcal{N}(0, I_{d^{\leq L}})$, and $\hat{K} \to K$ in probability under the prior by assumption

$$\log Z \to -\tfrac{1}{2}[\|y\|^2_{K^{-1}_{\sigma^2}} + \log\det(K_{\sigma^2})] \quad \text{as} \quad d_{\min} \to \infty\,,$$

by the definition of convergence in distribution (implied by convergence in probability), and the continuity of the $\log$.

All that remains is thus to prove

$$\mathbb{E}_{\phi^{\leq L} \sim \mathcal{N}(0, I)} \left[\|y\|^2_{\hat{K}^{-1}_{\sigma^2}} + \log\det(\hat{K}_{\sigma^2}) \right] \to \|y\|^2_{K^{-1}_{\sigma^2}} + \log\det(K_{\sigma^2}) \quad \text{as} \quad d_{\min} \to \infty\,.$$

Note that the integrand is continuous in $\hat{K}$ by the above argument. Using the convergence of $\hat{K}$ to K in probability under the prior, we thus see that the integrand converges in probability to the above constant by the continuous mapping theorem.

Convergence of the expectation can then be ensured by establishing uniform integrability (UI), and invoking the Vitali's convergence theorem for finite measures (or theorem 3.5 in Billingsley, 1999). This is simple for the first term since

$$0 \leq \|y\|^2_{K^{-1}_{\sigma^2}} \leq \sigma^{-2} \|y\|_2^2 \quad \text{a.s.},$$

which trivially implies UI. For the log determinant, note $0 \leq \log\det(\hat{K}_{\sigma^2}) \leq \mathrm{Tr}(\hat{K}_{\sigma^2})$ (a.s.) by the arithmetic-geometric mean inequality. By Lemma A.1, UI integrability can thus be obtained by establishing UI of $\{\mathrm{Tr}(\hat{K})\}$ (indexed by width). Because $\mathrm{Tr}(\hat{K}) \geq 0$ (a.s.) and $\mathbb{E}[\mathrm{Tr}(\hat{K})] = \mathrm{Tr}(\mathbb{E}[\hat{K}]) \to \mathrm{Tr}(\mathbb{E}[K]) = \mathbb{E}[\mathrm{Tr}(K)]$ under the prior by assumption, the desired UI follows by theorem 3.6 in (Billingsley, 1999). The UI of the sum then follows by the triangle inequality. □

A.2. Proposition 2.2

Proposition 2.2. *Let $\phi \sim \mathcal{N}(0, I_d)$, $\theta \sim P_{\theta \mid \mathcal{D}}$, and denote the functions they parametrise by $f_{T(\phi)}$ and f_θ. Assume all nonlinearities are continuous, and each layer's outputs are jointly continuous in the layer parameters and inputs.*

Then $\mathrm{KL}(P_{f_{T(\phi)}} \,\|\, P_{f_\theta \mid \mathcal{D}}) \to 0$ *as* $d_{\min} \to \infty$*, where the* KL *is defined w.r.t. the usual product σ-algebra on $\mathbb{R}^{\mathcal{X}}$.*

Proof of Proposition 2.2. As mentioned in the main text, KL divergence between the pushforwards of two distributions through the *same* measurable mapping is never larger than that between the original distributions (e.g., Gray, 2011, corollary 5.2.2). It is thus sufficient to show that for any given fixed network architecture, the mapping from the parameters into the function space $\theta \in \mathbb{R}^d \mapsto f_\theta \in \mathbb{R}^{\mathcal{X}}$ is measurable with respect to the usual Borel product σ-algebras on $\mathbb{R}^d$ and $\mathbb{R}^{\mathcal{X}}$.

The product σ-algebra on $\mathbb{R}^{\mathcal{X}}$ is by definition generated by the product topology. The product topology is generated by a base composed of sets $A = \bigcap_{i=1}^k \pi_k^{-1}(A_i)$, where each $\pi_k \colon \mathbb{R}^{\mathcal{X}} \to \mathbb{R}$ is a coordinate projection associated with a point $x_k \in \mathcal{X}$, and each A_k is an open set in the output space $\mathcal{Y} \subseteq \mathbb{R}$. Because any finite intersection of open sets is open, it is sufficient to show that the mapping $\theta \mapsto f_\theta(x)$ is continuous for every $x \in \mathcal{X}$. Since the outputs of each layer are jointly continuous in the inputs and its own parameters by assumption, we can progress recursively from the input to the output, and conclude by observing that compositions of continuous functions are continuous. □

A.3. Proposition 2.3

Proposition 2.3. *Under the assumptions of Theorem 2.1*

$$\begin{aligned}&\mathrm{KL}(\,\mathcal{N}(0, I_d) \,\|\, P_{\theta \mid \mathcal{D}}\,)\\ &\to \tfrac{1}{2}[\sigma^{-2}\mathrm{Tr}(K) + \|y\|^2_{\sigma^{-2}I_n - K^{-1}_{\sigma^2}} - \log\det(\sigma^{-2}K_{\sigma^2})]\,,\end{aligned}$$

as $d_{\min} \to \infty$. The limit is equal to the KL divergence between the NNGP prior and posterior in function space, defined w.r.t. the usual product σ-algebra on $\mathbb{R}^{\mathcal{X}}$.

Proof of Proposition 2.3. By the conditional KL divergence identity

$$\mathrm{KL}(\mathcal{N}(0, I_d) \,\|\, P_{\theta \mid \mathcal{D}}) = \mathrm{KL}(\mathcal{N}(0, I_{d^{\leq L}}) \,\|\, P_{\theta^{\leq L} \mid \mathcal{D}}) + \mathbb{E}_{\theta^{\leq L} \sim \mathcal{N}(0, I_{d^{\leq L}})}[\mathrm{KL}(\mathcal{N}(0, I_{d^L}) \,\|\, P_{\theta^{L+1} \mid \theta^{\leq L}, \mathcal{D}})]\,.$$

