Appendix A A Simple Working Example

Let $\Theta := \mathbb{R}^d$ and let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ defined by $\mathcal{L}(\boldsymbol{\theta}) := \sum_{i=1}^d (\boldsymbol{\theta}_i)^2$ be a twice-differentiable function on Θ. Suppose $\varphi : \Theta \to \Psi$ defined by $\varphi(\boldsymbol{\theta}) = 3\boldsymbol{\theta} =: \boldsymbol{\psi}$ be the reparametrization of choice. In this example, we shall see how this often-used reparametrization leads to pathologies in the values derived from the Hessian of $\mathcal{L}$ computed by an autodiff system.

Notice that the Jacobian of φ is given by $\boldsymbol{J}(\boldsymbol{\theta}) = \operatorname{diag}(3, \dots, 3) \in \mathbb{R}^{d \times d}$ and its inverse is $\boldsymbol{J}^{-1}(\boldsymbol{\psi}) = \operatorname{diag}(1/3, \dots, 1/3)$. The autodiff Hessian matrix is given by $\boldsymbol{H}(\boldsymbol{\theta}) \equiv 2\boldsymbol{I} \in \mathbb{R}^{d \times d}$. When we transform $\boldsymbol{\theta} \to \boldsymbol{\psi}$, the function $\mathcal{L}(\boldsymbol{\theta})$ becomes $\hat{\mathcal{L}}(\boldsymbol{\psi}) := \sum_{i=1}^d (1/3 \boldsymbol{\psi}_i)^2$. The autodiff will take into account this change and thus will compute the Hessian $\hat{\boldsymbol{H}}$ of $\hat{\mathcal{L}}$ as

$$\hat{\boldsymbol{H}}_{ij}(\boldsymbol{\psi}) = \operatorname{diag}(1/3, \dots, 1/3)\,(2\boldsymbol{I})\operatorname{diag}(1/3, \dots, 1/3) = \operatorname{diag}(2/9, \dots, 2/9).$$

Notice that, $\det \boldsymbol{H}(\boldsymbol{\theta}) \equiv 2^d$, meanwhile, we have $\det \hat{\boldsymbol{H}}(\boldsymbol{\psi}) \equiv (2/9)^d$. Therefore, the Hessian determinant computed by an autodiff system is not invariant under reparametrization.

But if instead we explicitly take into account the metric into the Hessian, which by default is (implicitly) chosen to be $\boldsymbol{G}(\boldsymbol{\theta}) \equiv \boldsymbol{I}$, resulting in the seemingly redundant expression $\boldsymbol{E}(\boldsymbol{\theta}) = \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{H}(\boldsymbol{\theta}) = \boldsymbol{I}^{-1}(2\boldsymbol{I})$, and perform the correct transformation on both identity matrices, we obtain

$$\begin{aligned}
\hat{\boldsymbol{E}}(\boldsymbol{\psi}) &= (\operatorname{diag}(1/3, \dots, 1/3)\,\boldsymbol{I}\operatorname{diag}(1/3, \dots, 1/3))^{-1}\operatorname{diag}(1/3, \dots, 1/3)\,(2\boldsymbol{I})\operatorname{diag}(1/3, \dots, 1/3) \\
&= \operatorname{diag}(3, \dots, 3)\cancel{\operatorname{diag}(3, \dots, 3)}\cancel{\operatorname{diag}(1/3, \dots, 1/3)}\,(2\boldsymbol{I})\operatorname{diag}(1/3, \dots, 1/3) \\
&= \operatorname{diag}(3, \dots, 3)\,(2\boldsymbol{I})\operatorname{diag}(1/3, \dots, 1/3).
\end{aligned}$$

Thus, the determinant of the transformed "metric-conditioned Hessian" equals

$$\det \hat{\boldsymbol{E}}(\boldsymbol{\psi}) = (\det \operatorname{diag}(3, \dots, 3))\,(\det 2\boldsymbol{I})(\det \operatorname{diag}(1/3, \dots, 1/3)) = 3^d 2^d (1/3)^d = 2^d,$$

which coincides with the determinant of the original $\boldsymbol{E}$.

Appendix B Derivations

Note. Throughout this section, we use the Einstein summation convention: If the same index appears twice, once as an upper index and once as a lower index, we sum them over. For example: $z = x^i y_i$ means $z = \sum_i x^i y_i$ and $B_j^k = A_{ij}^k x^i$ means $B_j^k = \sum_k A_{ij}^k x^i$, etc. Specifically for partial derivatives, the index of the denominator is always treated as a lower index. ■

B.1 The Riemannian Hessian Under Reparametrization

Let $\mathcal{L} : M \to \mathbb{R}$ be a function on a Riemannian manifold M with metric G. The Riemannian Hessian $\operatorname{Hess}\mathcal{L}$ of $\mathcal{L}$ is defined in coordinates θ by

$$\boldsymbol{H}_{ij} = \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}^i \partial \boldsymbol{\theta}^j} - \boldsymbol{\Gamma}_{ij}^k \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}^k}, \tag{6}$$

where $\boldsymbol{\Gamma}_{ij}^k$ is the connection coefficient.

Under a change of coordinates $\varphi : \boldsymbol{\theta} \mapsto \boldsymbol{\psi}$, we have $\tilde{\mathcal{L}} = \mathcal{L} \circ \varphi^{-1}$ and

$$\tilde{\boldsymbol{\Gamma}}_{ij}^k = \boldsymbol{\Gamma}_{mn}^o \frac{\partial \boldsymbol{\psi}^k}{\partial \boldsymbol{\theta}^o} \frac{\partial \boldsymbol{\theta}^m}{\partial \boldsymbol{\psi}^i} \frac{\partial \boldsymbol{\theta}^n}{\partial \boldsymbol{\psi}^j} + \frac{\partial^2 \boldsymbol{\theta}^o}{\partial \boldsymbol{\psi}^i \partial \boldsymbol{\psi}^j} \frac{\partial \boldsymbol{\psi}^k}{\partial \boldsymbol{\theta}^o}, \tag{7}$$

where m, n, o are just dummy indices—present to express summations. Note that the transformation rule for $\boldsymbol{\Gamma}_{ij}^k$ implies that it is *not* a tensor—to be a tensor, there must not be the second term in the previous formula.

Using (7) and the standard chain & product rules to transform the partial derivatives in (6), we obtain the coordinate representation of the transformed Hessian $\operatorname{Hess}\tilde{\mathcal{L}}$:

$$\begin{aligned}
\tilde{\boldsymbol{H}}_{ij} &= \frac{\partial^2(\mathcal{L}\circ\varphi^{-1})}{\partial\boldsymbol{\psi}^i\partial\boldsymbol{\psi}^j} - \tilde{\boldsymbol{\Gamma}}^k_{ij}\frac{\partial(\mathcal{L}\circ\varphi^{-1})}{\partial\boldsymbol{\psi}^k} \\
&= \frac{\partial^2\mathcal{L}}{\partial\boldsymbol{\theta}^m\partial\boldsymbol{\theta}^n}\frac{\partial\boldsymbol{\theta}^m}{\partial\boldsymbol{\psi}^i}\frac{\partial\boldsymbol{\theta}^n}{\partial\boldsymbol{\psi}^j} + \frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}^o}\frac{\partial^2\boldsymbol{\theta}^o}{\partial\boldsymbol{\psi}^i\partial\boldsymbol{\psi}^j} - \left(\boldsymbol{\Gamma}^o_{mn}\frac{\partial\boldsymbol{\psi}^k}{\partial\boldsymbol{\theta}^o}\frac{\partial\boldsymbol{\theta}^m}{\partial\boldsymbol{\psi}^i}\frac{\partial\boldsymbol{\theta}^n}{\partial\boldsymbol{\psi}^j} + \frac{\partial^2\boldsymbol{\theta}^o}{\partial\boldsymbol{\psi}^i\partial\boldsymbol{\psi}^j}\frac{\partial\boldsymbol{\psi}^k}{\partial\boldsymbol{\theta}^o}\right)\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}^o}\frac{\partial\boldsymbol{\theta}^o}{\partial\boldsymbol{\psi}^k} \\
&= \frac{\partial^2\mathcal{L}}{\partial\boldsymbol{\theta}^m\partial\boldsymbol{\theta}^n}\frac{\partial\boldsymbol{\theta}^m}{\partial\boldsymbol{\psi}^i}\frac{\partial\boldsymbol{\theta}^n}{\partial\boldsymbol{\psi}^j} + \frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}^o}\frac{\partial^2\boldsymbol{\theta}^o}{\partial\boldsymbol{\psi}^i\partial\boldsymbol{\psi}^j} - \boldsymbol{\Gamma}^o_{mn}\cancel{\frac{\partial\boldsymbol{\psi}^k}{\partial\boldsymbol{\theta}^o}}\frac{\partial\boldsymbol{\theta}^m}{\partial\boldsymbol{\psi}^i}\frac{\partial\boldsymbol{\theta}^n}{\partial\boldsymbol{\psi}^j}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}^o}\cancel{\frac{\partial\boldsymbol{\theta}^o}{\partial\boldsymbol{\psi}^k}} - \frac{\partial^2\boldsymbol{\theta}^o}{\partial\boldsymbol{\psi}^i\partial\boldsymbol{\psi}^j}\cancel{\frac{\partial\boldsymbol{\psi}^k}{\partial\boldsymbol{\theta}^o}}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}^o}\cancel{\frac{\partial\boldsymbol{\theta}^o}{\partial\boldsymbol{\psi}^k}} \\
&= \frac{\partial^2\mathcal{L}}{\partial\boldsymbol{\theta}^m\partial\boldsymbol{\theta}^n}\frac{\partial\boldsymbol{\theta}^m}{\partial\boldsymbol{\psi}^i}\frac{\partial\boldsymbol{\theta}^n}{\partial\boldsymbol{\psi}^j} \cancel{+ \frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}^o}\frac{\partial^2\boldsymbol{\theta}^o}{\partial\boldsymbol{\psi}^i\partial\boldsymbol{\psi}^j}} - \boldsymbol{\Gamma}^o_{mn}\frac{\partial\boldsymbol{\theta}^m}{\partial\boldsymbol{\psi}^i}\frac{\partial\boldsymbol{\theta}^n}{\partial\boldsymbol{\psi}^j}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}^o} \cancel{- \frac{\partial^2\boldsymbol{\theta}^o}{\partial\boldsymbol{\psi}^i\partial\boldsymbol{\psi}^j}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}^o}} \\
&= \frac{\partial\boldsymbol{\theta}^m}{\partial\boldsymbol{\psi}^i}\frac{\partial\boldsymbol{\theta}^n}{\partial\boldsymbol{\psi}^j}\left(\frac{\partial^2\mathcal{L}}{\partial\boldsymbol{\theta}^m\partial\boldsymbol{\theta}^n} - \boldsymbol{\Gamma}^o_{mn}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}^o}\right) \\
&= \frac{\partial\boldsymbol{\theta}^m}{\partial\boldsymbol{\psi}^i}\frac{\partial\boldsymbol{\theta}^n}{\partial\boldsymbol{\psi}^j}\boldsymbol{H}_{mn}.
\end{aligned} \tag{8}$$