The first term converges to zero (see the proof of Theorem 2.1). We have also already seen $P_{\theta^{L+1} \mid \theta^{\leq L}, \mathcal{D}} = \mathcal{N}(\mu, \Sigma)$ (see the discussion around Equation (3)). Using the formula for KL divergence between two multivariate normal distributions

$$\begin{aligned}\mathrm{KL}(\mathcal{N}(0, I_{d^L}) \,\|\, P_{\theta^{L+1} \mid \theta^{\leq L}, \mathcal{D}}) &\propto \mathrm{Tr}(\Sigma^{-1}) + \|\mu\|^{-1}_{\Sigma} - d^L + \log\det(\Sigma)\\ &= \sigma^{-2}\mathrm{Tr}(\hat{K}) - \sigma^{-2}[I_n - \sigma^{-2}\Psi(I_{d^L} + \sigma^{-2}\Psi^\top\Psi)^{-1}\Psi^\top - I_n] - \log\det(\sigma^{-2}\hat{K}_{\sigma^2})\\ &= \sigma^{-2}\mathrm{Tr}(\hat{K}) + \|y\|^2_{\sigma^{-2}I_n - \hat{K}^{-1}_{\sigma^2}} + n\log(\sigma^2) - \log\det(\hat{K}_{\sigma^2})\,,\end{aligned}$$

where we used the Woodbury identity for the last equality. Convergence of all the above terms and their expectations has already been established in the proof of Theorem 2.1. Reintroducing the dropped constant, we therefore have

$$\mathrm{KL}(\mathcal{N}(0, I_d) \,\|\, P_{\theta \mid \mathcal{D}}) \to \tfrac{1}{2}\big[\sigma^{-2}\mathrm{Tr}(K) + \|y\|^2_{\sigma^{-2}I_n - K^{-1}_{\sigma^2}} + n\log(\sigma^2) - \log\det(K_{\sigma^2})\big]\,.$$

To establish equivalence to the KL between the NNGP prior and posterior, it is sufficient to employ equation (12) from (Matthews et al., 2016) with, in their notation, $Q = P$ set to be the prior. This results in

$$\log Z + \tfrac{1}{2}\left[n\log(2\pi\sigma^2) + \sigma^{-2}\,\mathbb{E}_{f(X)\sim\mathcal{N}(0,K)}\big[\|y - f(X)\|_2^2\big]\right] = \log Z + \tfrac{1}{2}\left[n\log(2\pi\sigma^2) + \sigma^{-2}[\|y\|_2^2 + \mathrm{Tr}(K)]\right]\,,$$

where

$$\begin{aligned}\log Z &= \mathbb{E}_{f(X)\sim\mathcal{N}(0,K)}\big[(2\pi\sigma^2)^{-n/2}\exp\big\{-\tfrac{1}{2\sigma^2}\|y - f(X)\|_2^2\big\}\big]\\ &= -\tfrac{1}{2}\big[n\log(2\pi\sigma^2) + \log\det(K) + \log\det(K^{-1} + \sigma^{-2}I_n) + \|y\|^2_{K^{-1}_{\sigma^2}}\big]\\ &= -\tfrac{1}{2}\big[n\log(2\pi) + \log\det(K_{\sigma^2}) + \|y\|^2_{K^{-1}_{\sigma^2}}\big]\,,\end{aligned}$$

by basic determinant identities. Substituting $\log Z$ into the previous equation yields the result. □

A.4. Proposition 3.1

The two additional technical assumptions mentioned in the main text are:

1. The set of last layer preactivations f^L is exchangeable over the embedding (width) index, and converges in distribution to a well-defined limit (not necessarily a GP, even though that is the case in most scenarios). These facts are true for almost all architectures—(see, e.g., Matthews et al., 2018; Garriga-Alonso et al., 2019; Hron et al., 2020b; Yang, 2020).

2. Defining $\tilde{\Theta}_{11} \coloneqq \frac{\partial h_{:1}}{\partial \phi^{\leq L}} \frac{\partial h_{:1}^\top}{\partial \phi^{\leq L}} \in \mathbb{R}^{n \times n}$, we assume $\mathbb{E}[\|\tilde{\Theta}_{11}\|_2^2]$ converges as well. This will again be generally true by $\|\tilde{\Theta}_{11}\|_2^2 \leq \operatorname{Tr}(\tilde{\Theta})^2$ and theorem A.2 in (Yang, 2020).

Proposition 3.1. *Let $\Delta_\phi \coloneqq -\phi - \nabla_\phi \log p(\phi \,|\, \mathcal{D})$ where $-\phi$ is the gradient of the log density of $\mathcal{N}(0, I)$. Assume the conditions in Theorem 2.1, and continuous nonlinearities.*

Then $P_{\phi \,|\, \mathcal{D}}(\|\Delta_\phi\|_2 > \epsilon) \to 0$ as $d_{\min} \to \infty$, $\forall \epsilon > 0$.

Proof of Proposition 3.1. By $p(\phi^{L+1} \,|\, \phi^{\leq L}, \mathcal{D}) \sim \mathcal{N}(0, I_d^L)$ (see the discussion after Equation (3)), the last layer gradients always exactly agree with that of the standard normal, i.e., $\nabla_{\phi^{L+1}} \log e^{-\|\phi^{L+1}\|_2^2/2} = -\phi^{L+1}$. Hence Δ_ϕ depends only on $\phi^{\leq L}$. Substituting from Equation (5) (after application of the Weinstein–Aronszajn identity)

$$\|\Delta_\phi\|_2 \leq \|\nabla_{\phi^{\leq L}} \log\det(\hat{K}_{\sigma^2})\|_2 + \|\nabla_{\phi^{\leq L}} y^\top \hat{K}_{\sigma^2}^{-1} y\|_2 \,.$$

We will use the shortcut ∂_i to denote the derivative w.r.t. $\phi_i^{\leq L}$, so that $\|\nabla_{\phi^{\leq L}} \cdots\|_2 = \sum_{i=1}^{d^{\leq L}} (\partial_i \cdots)^2$. Applied to the individual summands, we have $\partial_i \log\det(\hat{K}_{\sigma^2}) = \operatorname{Tr}(\hat{K}_{\sigma^2}^{-1}(\partial_i \hat{K})) = 1_n^\top \hat{K}_{\sigma^2}^{-1}(\partial_i \hat{K}) 1_n$ where $1_n \in \mathbb{R}^n$ is a vector of all ones. Similarly, $\partial_i \, y^\top \hat{K}_{\sigma^2}^{-1} y = -y^\top \hat{K}_{\sigma^2}^{-1}(\partial_i \hat{K}) \hat{K}_{\sigma^2}^{-1} y$. Hence the summands in both gradients have the form $(r^\top(\partial_i \hat{K}) s)^2$ for some possibly random vectors $r, s \in \mathbb{R}^n$ whose distribution may depend on the width. Substituting the definition of Ψ

$$\partial_i \hat{K}_{ab} \propto \frac{1}{d^L} \sum_{j=1}^{d^L} (\partial_i h_{aj}) h_{bj} + h_{aj} (\partial_i h_{bj}) \,,$$