In the matrix form, we can write the above as $\tilde{\boldsymbol{H}} = \boldsymbol{J}^{-\top}\boldsymbol{H}\boldsymbol{J}^{-1}$, where $\boldsymbol{J}$ is the Jacobian of φ. Thus, the Riemannian Hessian at any $\boldsymbol{\theta}$ (not just at critical points) transforms just like the metric and thus invariant as discussed in Example 3. Note: this only holds when the term containing the connection coefficients $\boldsymbol{\Gamma}^k_{ij}$ is explicitly considered. In particular, the Euclidean Hessian does not follow this tensorial transformation under autodiff due to the fact that (i) $\boldsymbol{\Gamma}^k_{ij} = 0$ for any i, j, k and thus dropped from the equation, *and* (ii) autodiff is not designed to handle advanced geometric objects like $\boldsymbol{\Gamma}^k_{ij}$.

B.2 Hessian-Trace Under Reparametrization

Let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ be a function of $\mathbb{R}^d$ under the Cartesian coordinates and $\boldsymbol{G}$ a Riemannian metric. The ***Riemannian trace*** of the Hessian matrix $\boldsymbol{H}$ of $\mathcal{L}$ is defined by [47]:

$$(\operatorname{tr}_{\boldsymbol{G}}\boldsymbol{H})(\boldsymbol{\theta}) = \operatorname{tr}(\boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{H}(\boldsymbol{\theta})). \tag{9}$$

That is, it is defined as the standard trace of the Hessian operator $\boldsymbol{E}$.

Let $\varphi : \boldsymbol{\theta} \mapsto \boldsymbol{\psi}$ be a reparametrization on $\mathbb{R}^d$. Then, using (2) and the property $\operatorname{tr}(\boldsymbol{AB}) = \operatorname{tr}(\boldsymbol{BA})$ twice, the Riemannian trace of the Hessian transforms into

$$\begin{aligned}
(\operatorname{tr}_{\tilde{\boldsymbol{G}}}\tilde{\boldsymbol{H}})(\boldsymbol{\psi}) &= \operatorname{tr}(\tilde{\boldsymbol{E}}(\boldsymbol{\psi})) \\
&= \operatorname{tr}((\boldsymbol{J}^{-1}(\boldsymbol{\psi}))^{-1}\boldsymbol{G}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{H}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{J}^{-1}(\boldsymbol{\psi})) \\
&= \operatorname{tr}(\boldsymbol{G}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{H}(\varphi^{-1}(\boldsymbol{\psi}))) \\
&= (\operatorname{tr}_{\boldsymbol{G}}\boldsymbol{H})(\varphi^{-1}(\boldsymbol{\psi})).
\end{aligned} \tag{10}$$

Since $\boldsymbol{\psi} = \varphi(\boldsymbol{\theta})$, we have that $(\operatorname{tr}_{\tilde{\boldsymbol{G}}}\tilde{\boldsymbol{H}})(\boldsymbol{\psi}) = (\operatorname{tr}_{\boldsymbol{G}}\boldsymbol{H})(\boldsymbol{\theta})$ for any given $\boldsymbol{\theta}$. Therefore, the trace of the Hessian operator (or the Riemannian trace of the Hessian) is invariant.

B.3 Hessian-Eigenvalues Under Reparametrization

We use the setting from the preceding section. Recall that λ is an eigenvalue of the linear map $\boldsymbol{E}(\boldsymbol{\theta}) = \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{H}(\boldsymbol{\theta})$ on the tangent space at $z \in \mathbb{R}^d$ that is represented $\boldsymbol{\theta}$ if $\boldsymbol{E}(\boldsymbol{\theta})\boldsymbol{v} = \lambda\boldsymbol{v}$ for an eigenvector $\boldsymbol{v} \in T_z\mathbb{R}^d$. We shall show that $\tilde{\lambda}$, the eigenvalue under under the reparametrization $\varphi : \boldsymbol{\theta} \mapsto \boldsymbol{\psi}$, equals the original eigenvalue λ.