where we used the abbreviation $h_{aj} \coloneqq h_j^L(x_a)$. Therefore $r^\top(\partial_i \hat{K}) s \propto \frac{1}{d^L} \sum_j \langle r, \partial_i h_{:j} \rangle \langle s, h_{:j} \rangle + \langle s, \partial_i h_{:j} \rangle \langle r, h_{:j} \rangle$, so

$$\begin{aligned}
\sum_{i=1}^{d^{\leq L}} (r^\top(\partial_i \hat{K}) s)^2 &\propto \frac{1}{(d^L)^2} \sum_{i=1}^{d^{\leq L}} \sum_{j,k=1}^{d^L} [r^\top (\partial_i h_{:j})(\partial_i h_{:k})^\top r][s^\top h_{:j} h_{:k}^\top s] + [r^\top (\partial_i h_{:j})(\partial_i h_{:k})^\top s][s^\top h_{:j} h_{:k}^\top r] + \dots \\
&= \frac{1}{(d^L)^2} \sum_{j,k=1}^{d^L} [s^\top h_{:j} h_{:k}^\top s] \big[r^\top \tfrac{\partial h_{:j}}{\partial \phi^{\leq L}} \tfrac{\partial h_{:k}^\top}{\partial \phi^{\leq L}} r\big] + [s^\top h_{:j} h_{:k}^\top r] \big[r^\top \tfrac{\partial h_{:j}}{\partial \phi^{\leq L}} \tfrac{\partial h_{:k}^\top}{\partial \phi^{\leq L}} s\big] + \dots \,,
\end{aligned}$$

where we have expanded the square (the last two terms are omitted as signified by the ellipsis). Defining $\tilde{\Theta}_{jk} \coloneqq \frac{\partial h_{:j}}{\partial \phi^{\leq L}} \frac{\partial h_{:k}^\top}{\partial \phi^{\leq L}}$

$$\sum_{i=1}^{d^{\leq L}} (r^\top(\partial_i \hat{K}) s)^2 \propto \frac{1}{(d^L)^2} \sum_{j,k=1}^{d^L} (\langle s, h_{:j} \rangle \, r + \langle r, h_{:j} \rangle \, s)^\top \tilde{\Theta}_{jk} (\langle s, h_{:k} \rangle \, r + \langle r, h_{:k} \rangle \, s)$$

By Theorem 2.1, we can now integrate w.r.t. to the prior $\mathcal{N}(0, I_{d^{\leq L}})$ instead of the posterior $P_{\phi^{\leq L} \,|\, \mathcal{D}}$, because by Pinsker's inequality $\operatorname{TV}(\mathcal{N}(0, I_{d^{\leq L}}), P_{\phi^{\leq L} \,|\, \mathcal{D}}) \lesssim [\operatorname{KL}(\mathcal{N}(0, I_{d^{\leq L}}) \,\|\, P_{\phi^{\leq L} \,|\, \mathcal{D}})]^{1/2}$ where the right-hand side goes to zero as widths go to infinity (see the proof of Theorem 2.1 for details). Using $z_j \coloneqq \langle s, h_{:j} \rangle \, r + \langle r, h_{:j} \rangle \, s \in \mathbb{R}^n$

$$\sum_{i=1}^{d^{\leq L}} \mathbb{E}[(r^\top(\partial_i \hat{K}) s)^2] = \tfrac{1}{d^L} \, \mathbb{E}[z_1^\top \tilde{\Theta}_{11} z_1] + (1 - \tfrac{1}{d^L}) \, \mathbb{E}[z_1^\top \tilde{\Theta}_{12} z_2] \tag{8}$$

by the assumed exchangeability of the readout units. By the linear readout assumption, $\tilde{\Theta}_{ij}$ is just a scaled version of $\hat{\Theta} - \hat{K}$, where $\hat{\Theta} \coloneqq (\partial_\phi f^{L+1})(\partial_\phi f^{L+1})^\top$ is the *empirical NTK*, with the output layer 'integrated out'. It thus converges in

probability under the prior (and thus the posterior) by the above argument and (Yang, 2020, theorem 2.10), with $\tilde{\Theta}_{ij}$ going to a constant $\tilde{\Theta} \in \mathbb{R}^{n\times n}$ if $i = j$, and to zero otherwise. Since $\hat{K}_{\sigma^2}^{-1}$ also converges to a constant in probability (see the proof of Theorem 2.1), the vectors r and s do so as well. Since the vectors h are just ψ applied pointwise to the previous layer outputs, which converge to a GP by assumption, they will converge in distribution to the corresponding $\psi_\#$ pushforward by the continuity of ψ. Their limit will be denoted simply by z. By the Slutsky's lemma, the integrand $z_1^\top \tilde{\Theta}_{11} z_1$ thus converges in distribution to $z^\top \tilde{\Theta} z$, and $z_1^\top \tilde{\Theta}_{12} z_2$ to zero.

Since $|z_1^\top \tilde{\Theta}_{12} z_2| \lesssim z_1^\top \tilde{\Theta}_{11} z_1 + z_2^\top \tilde{\Theta}_{11} z_2$ by the outer product definition of $\tilde{\Theta}_{11}$, and the arithmetic-geometric mean inequality, it will be sufficient to establish uniform integrability (UI) of $z_1^\top \tilde{\Theta}_{11} z_1$ to show the expectations converge (Billingsley, 1999, theorem 3.5). This can be obtained by observing $z_1^\top \tilde{\Theta}_{11} z_1 \leq \|\tilde{\Theta}_{11}\|_2 \|z\|_2^2 \leq \|\tilde{\Theta}_{11}\|_2^2 + \|z\|_2^4 \leq \mathrm{Tr}(\tilde{\Theta}_{11})^2 + \|z\|_2^4$ which are both non-negative random variables, which converge in probability (resp. in distribution) by continuity of the trace and the the square function. Their expectations then converge by theorem A.2 in (Yang, 2020). The collection of $\{z_1^\top \tilde{\Theta}_{11} z_1\}$ indexed by width is thus UI by Lemma A.1. Hence the expectations on the right hand side of Equation (8) converge, implying their sequence is bounded, and thus

$$\mathbb{E}[\|\Delta_\phi\|_2] \leq \sqrt{\mathbb{E}[\|\Delta_\phi\|_2^2]} \lesssim \tfrac{1}{\sqrt{d^L}} + o(1)\,,$$

Convergence in posterior probability follows by the above Pinsker's inequality argument, and the Markov's inequality. □

Auxiliary results

Lemma A.1. *If $\mathcal{C}$ and $\mathcal{C}'$ are two collections of random variables such that for every $X \in \mathcal{C}$ there exists $Y \in \mathcal{C}'$ such that $|X| \leq |Y|$ a.s., and $\mathcal{C}'$ is uniformly integrable (UI), then $\mathcal{C}$ is uniformly integrable.*

Proof of Lemma A.1. Clearly $\sup_{X\in\mathcal{C}} \mathbb{E}\,|X| \leq \sup_{Y\in\mathcal{C}'} \mathbb{E}\,|Y| < \infty$ by UI. Furthermore, for arbitrary $\epsilon > 0$, $\exists \delta > 0$ s.t.

$$\sup_{\mathbb{P}(A)\leq\delta} \sup_{X\in\mathcal{C}} \mathbb{E}\,|X|\mathbb{1}_A \leq \sup_{\mathbb{P}(A)\leq\delta} \sup_{Y\in\mathcal{C}'} \mathbb{E}\,|Y|\mathbb{1}_A \leq \epsilon$$

again by the uniform integrability of $\mathcal{C}'$. Taken together, the two facts imply $\mathcal{C}$ is also uniformly integrable. □

B. Feature-space vs. data-space reparametrisation

The **feature-space version** of our algorithm—presented in the main body and utilised in the experiments—scales cubically in the top layer width d^L rather than the number of datapoints n (cubic scaling in n is the main bottleneck of exact GP inference). This is preferable in most practical circumstances since d^L is hardly ever significantly larger than n, making the additional computational overhead negligible Figure 6. Here we present a **data-space version** which scales cubically in n as in GPs, but only quadratically in d^L. This may be useful, e.g., when investigating properties of very wide BNNs.

The data-space version can be obtained using ideas related to the 'kernel trick'. Starting with $T(\phi) = \mu + \Sigma^{1/2}\phi$, we have $\mu = (\sigma^2 I_{d^L} + \Psi^\top\Psi)^{-1}\Psi^\top y = \Psi^\top(\sigma^2 I_n + \Psi\Psi^\top)^{-1} y$. For the covariance

$$\Sigma = (I_{d^L} + \sigma^{-2}\Psi^\top\Psi)^{-1} = I_{d^L} - \Psi^\top(\sigma^2 I_n + \Psi\Psi^\top)^{-1}\Psi\,,$$

by the Woodbury identity. The reparametrisation requires the square root $\Sigma^{1/2}$ though, for which we can use a **Woodbury-like identity for the matrix square root**. To our knowledge, this identity is *new* and may thus be of independent interest:

$$(I_p + A^\top A)^{1/2} = I_p + A^\top \left[I_m + (I_m + AA^\top)^{1/2}\right]^{-1} A\,, \tag{9}$$

for any $m, p \in \mathbb{N}$ and $A \in \mathbb{R}^{m\times p}$. Clearly this reduces the $\mathcal{O}(p^3)$ complexity of the square root on the l.h.s. to the $\mathcal{O}(m^3 + m^3) = \mathcal{O}(m^3)$ complexity of the square root and matrix inverse on the r.h.s. To see the identity holds, use SVD decomposition $A = USV^\top$ to rewrite the l.h.s. as $V(I_p + S^\top S)^{1/2}V^\top$, and the r.h.s. as

$$I_p + A^\top \left[I_m + (I_m + AA^\top)^{1/2}\right]^{-1} A = V\left\{I_p + S^\top \left[I_m + (I_m + SS^\top)^{1/2}\right]^{-1} S\right\} V^\top\,,$$

and observe that $\sqrt{1+x^2} = 1 + \frac{x^2}{1+\sqrt{1+x^2}}$ for any $x \in \mathbb{R}$. Since we need the inverse, $(I_{d^L} + \sigma^{-2}\Psi^\top\Psi)^{-1/2} = \Sigma^{1/2}$, we can combine Equation (9) with the *usual* Woodbury identity for matrix inverse to obtain

$$\Sigma^{1/2} = I_{d^L} - \Psi^\top \left[\sqrt{\sigma^2 I_n + \Psi\Psi^\top}\left(\sigma I_n + \sqrt{\sigma^2 I_n + \Psi\Psi^\top}\right)\right]^{-1} \Psi\,.$$

We can then perform the *eigendecomposition* $\sigma^2 I_n + \Psi\Psi^\top = Q\Lambda Q^\top$ which has $\mathcal{O}(n^3)$ complexity (same as the Cholesky decomposition), and thus allows efficient computation of *both* $\mu = \Psi^\top Q\Lambda^{-1}Q^\top y$, and

$$\Sigma^{1/2}\phi = \phi - \Psi^\top Q\Lambda^{-1/2}(\sigma I_n + \Lambda^{1/2})^{-1}Q^\top\Psi\phi\,,$$

since Λ is a diagonal matrix. Similar to Section 3.2, we can *reuse* the decomposition result to obtain the density value essentially for free using $p(\phi \mid \mathcal{D}) = p(\theta \mid \mathcal{D})\left|\det \partial_\phi\theta\right|$, and Equation (4) together with the Weinstein–Aronszajn identity

$$\left|\det \partial_\phi\theta\right| = \sqrt{\det(\Sigma)} = \sqrt{\det(I_{d^L} + \sigma^{-2}\Psi^\top\Psi)} \propto \sqrt{\det(\sigma^2 I_n + \Psi\Psi^\top)} = \prod_{i=1}^{n}\sqrt{\Lambda_{ii}}\,.$$

C. Extension to multi-dimensional outputs

In the main text, we assumed the output dimension $d^{L+1} \in \mathbb{N}$ is equal to one for simplicity of exposition. The extension to $d^{L+1} > 1$ is simple. In particular, we take the factorised Gaussian likelihood (see Appendix F for the categorical likelihood)

$$\log p(y \mid X, \theta) = -\frac{nd^{L+1}}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{d^{L+1}}(y_{ij} - f_\theta(x_i)_j)^2\,,$$

where $f_\theta(x_i) \in \mathbb{R}^{d^{L+1}}$. Each output dimension is associated with an independent readout weight $W_{:j}^{L+1} \in \mathbb{R}^{d^L}$ s.t. $f_\theta(x)_j = \sigma_W^{L+1}h(x)^L W_{:j}^{L+1}/\sqrt{d^L}$. To compute $\theta = T(\phi)$, the only modification occurs in the last layer where the mean $\mu_{:j} = \Psi^\top(\sigma^2 I_n + \Psi\Psi^\top)^{-1}y_{:j}$ is different for each $W_{:j}^{L+1}$, with $y_{:j} \in \mathbb{R}^n$ the j^{th} column of the targets $y \in \mathbb{R}^{n\times d^{L+1}}$. Importantly, Ψ is the same for all j, meaning a single Cholesky decomposition is sufficient. Modern solvers allow for multi-column r.h.s. meaning that we can solve for the reparametrised value of the output weights at once, limiting the computational overhead.