Using the transformation rule of $\boldsymbol{E}(\boldsymbol{\theta})$ in (2) and the transformation rule of tangent vectors in (3), along with the relation $(\boldsymbol{J}^{-1}(\boldsymbol{\psi}))^{-1} = \boldsymbol{J}(\varphi^{-1}(\boldsymbol{\psi}))$, we get

$$\begin{aligned}
\tilde{\boldsymbol{E}}(\boldsymbol{\psi})\tilde{\boldsymbol{v}} &= \tilde{\lambda}\tilde{\boldsymbol{v}} \\
(\boldsymbol{J}^{-1}(\boldsymbol{\psi}))^{-1}\boldsymbol{G}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{H}(\varphi^{-1}(\boldsymbol{\psi}))\cancel{\boldsymbol{J}^{-1}(\boldsymbol{\psi})\boldsymbol{J}(\varphi^{-1}(\boldsymbol{\psi}))}\boldsymbol{v} &= \tilde{\lambda}\boldsymbol{J}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{v} \\
\boldsymbol{J}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{G}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{H}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{v} &= \tilde{\lambda}\boldsymbol{J}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{v} \\
\cancel{(\boldsymbol{J}(\varphi^{-1}(\boldsymbol{\psi})))^{-1}}\cancel{\boldsymbol{J}(\varphi^{-1}(\boldsymbol{\psi}))}\boldsymbol{G}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{H}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{v} &= \tilde{\lambda}\boldsymbol{v} \\
\boldsymbol{E}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{v} &= \tilde{\lambda}\boldsymbol{v}.
\end{aligned} \tag{11}$$

Since $\varphi^{-1}(\boldsymbol{\psi}) = \boldsymbol{\theta}$, we arrive back at the original equation before the reparametrization and thus we conclude that $\tilde{\lambda} = \lambda$.

Appendix C The Invariance of the Fisher Metric

We have seen that any metric on the parameter space yields inv(equi)variance when all transformation rules of geometric objects are followed. However, in practical settings, one uses autodiff libraries [1, 66], whose job is to compute only the elementary rules of derivatives such as the chain and product rules. Notice that derivatives and Hessians (at least at critical points) are transformed correctly under autodiff, while the metric is not in general. It is thus practically interesting to find a family of metrics that transform correctly and automatically under reparametrization, given only an autodiff library. Under those metrics, the aforementioned inv(equi)variances can thus be obtained effortlessly.

Martens [55, Sec. 12] mentions the following family of curvature matrices satisfying this transformation behavior

$$\boldsymbol{B}(\boldsymbol{\theta}) \propto \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}\sim D} \left[\boldsymbol{J}(\boldsymbol{\theta};\boldsymbol{x})^\top \boldsymbol{A}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y}) \boldsymbol{J}(\boldsymbol{\theta};\boldsymbol{x})\right] , \tag{12}$$

where $\boldsymbol{J}(\boldsymbol{\theta};\cdot)$ is the network's Jacobian $\partial f(\,\cdot\,;\boldsymbol{\theta})/\partial\boldsymbol{\theta}$, with an arbitrary data distribution D and an invertible matrix $\boldsymbol{A}$ that transforms like $\boldsymbol{A}(\varphi^{-1}(\boldsymbol{\psi}))$ under $\varphi : \boldsymbol{\theta} \mapsto \boldsymbol{\psi}$. Under a reparametrization φ, an autodiff library will compute

$$\hat{\boldsymbol{B}}(\boldsymbol{\psi}) = \boldsymbol{J}^{-1}(\boldsymbol{\psi})^\top \boldsymbol{B}(\varphi^{-1}(\boldsymbol{\psi}))\boldsymbol{J}^{-1}(\boldsymbol{\psi}) , \tag{13}$$

given $\mathcal{L} \circ \varphi^{-1}$, due to the elementary transformation rule of $\boldsymbol{J}(\boldsymbol{\theta};\cdot)$, just like Example 3(c). This family includes the Fisher and the generalized Gauss-Newton matrices, as well as the empirical Fisher matrix [44, 55].

However, note that any $\boldsymbol{B}$ as above is sufficient. Thus, this family is much larger than the Fisher metric, indicating that automatic invariance is not unique to that metric and its denominations. Moreover, in practice, these metrics are often substituted by structural approximations, such as their diagonal and Kronecker factorization [56]. This restricts the kinds of reparametrizations under which these approximate metrics transform automatically: Take the diagonal of (12) as an example. Its ψ-representation computed by the autodiff library will only match the correct transformation for element-wise reparametrizations, whose Jacobian $\boldsymbol{J}(\boldsymbol{\theta})$ is diagonal. On top of that, in practical algorithms, such approximations are usually combined with additional techniques such as damping or momentum, which further break their automaticness.

Due to the above and because any metric is theoretically in(equi)variant if manual intervention is performed, the inv(equi)variance property of the Fisher metric should not be the determining factor of using the Fisher metric. Instead, one should look into its more unique properties such as its statistical efficiency [3] and guarantees in optimization [34, 78]. Automatic transformation is but a cherry on top.

It is interesting for future work to extend autodiff libraries to take into account the invariant transformation rules of geometric objects. By doing so, *any* metric—not just the Fisher metric—will yield invariance *automatically*.