The only other change is in the determinant $\det(\partial_\phi\theta)$. Because each $W_{:j}^{L+1}$ only depends on $\phi^{\leq L} = \theta^{\leq L}$, the determinant still has the upper-triangular structure analogous to Equation (4), implying

$$\log\left|\det(\partial_\phi\theta)\right| = \text{constant} + d^{L+1}\log\det(\sigma^2 I_n + \Psi\Psi^\top)\,.$$

We can thus again re-use the result of the Cholesky decomposition to obtain the determinant value essentially for free.

D. Repriorisation induced LMC speed-up in linear models

While the linear case $f_\theta(x) = \langle x, \theta\rangle$ is of course not equivalent to deep BNNs, we use it here to provide a rough intuition for the source and magnitude of the improvement that could be encountered when running LMC with $p(\phi \mid \mathcal{D})$ instead of $p(\theta \mid \mathcal{D})$. To sample from a generic density $p(z)$, LMC (and HMC) typically uses the Leapfrog integrator to simulate the Hamiltonian dynamics of a system with *potential energy* $U(z) = -\log p(z)$ and *kinetic energy* $K(m) = \frac{1}{2}\|m\|^2_{M^{-1}}$, where $m \in \mathbb{R}^d$ is an auxiliary momentum variable, and $M \in \mathbb{R}^{d\times d}$ a positive definite mass matrix (Brooks et al., 2011).

The integrator simulates movement of a particle with initial state $(z_0, m_0) \in \mathbb{R}^d \times \mathbb{R}^d$ along a trajectory of constant energy $H(z, m) = U(z) + K(m)$ in the 'counter-clockwise direction'

$$\begin{pmatrix}\dot{m}_t\\ \dot{z}_t\end{pmatrix} = \begin{pmatrix}O & -I_d\\ I_d & 0\end{pmatrix}\begin{pmatrix}\partial_{m_t}H(z_t, m_t)\\ \partial_{z_t}H(z_t, m_t)\end{pmatrix} = \begin{pmatrix}-\nabla U(z_t)\\ \nabla K(m_t)\end{pmatrix}\,.$$

A nice property of the linear case is that we can easily solve this matrix ODE, and thus determine the error accrued by the integrator for any given $t > 0$.[6] Without reparametrisation ($z = \theta$), $\nabla U(\theta) = \theta + \sigma^{-2}X^\top(X\theta - y) = C\theta - b_\mathcal{D}$, where

[6] Whether this can be exploited to design a more efficient integrator for wide BNNs is an interesting question for future research.

$C = I + \sigma^{-2} X^\top X = \Sigma^{-1}$ and $b_\mathcal{D} = \sigma^{-2} X^\top y = \Sigma^{-1}\mu$. This translates into the matrix ODE

$$\begin{pmatrix} \dot{m}_t \\ \dot{\theta}_t \end{pmatrix} = \underbrace{\begin{pmatrix} O & -C \\ M^{-1} & 0 \end{pmatrix}}_{=A} \begin{pmatrix} m_t \\ \theta_t \end{pmatrix} + \underbrace{\begin{pmatrix} b_\mathcal{D} \\ 0 \end{pmatrix}}_{=Ab},$$

which has the usual solution

$$\begin{pmatrix} m_t \\ \theta_t \end{pmatrix} = (I - e^{tA})b + e^{tA} \begin{pmatrix} m_0 \\ \phi_0 \end{pmatrix}.$$

Accuracy of the integrator is thus determined by how closely it can approximate the matrix exponential e^{tA}. The value of this exponential is known; to simplify, we take $M = I$ as used in our experiments

$$\begin{aligned} e^{tA} &= \sum_{k=0}^{\infty} \frac{t^{2k}}{(2k)!} \begin{pmatrix} (-C)^k & 0 \\ 0 & (-C)^k \end{pmatrix} + \sum_{k=0}^{\infty} \frac{t^{2k+1}}{(2k+1)!} \begin{pmatrix} 0 & (-C)^{k+1} \\ (-C)^k & 0 \end{pmatrix} \\ &= \begin{pmatrix} \cos(t\sqrt{C}) & -\sqrt{C}\sin(t\sqrt{C}) \\ (\sqrt{C})^{-1}\sin(t\sqrt{C}) & \cos(t\sqrt{C}) \end{pmatrix}. \end{aligned}$$

Note that C is the inverse posterior covariance, and the above equations describe a 'rotation' along an ellipsoid defined by the eigendecomposition of C, as expected. Most importantly, the integrator error will be driven by $t\sqrt{C}$, where in particular the LMC *stepsize has to be inversely proportional to* $\|\sqrt{C}\|_2$ if a local approximation of $\sin$ and $\cos$ is to be accurate.

An analogous argument holds for the reparametrised version ($z = \phi$), where $\nabla U(\phi) = \frac{\partial\theta}{\partial\phi}^\top [\Sigma^{-1}(\sigma^{-2}\Sigma X^\top y + \sqrt{\Sigma}\,\phi) - \sigma^{-2} X^\top y] = \phi$ by $\Sigma = C^{-1}$ and $\frac{\partial\theta}{\partial\phi} = \sqrt{\Sigma}$. The stepsize can therefore be (inversely) proportional to $\|I_d\|_2 = 1$, disregards of both n and d. In contrast, without the reparametrisation, the stepsize must be *inversely proportional to* (see above)

$$\|\sqrt{C}\|_2 = \sqrt{\|C\|_2} = \sqrt{1 + \tfrac{\|X^\top X\|_2}{\sigma^2}} \sim \frac{\sqrt{n}}{\sigma},$$

where we used $\|X^\top X\|_2 = n\,\|\frac{1}{n}\sum_i x_i x_i^\top\|_2$ and assumed $\|\frac{1}{n}\sum_i x_i x_i^\top\|_2 = \mathcal{O}(1)$. This assumption will be true for most sufficiently well-behaved distributions over x_i by the law of large numbers (if $\mathbb{E}[xx^\top]$ exists), and the continuity of the operator norm (see chapter 6 in Wainwright, 2019, for a more detailed discussion).

In summary: without the reparametrisation, the maximum stepsize scales as $\frac{\sigma}{\sqrt{n}}$, whereas it is independent of n and σ in the reparametrised case. The higher the n and/or lower the σ, the bigger the advantage of our reparametrisation. In the deep BNN case, we replace X with Ψ, and look at the eigenspectrum of the NNGP kernel divided by n (which is related to its kernel integral operator in wide enough BNNs). While only an approximation in deep BNNs, we note that a constraint of the above form must be satisfied by the sampler in order to successfully sample the readout layer (with the asymptotic NNGP replaced by the empirical one), which is why we intuitively expect the stepsize to scale *at least* as $\frac{\sigma}{\sqrt{n}}$. This was observed in our experiments, where we indeed applied this stepsize scaling for the final hyperparameter sweep used in Figures 7 and 8.