Appendix D General Manifold: Local Coordinate Charts

In the main text, we have exclusively used global coordinate charts $(\mathbb{R}^d, \theta)$ and $(\mathbb{R}^d, \psi)$—these charts cover the entire $\mathbb{R}^d$ and θ, ψ are homeomorphisms on $\mathbb{R}^d$. However, there exist other coordinate systems that are not global, i.e. they are constructed using multiple charts $\{(U_i \subseteq \mathbb{R}^d, \theta_i : U_i \to \Theta_i \subseteq \mathbb{R}^d)\}_i$ s.t. $\mathbb{R}^d = \cup_i U_i$ and each θ_i is a diffeomorphism on U_i.

If $\{(U_i, \theta_i)\}_i$ and $\{(V_j, \psi_j)\}_j$ be two local coordinate systems of $\mathbb{R}^d$. Let (U_i, θ_i) and (V_j, ψ_j) be an arbitrary pair of charts from the above collections where $U_i \cap V_j \neq \varnothing$. In this setting, reparametrization between these two charts amounts to the condition that the transition map φ between $\theta(U_i \cap V_j)$ and $\psi(U_i \cap V_j)$ is a diffeomorphism, see Fig. 6. Reparametrization on the whole manifold can then be defined if for all pairs of two charts from two coordinate systems, their transition maps are all diffeomorphism whenever they overlap with each other.

Note that this definition is a generalization of global coordinate charts. In this case, there is only a single chart for each coordinate system, i.e. (U, θ) and V, ψ, and they trivially overlap since $U = \mathbb{R}^d$ and $V = \mathbb{R}^d$. Moreover, there is only a single diffeomorphism of concern, as shown in Fig. 3.

Our results in the main text hold in this general case by simply replacing the domain for the discussion to be the overlapping regions of a pair of two charts, instead of the whole $\mathbb{R}^d$.

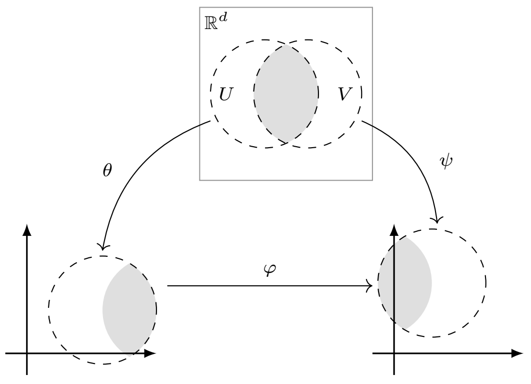

Figure 6: The diffeomorphism φ is defined from $\theta(U \cap V)$ to $\psi(U \cap V)$.

Appendix E Details on Applications

E.1 Infinite-Width Neural Networks

Note We use a 2-layer NN without bias for clarity. The extension to deep networks follows directly by induction—see e.g. [6, 45]. We also use the same prior variance σ^2 without loss of generalization for simplicity. See also Kristiadi et al. [41, Appendix A] for further intuition. Recall the property of Gaussians under linear transformation: $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \implies \boldsymbol{A}\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{\mu}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top)$.

Neural-network Gaussian process (NNGP)

Let $f(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ defined by $f(\boldsymbol{x}) := \boldsymbol{w}^\top \phi(\boldsymbol{W}\boldsymbol{x})$ be a real-valued, 2-layer NN with parameter $\boldsymbol{\theta} := \{\boldsymbol{W} \in \mathbb{R}^{h \times n}, \boldsymbol{w} \in \mathbb{R}^h\}$ and component-wise nonlinearity ϕ. Note that we assume $\boldsymbol{x}$ is i.i.d. Let $\operatorname{vec}(\boldsymbol{W}) \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ and $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \frac{\sigma^2}{h}\boldsymbol{I})$ be priors over the weights. This parametrization of the weights and the priors is called the ***standard parametrization (SP)*** [45, 60].

Step (a) Given a particular preactivation value $\boldsymbol{z}$, we have a linear model $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{z})$. We can view this as a Gaussian process with mean and covariance

$$\mathbb{E}[f(\boldsymbol{x})] = \boldsymbol{0}^\top \phi(\boldsymbol{z}) = 0,$$
$$\operatorname{Cov}[f(\boldsymbol{x}), f(\boldsymbol{x}')] = \frac{\sigma^2}{h}\phi(\boldsymbol{z})^\top \phi(\boldsymbol{z}') = \frac{\sigma^2}{h}\sum_{i=1}^{h} \phi(\boldsymbol{z}_i)\, \phi(\boldsymbol{z}_i').$$

Taking the limit as $h \to \infty$, the mean stays trivially zero and we have by the law of large numbers:

$$K(\boldsymbol{x}, \boldsymbol{x}') := \lim_{h\to\infty} \operatorname{Cov}[f(\boldsymbol{x}), f(\boldsymbol{x}')] = \lim_{h\to\infty} \frac{\sigma^2}{h}\sum_{i=1}^{h} \phi(\boldsymbol{z}_i)\, \phi(\boldsymbol{z}_i') = \sigma^2 \mathop{\mathbb{E}}_{\boldsymbol{z}_i, \boldsymbol{z}_i'}[\phi(\boldsymbol{z}_i)\phi(\boldsymbol{z}_i')].$$

In particular, both the mean and covariance over the output does *not* depend on the particular realization of the hidden units $\phi(\boldsymbol{z})$. That is, they only depend on the *distribution of $\boldsymbol{z}$ induced by the prior*, which we will obtain now.