E. Alternative reparametrisations

E.1. Reparametrising other layers than readout

The reparametrisation $T\colon \phi \mapsto \theta$ we define in Equation (3) ensures that θ^{L+1} follows its true conditional posterior when $\phi^{L+1} \sim \mathcal{N}(0, I_{d^L})$, which we have seen is the case under $p(\phi \,|\, \mathcal{D})$. One may wonder whether there exists an alternative reparametrisation(s) $R\colon \phi \mapsto \theta$ which still ensures the KL divergence between $p(R(\phi) \,|\, \mathcal{D})|\det \partial_\phi R(\phi)|$ and $\mathcal{N}(0, I_d)$ goes to zero as $d_{\min} \to \infty$, but maps some other layer than the readout to its conditional posterior. In other words, is the readout 'special', or is mapping of *any* one layer to its conditional sufficient to ensure reversion to the prior in the wide limit?

For an example of a case where reparametrisation of a layer other than the readout is sufficient, consider the *linear* 1-hidden layer FCN with a *single* input and output dimension, $f(x) = (x \cdot u^\top)v/\sqrt{d^1}$, with $u, v \in \mathbb{R}^{d^1}$ respectively the input and readout weights (w.l.o.g. $\sigma_W = 1$). By symmetry, $f(x) = (x \cdot v^\top)u/\sqrt{d^1}$, implying we can apply the T from Equation (3) to the *input* weights u, and use Theorem 2.1 (mutatis mutandis) to obtain the desired KL-convergence.

However, this argument no longer holds when we introduce a nonlinearity, as the symmetry between u and v is broken. Moreover, we now show that the KL-convergence to the prior does not happen if we consider the only available alternative R, i.e., mapping u to its conditional posterior given the readout v and $\mathcal{D}$,[7] and leaving the *readout* v without reparametrisation.

To start, observe that the conditional distribution of v given u and $\mathcal{D}$ remains $\mathcal{N}(\mu, \Sigma)$ with $\mu = \Psi^\top \hat{K}_{\sigma^2}^{-1} y$ (kernel trick), and $\Sigma = I - \Psi^\top \hat{K}_{\sigma^2}^{-1} \Psi$ (Woodbury identity). For simplicity, we prove that KL-convergence to prior cannot happen when the nonlinearity ψ is *bounded*. Using a proof-by-contradiction, assume the KL *does* converge to zero. By the chain rule for KL divergence, this implies the KL between the *reparametrised marginal posterior* of the readout v given $\mathcal{D}$ converges in KL to the prior over v, i.e., $\mathcal{N}(0, I_{d^1})$. By the assumed boundedness of ψ and the Pinsker's inequality, this in turn implies

$$\mathbb{E}[v_{:1} \,|\, \mathcal{D}] = \mathbb{E}[\Psi_{:1}^\top \hat{K}_{\sigma^2}^{-1} y \,|\, \mathcal{D}] \to \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, I_{d^0})}[\psi(X\varepsilon)]^\top K_{\sigma^2}^{-1} y \,, \quad \text{as } d_{\min} \to \infty \,.$$

However, the Pinsker's inequality and boundedness of ψ also imply that $\mathbb{E}[v_{:1} \,|\, \mathcal{D}] \to \mathbb{E}[v_{:1}] = 0$ (prior mean), which is a contradiction unless $\mathbb{E}_{\varepsilon \sim \mathcal{N}(0, I_{d^0})}[\psi(X\varepsilon)] = 0$ (or $y = 0$). This is not true, e.g., for the popular sigmoid non-linearity.

In summary, the answer to our question from the first paragraph is 'it depends'. There are some linear networks in which symmetry enables reparametrising layers other than the readout and still obtaining the KL-convergence. However, introduction of non-linearities breaks this symmetry, which in turn prevents reversion to the prior. Reparametrisation of the readout is thus indeed 'special', as it is the only layer which ensures convergence in *all* cases (modulo the assumptions of Theorem 2.1).

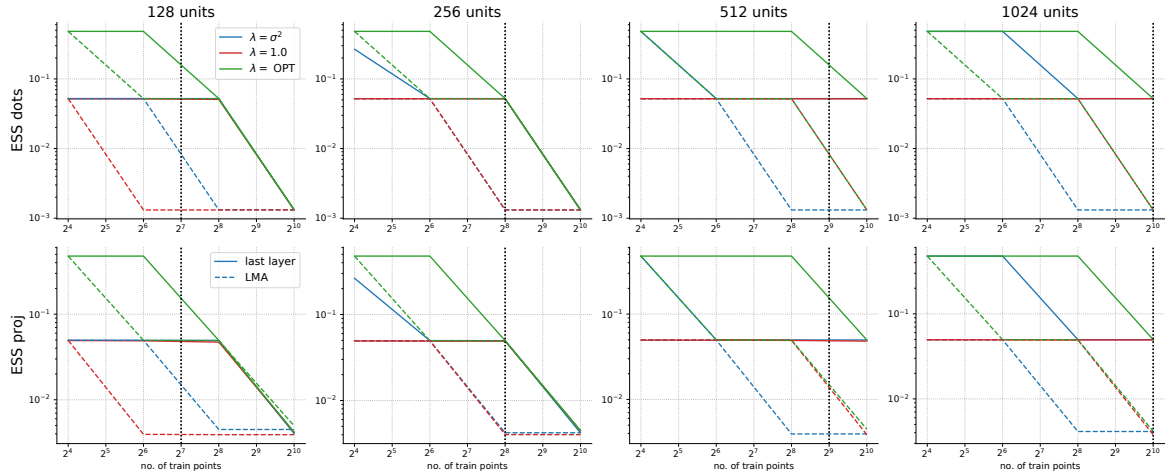

Figure 10. **Comparing NNGP T and NTK R** in terms of ESS per-step in the θ space for subsets of `cifar-10`. `ESS proj` is defined the same as in Section 4 (showing only the mean over projection dimensions). `ESS dots` is the ESS of the squared norm statistic $\|\theta\|_2^2$. Since these are older experiments, T is described as 'last layer' (solid lines), and R as 'LMA' (dashed lines). Colours distinguish different values of the regulariser λ. The $\lambda = \text{OPT}$ corresponds to the maximum ESS per-step values optimised over $\log \lambda = -4, -3, \ldots, 1$ for each dataset size separately. While the NTK reparametrisation is sometimes competitive, the NNGP one is clearly preferable.