Step (b) Notice that each $\boldsymbol{z}_i = \boldsymbol{W}_i^\top \boldsymbol{x}$ where $\boldsymbol{W}_i$ is the i-th row of $\boldsymbol{W}$. This is a linear model and thus, $\boldsymbol{z}$ is distributed as a GP with mean and covariance

$$\mathbb{E}[\boldsymbol{z}_i] = \boldsymbol{0}^\top \boldsymbol{x} = 0,$$
$$\operatorname{Cov}[\boldsymbol{z}_i, \boldsymbol{z}_i'] = \sigma^2 \boldsymbol{x}^\top \boldsymbol{x}' =: K_z(\boldsymbol{z}_i, \boldsymbol{z}_i').$$

So, $\boldsymbol{z}_i \sim \mathcal{GP}(0, K_z)$ Since the prior over $\boldsymbol{W}$ is i.i.d., this holds for all $i = 1, \dots, h$. We can thus now compute the expectation in $K(\boldsymbol{x}, \boldsymbol{x}')$: it is done w.r.t. this GP over $\boldsymbol{z}_i$.

To obtain the GP over the function output of a deep network, simply apply steps (a) and (b) above recursively. The crucial message from this derivation is that as the width of each layer of a deep net goes to infinity, the network loses representation power—the output of each layer only depends on the prior, and not on particular values (e.g. learned) of the previous hidden units. In this sense, an infinite-width L-layer NN is simply a linear model with a constant feature extractor induced by the network's first $L-1$ layers that are fixed at initialization. Note that the kernel K over the function output is called the ***NNGP kernel*** [45].

Neural tangent kernel (NTK)

Let us transform $\boldsymbol{w}$ into $\boldsymbol{v} := \frac{\sigma}{\sqrt{h}}\boldsymbol{w}$ and $\boldsymbol{W}$ into $\boldsymbol{V} := \frac{\sigma}{\sqrt{h}}\boldsymbol{W}$ and define the prior to be $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\operatorname{vec}(\boldsymbol{W}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Then, we define the transformed network as $\hat{f}(\boldsymbol{x}) := \boldsymbol{v}^\top \phi(\boldsymbol{V}\boldsymbol{x}) = \frac{\sigma}{\sqrt{h}}\boldsymbol{w}^\top \phi\left(\frac{\sigma}{\sqrt{h}}\boldsymbol{W}\boldsymbol{x}\right)$ with parameter $\boldsymbol{\psi} := \{\boldsymbol{V} \in \mathbb{R}^{h \times n}, \boldsymbol{v} \in \mathbb{R}^h\}$. This is called the ***NTK parametrization (NTP)*** [29]. We will see below that even though $\boldsymbol{v}$, $\boldsymbol{V}$ have the same prior distributions as $\boldsymbol{w}$, $\boldsymbol{W}$ in the SP, they have different behavior in terms of the NTK.

As before, let us assume a particular preactivation value $\boldsymbol{z}$. The ***empirical NTK*** (i.e. finite-width NTK) on the last layer is defined by:

$$\hat{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \nabla_{\boldsymbol{w}} \hat{f}(\boldsymbol{x}), \nabla_{\boldsymbol{w}} \hat{f}(\boldsymbol{x}') \rangle = \frac{\sigma^2}{h} \sum_{i=1}^{h} \phi(\boldsymbol{z}_i)\, \phi(\boldsymbol{z}'_i).$$

The ***(asymptotic) NTK*** is obtained by taking the limit of $h \to \infty$:

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') := \lim_{h \to \infty} \hat{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\sigma^2}{h} \sum_{i=1}^{h} \phi(\boldsymbol{z}_i)\, \phi(\boldsymbol{z}'_i) = \mathop{\mathbb{E}}_{\boldsymbol{z}_i, \boldsymbol{z}'_i} \left[\phi(\boldsymbol{z}_i)\phi(\boldsymbol{z}'_i)\right], \tag{14}$$

which coincides the NNGP kernel K. Crucially, this is obtained via a backward propagation from the output of the network and thus the linear-Gaussian property we have used to derive the NNGP via forward propagation does not apply.[3] This is why the scaling of $\frac{\sigma}{h}$ is required in the NTP. That is, using the SP, the empirical NTK is not scaled by $\frac{\sigma^2}{h}$ and thus when taking the limit to obtain $\mathcal{K}$, the sum diverges and the limit does not exist.

Is the NTP a reparametrization of the SP?

It is tempting to treat the NTP as a reparametrization of the SP—in fact, it is standard in the literature to treat them as two different parametrizations of the same network. However, we show that geometrically, this is inaccurate. Indeed from the geometric perspective, if two functions are reparametrization of each other, they should be invariant, as we have discussed in the main text. Instead, we show that the different limiting behaviors are present because the NTP and SP assume two different functions and two different priors—they are *not* connected by a reparametrization. This clears up confusion and provides a foundation for future work in this field: To obtain a desired limiting behavior, study the network architecture and its prior, instead of the parametrization.

Suppose $\boldsymbol{\psi}$ in the NTP is a reparametrization of $\boldsymbol{\theta}$ in the SP. Then the function $\varphi : \boldsymbol{\theta} \mapsto \boldsymbol{\psi}$ defined by $\boldsymbol{\theta} \mapsto \frac{\sigma}{\sqrt{h}}\boldsymbol{\theta}$ is obviously the smooth reparametrization with an invertible (diagonal) Jacobian $\boldsymbol{J}(\boldsymbol{\theta}) = \frac{\sigma}{\sqrt{h}}\boldsymbol{I}$. In this case, the network in the NTP must be defined by $\tilde{f} = f \circ \varphi^{-1}$, where f is the SP-network, by Example 3. That is, with some abuse of notation,

$$\tilde{f}(\boldsymbol{x}) = \varphi^{-1}(\boldsymbol{v})^\top \phi(\varphi^{-1}(\boldsymbol{V})\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{W}\boldsymbol{x}) = f(\boldsymbol{x}).$$

This is different from the definition of the NTP-network $\hat{f}(\boldsymbol{x}) = \boldsymbol{v}^\top \phi(\boldsymbol{V}\boldsymbol{x})$. So, obviously, the NTP is not the reparametrization of the SP. Therefore, a clearer way of thinking about the NTP and SP is to treat them as two separate network functions (i.e. two separate architectures)—the scaling factor $\frac{\sigma}{\sqrt{h}}$ should be thought of as part of the layer's functional form instead of as part of the parameter. In particular, they are *not* two representations of a single abstract function.

To verify this, let us compute the NTK of $\tilde{f}(\boldsymbol{x})$ (i.e. treating the scaling as a reparametrization) at its last layer. The derivation is based on Section 3.2. First, notice that the differential $\nabla_{\boldsymbol{w}} f(\boldsymbol{x})$ transforms into $\boldsymbol{J}^{-1}(\boldsymbol{v})^\top \boldsymbol{\nabla} \tilde{f}(\boldsymbol{x})|_{\varphi^{-1}(\boldsymbol{v})}$ for any $\boldsymbol{x} \in \mathbb{R}^n$. Next, notice that the Euclidean metric transforms into $\tilde{\boldsymbol{G}}(\boldsymbol{v}) := \boldsymbol{J}^{-1}(\boldsymbol{v})^\top \boldsymbol{J}^{-1}(\boldsymbol{v})$. So the gradient transforms into $\boldsymbol{J}(\varphi^{-1}(\boldsymbol{v}))\boldsymbol{\nabla} f(\boldsymbol{x})|_{\varphi^{-1}(\boldsymbol{v})}$.[4] Therefore,

[3]It still applies for obtaining the distribution of $\boldsymbol{z}_i$. The NTK can thus be thought of as a kernel that arises from performing forward *and* backward propagations once at initialization [6, 75]. This can be seen in the expression of the NTKs on lower layers which decompose into the NNGP and an expression similar to (14), but involving the derivative of ϕ [29].

[4]We use the gradient to get the NTK since otherwise it does not make sense to take the inner product of differentials w.r.t. the metric.

Table 2: Test accuracies, averaged over 5 random seeds.

Methods	MNIST	FMNIST	CIFAR10	CIFAR100
SGD	99.3	92.9	94.9	76.8
ADAM	99.2	92.6	92.4	71.9

the empirical NTK kernel $\hat{\mathcal{K}}_\Psi$ for $\tilde{f}$ is given by

$$\begin{aligned}\hat{\mathcal{K}}_\Psi(\boldsymbol{x}, \boldsymbol{x}') &= \langle \boldsymbol{J}(\varphi^{-1}(\boldsymbol{v}))\boldsymbol{\nabla} f(\boldsymbol{x})|_{\varphi^{-1}(\boldsymbol{v})}, \boldsymbol{J}(\varphi^{-1}(\boldsymbol{v}))\boldsymbol{\nabla} f(\boldsymbol{x}')|_{\varphi^{-1}(\boldsymbol{v})}\rangle_{\tilde{\boldsymbol{G}}(\boldsymbol{v})} \\ &= (\boldsymbol{J}(\varphi^{-1}(\boldsymbol{v}))\boldsymbol{\nabla} f(\boldsymbol{x})|_{\varphi^{-1}(\boldsymbol{v})})^\top \tilde{\boldsymbol{G}}(\boldsymbol{v}) \boldsymbol{J}(\varphi^{-1}(\boldsymbol{v}))\boldsymbol{\nabla} f(\boldsymbol{x}')|_{\varphi^{-1}(\boldsymbol{v})} \\ &= (\boldsymbol{\nabla} f(\boldsymbol{x})|_{\varphi^{-1}(\boldsymbol{v})})^\top \cancel{\boldsymbol{J}(\varphi^{-1}(\boldsymbol{v}))^\top} \cancel{\boldsymbol{J}^{-1}(\boldsymbol{v})^\top} \cancel{\boldsymbol{J}^{-1}(\boldsymbol{v})} \cancel{\boldsymbol{J}(\varphi^{-1}(\boldsymbol{v}))} \boldsymbol{\nabla} f(\boldsymbol{x}')|_{\varphi^{-1}(\boldsymbol{v})} \\ &= \langle \boldsymbol{\nabla} f(\boldsymbol{x})|_{\varphi^{-1}(\boldsymbol{v})}, \boldsymbol{\nabla} f(\boldsymbol{x}')|_{\varphi^{-1}(\boldsymbol{v})}\rangle.\end{aligned}$$

Thus, the empirical NTK is invariant and the asymptotic NTK also is. Therefore, we still have a problem with the NTK blow-up in this parametrization. This reinforces the fact that the difference between the SP and NTP is *not* because of parametrization.