E.2. NTK reparametrisation

Another alternative is to *linearise* f, i.e., to use the network Jacobian $J \coloneqq \partial_\phi f_\phi(X)$ to define the reparametrisation

$$\theta = R(\phi) = \phi + J^\top \hat{\Theta}_\lambda^{-1}(y - f_\phi(X)) \,, \quad \text{with } \hat{\Theta}_\lambda \coloneqq \lambda I_d + \hat{\Theta} = \lambda I_d + JJ^\top \,, \text{ for some } \lambda \geq 0 \,. \tag{10}$$

This reparametrisation is inspired by the NTK theory where optimising the NN linearised around its (random) initial point by gradient descent yields the same large width behaviour as optimising the original NN (Lee et al., 2019). *Informally*, if $\phi \sim \mathcal{N}(0, I)$ (the initialisation distribution in the NTK world), and the NN is wide enough

$$f_{R(\phi)}(x) \approx f_\phi(x) + J_x(R(\phi) - \phi) = f_\phi(x) + J_x J^\top \hat{\Theta}_\lambda^{-1}(y - f_\phi(X)) \,,$$

where $J_x \coloneqq \partial_\phi f_\phi(x)$. By the NNGP theory, $f_\phi \sim \text{GP}(0, k)$ is the NNGP limit; by the NTK theory, $\hat{\Theta}_{xX} \coloneqq J_x J^\top \to \Theta_{xX} \in \mathbb{R}^{1 \times n}$ and $\hat{\Theta} \to \Theta \in \mathbb{R}^{n \times n}$ in probability as $d_{\min} \to \infty$, where Θ is the NTK (see, e.g., Yang, 2020, for an

[7] A deterministic map from prior to posterior will exist because we are in $\mathbb{R}^{d^1}$, and both prior and posterior are continuous.

overview). If we approximate by substituting the limits, $f_{R(\phi)} \approx f_\phi + \Theta_{\cdot X}\Theta_\lambda^{-1}(y - f_\phi(X))$, we notice this is a fixed *linear* transformation of the NNGP, and thus also a GP. In fact, it is the NTK-GP when $\lambda = 0$ (Lee et al., 2019, corollary 1), i.e., the distributional limit of NNs optimised by gradient descent, where the randomness comes from the initialisation $\mathcal{N}(0, I)$.

The above sketch shows that $\theta = R(\phi)$ provides parameter-space samples which converge to the NTK-GP limit when $\phi \sim \mathcal{N}(0, I)$, forming an *informal* NTK counterpart to Proposition 2.2. An alternative is to interpret Equation (10) as a *single* step of the Levenberg–Marquardt algorithm (LMA; Levenberg, 1944; Marquardt, 1963). This perspective inspired us to also experiment with *multi*-step LMA update (i.e., iterative application of R from Equation (10)) so as to adjust for potential inaccuracy of the linearisation; we also experimented with just using multiple steps of *gradient descent* (GD), i.e., iterative application of $\phi \mapsto \phi + \alpha J^\top(y - f_\phi(X))$ with a fixed α. Neither the GD update nor the iterative application worked significantly better than Equation (10) in our experiments, which is why we abandon their further discussion here.

In the remainder of this section, we discuss the technical details of the NTK reparametrisation (Equation (10)) implementation. Figure 10 shows a comparison of this R with the 'NNGP reparametrisation' T we presented in the main paper (Equation (3)). While R is sometimes competitive with T, T is clearly better overall. This is likely because R yields samples from the NTK-GP, which may induce a complicated $p(\phi \mid \mathcal{D})$ when the NTK-GP significantly differs from the NNGP (i.e., the correct posterior, Hron et al., 2020a). Since T is also orders of magnitude more computationally efficient (scales with the top layer embedding size d^L instead of the full parameter space dimension d), and we have rigorous understanding of its behaviour (Section 2), we do not recommend using the NTK reparametrisation. The motivation for describing it here is to inform any potential future research, documenting what we tried, and contrasting it with the NNGP reparametrisation.

E.2.1. Computing $R(\phi)$

Denoting the residuals by $r \coloneqq y - f_\phi(X)$, R becomes $\phi \mapsto \phi + J^\top \hat{\Theta}_\lambda^{-1} r$. Because computing r costs about the same as the forward pass, and multiplication by $J^\top$ the same as the backward pass (vector-Jacobian product), the only operations not common in standard NN training are the computation of $\hat{\Theta}_\lambda$, and the subsequent application of the linear solver $r \mapsto \hat{\Theta}_\lambda^{-1} r$.

Fortunately, the matrix $\hat{\Theta} = JJ^\top$, known as the *empirical NTK*, has recently become a subject of significant interest. We can thus use the *Neural Tangents* library (Novak et al., 2020) which provides a very efficient way of computing $\hat{\Theta}$, and related matrix-vector products (Novak et al., 2022). As for the $r \mapsto \hat{\Theta}_\lambda^{-1} r$, we tried several implicit and explicit solvers, approximate and exact, but the fastest and most numerically stable was the explicit Cholesky solver. We recommend using at least `float32` precision (or its emulation if on TPU) to prevent significant deterioration of the acceptance probability.

Another possible approximation is subsetting the data. Specifically, we can replace the r, $J^\top$, and $JJ^\top$ terms in $\theta = \phi + J^\top(\sigma^2 I_n + JJ^\top)^{-1} r$ by their equivalent evaluated on a fixed data subset. This may be especially tempting when computational and memory bottlenecks are an issue. One nevertheless has to be careful about designing any approximations to R as—especially in the data subsetting case—these may inadvertently cause the resulting $p(\phi \mid \mathcal{D})$ to be as constrained as $p(\theta \mid \mathcal{D})$, and thus erase any possible LMC speed up. For example, in the linear case, if we subset the data and some of the omitted inputs is not close to the linear span of the included ones, there will be a posterior covariance eigenvalue as low as $\propto \frac{\sigma^2}{n}$ (instead of $\propto 1$), which would necessitate scaling the LMC stepsize by $\frac{\sigma}{\sqrt{n}}$ (cf. Appendix D).

While we aim to approximate the R (Equation (10)), any approximation of R *itself* is a valid reparametrisation (if bijective and differentiable), as long as the gradients and $\left|\det \partial_\phi \theta\right|$ are both computed for the *approximated* instead of the true R.

E.2.2. Approximating $\det \partial_\phi R(\phi)$

Unlike above, the approximations her *can* affect the stationary distribution of the Markov chain or even prevent its existence. Hence the obtained samples may not be drawn from the true posterior even in the infinite *time* limit.

Our log determinant approximation combines the Taylor expansion of $z \mapsto \log(1 + z)$ around $z = 0$, and the Hutchinson trick for trace estimation. In particular, for a real *symmetric* matrix A with $\|A\|_2 < 1$

$$\log|\det(I + A)| = -\sum_{j=1}^{\infty} \frac{(-1)^j}{j} \mathrm{Tr}(A^j) \approx -\frac{1}{S} \sum_{i=1}^{S} \sum_{j=1}^{M} \frac{(-1)^j}{j} s_i^\top A^j s_i \,, \tag{11}$$

where s_i are i.i.d. zero mean vectors with $E[s_i s_i^\top] = I$. We use vectors with i.i.d. Rademacher entries throughout.[8]

[8] We tried several alternatives to including Chebyshev and learned polynomials, but did not observe a significant difference.

Unfortunately, substituting $A = \frac{\partial\theta}{\partial\phi} - I$ into Equation (11) would violate the symmetry requirement since

$$\frac{\partial\theta}{\partial\phi} = (I - J^\top \hat{\Theta}_\lambda^{-1} J)(I + H_{\tilde{r}}) - J^\top \hat{\Theta}_\lambda^{-1} \begin{pmatrix} - & H_1 J^\top \tilde{r} & - \\ & \vdots & \\ - & H_n J^\top \tilde{r} & - \end{pmatrix}, \quad \text{with } H_{\tilde{r}} \coloneqq \sum_{i=1}^n \tilde{r}_i H_i \,, \tag{12}$$

where $\tilde{r} \coloneqq \hat{\Theta}_\lambda^{-1} r = \hat{\Theta}_\lambda^{-1}(y - f_\phi(X))$ is the 'preconditioned residual', and $H_i \coloneqq \frac{\partial^2 f_\phi(x_i)}{\partial\phi^2}$ the Hessian at x_i. A straightforward solution would be to use the identity $|\det(A)| = \sqrt{\det(AA^\top)}$. The downside is doubling of the computational time. We thus instead approximate by

$$\log\left|\det\left(\tfrac{\partial\theta}{\partial\phi}\right)\right| \approx \text{constant} + \log\left|\det(I_d + H_{\tilde{r}})\right| , \tag{13}$$

Roughly speaking, this approximation exploits the fact that the column spaces of $I_d - J^\top \hat{\Theta}_\lambda^{-1} J$ and $J^\top$ in Equation (12) are close to orthogonal for small λ relative to $\|\hat{\Theta}\|_2$. (Careful when tuning λ!) Rotating the space and using the block-matrix determinant formula allows us to decompose the log determinant into an asymptotically *constant* component aligned with the column space of $J^\top$, and the *stochastic* component $\log|\det(I_d + H_{\tilde{r}})|$.

Combining Equations (11) and (13), we have $A = H_{\tilde{r}}$ which is a symmetric matrix as desired. Furthermore, the Hutchinson estimator only requires evaluation of the vector products $v \mapsto H_{\tilde{r}}^j v$ which can be implemented efficiently by applying the JAX vector-Jacobian function to the map $\phi \mapsto J^\top \mathtt{sg}[\hat{\Theta}_\lambda^{-1} r]$ where `sg` is the stop-gradient function. Finally, LMC requires estimation of the gradient but naive application of reverse-mode automatic differentiation may result in an *out-of-memory* (OOM) error, especially for larger architectures (the Rademacher vectors s have dimension d). A more memory-efficient method is to compute both the value and gradient of the log determinant approximation in the forward pass.[9]

To illustrate, consider the *second-order Taylor expansion* ($M = 2$ in Equation (11)) which we used in preliminary experiments. Exploiting the symmetry of $A = H_{\tilde{r}}$, we define $\tilde{s}_i \coloneqq A s_i$, and compute *both* the log determinant and its gradient as

$$\begin{aligned} \log\left|\det \partial_\phi R(\phi)\right| &\approx \frac{1}{S}\sum_{i=1}^S s_i^\top A s_i - \tfrac{1}{2} s_i^\top A^2 s_i = \frac{1}{S}\sum_{i=1}^S \tilde{s}_i^\top (s_i - \tfrac{1}{2}\tilde{s}_i) \,, \\ \nabla_\phi \log\left|\det \partial_\phi R(\phi)\right| &\approx \nabla_\phi \frac{1}{S}\sum_{i=1}^S s_i^\top A s_i - \tfrac{1}{2} s_i^\top A^2 s_i = \frac{1}{S}\sum_{i=1}^S \left(\tfrac{\partial\tilde{s}_i}{\partial\phi}\right)^\top (s_i - \tilde{s}_i) \,, \end{aligned} \tag{14}$$

where $s_i - \tilde{s}_i \mapsto (\frac{\partial\tilde{s}_i}{\partial\phi})^\top (s_i - \tilde{s}_i)$ can be obtained via JAX `vjp`. Since evaluating all the summands at the same time leads to OOM issues, we compute the result sequentially in a loop over mini-batches of a size smaller than S.[10]

F. Non-Gaussian likelihoods

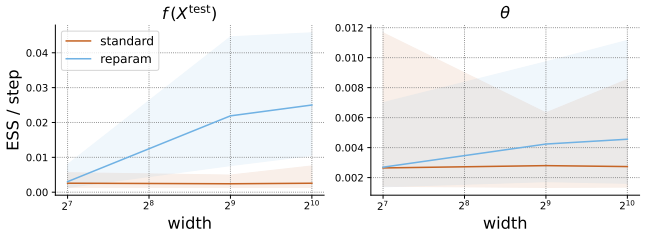

Figure 11. **Our reparametrisation can be combined with non-Gaussian likelihoods.** We ran a smaller version of the experiment in Figure 7 (left) with the categorical likelihood, (much) less extensive hyperparameter tuning, and *only* `100K` sampler steps. We had to lower the acceptance probability cutoff to 90% (from 98%) since the lowest stepsize in our grid was too high for the standard parametrisation (the reparametrised version achieved ∼99% with a higher stepsize). The observed boost is thus likely an underestimate of the real gap. Mapping readout to its true conditional posterior induced by the categorical likelihood would likely work even better.

[9] **Caveat:** JAX allows only defining a custom `jvp`, and can infer the `vjp` rule automatically. However, we found it more memory efficient to explicitly define the forward pass operation, cache the computed gradient, and use the cached value to evaluate the gradient.

[10] The first-order terms can be ignored for nonlinearities with zero second-derivative like ReLU as $\mathrm{diag}(H_i) = 0$ here.