Additionally, let us now inspect the priors in the SP and NTP. In the SP, the prior is $\mathcal{N}(\boldsymbol{\theta} \mid \mathbf{0}, \sigma^2/h \boldsymbol{I})$. Therefore, so that we have the same prior in both Θ and Ψ, the prior of $\boldsymbol{\psi} = \varphi(\boldsymbol{\theta})$ must be $\mathcal{N}(\boldsymbol{\psi} \mid \mathbf{0}, \boldsymbol{I})$. This is obviously not the case since we have $\mathcal{N}(\boldsymbol{\psi} \mid \mathbf{0}, \sigma^2/h \boldsymbol{I})$ because the NTP explicitly defines $\mathcal{N}(\boldsymbol{\theta} \mid \mathbf{0}, \boldsymbol{I})$ as the prior of $\boldsymbol{\theta}$. Thus, not only that the SP and NTP assume two different architectures, but they also assume two different prior altogether. It is thus not surprising that the distribution over their network outputs $f(\boldsymbol{x})$, $\hat{f}(\boldsymbol{x})$ are different, both in the finite- and infinite-width regimes.

Implication In his seminal work, Neal [60] concluded that the fact that infinite-width NNs are Gaussian processes disappointing. However, as we have seen in the discussion above, different functional forms, architectures, and priors of NNs yield different limiting behaviors. Therefore, this gives us hope that meaningful, non-GP infinite-width NNs can be obtained. Indeed, Yang and Hu [75], Yang et al. [76] have recently shown us a way to do so. However, they argue that their feature-learning limiting behavior is due to a different parametrization, contrary to the present work. Our work thus complements theirs and opens up the avenue for constructing non-trivial infinite-width NNs in a "Bayesian" way, in the sense that we achieve the desired limiting behaviors by varying the model and the prior.

E.2 Biases of Preconditioned Optimizers

For MNIST and FMNIST, the network is LeNet. Meanwhile, we use the WiderResNet-16-4 model for CIFAR-10 and -100. For ADAM, we use the default setting suggested by Kingma and Ba [39]. For SGD, we use the commonly-used learning rate of 0.1 with Nesterov momentum 0.9 [26]. The cosine annealing method is used to schedule the learning rate for 100 epochs. The test accuracies are in Table 2. Additionally, in Table 3, we discuss the effect of reparametrization to sharpness on ADAM and SGD.

E.3 Laplace Marginal Likelihood

Let $\boldsymbol{\theta}_{\text{MAP}}$ be a MAP estimate in an arbitrary θ-coordinates of $\mathbb{R}^d$, obtained by minimizing the MAP loss $\mathcal{L}_{\text{MAP}}$. Let $\log h = -\mathcal{L}_{\text{MAP}}$—note that $\mathcal{L}_{\text{MAP}}$ itself is a log-density function. The Laplace marginal likelihood [18, 28, 54] is obtained by performing a second-order Taylor's expansion:

$$\log h(\boldsymbol{\theta}) \approx \log h(\boldsymbol{\theta}_{\text{MAP}}) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}})^\top \boldsymbol{H}(\boldsymbol{\theta}_{\text{MAP}})(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}),$$

where $\boldsymbol{H}(\boldsymbol{\theta}_{\text{MAP}})$ is the Hessian matrix of $\mathcal{L}_{\text{MAP}}$ at $\boldsymbol{\theta}_{\text{MAP}}$. Then, by exponentiating and taking the integral over $\boldsymbol{\theta}$, we have

$$Z(\boldsymbol{\theta}_{\text{MAP}}) \approx h(\boldsymbol{\theta}_{\text{MAP}}) \int_{\mathbb{R}^d} \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}})^\top \boldsymbol{H}(\boldsymbol{\theta}_{\text{MAP}})(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}})\right) d\boldsymbol{\theta}. \qquad (15)$$

Table 3: Hessian-based sharpness measures can change under reparametrization without affecting the model's generalization (results on CIFAR-10). The generalization gap is the test accuracy, subtracted from the train accuracy—lower is better. Under the default parametrization, SGD achieves lower sharpness and generalizes better than ADAM which achieves higher sharpness. However, one can reparametrize SGD's minimum s.t. it achieves much higher (or lower) sharpness than ADAM while retaining the same generalization performance. Hence, it is hard to study the correlation between sharpness and generalization. This highlights the need for invariance.

Optimizer	**Reparametrization** $\boldsymbol{\psi}_{\text{MAP}} = \varphi(\boldsymbol{\theta}_{\text{MAP}})$	**Generalization gap** $[\%]$	**Sharpness** $\operatorname{tr}(\hat{\boldsymbol{H}}(\boldsymbol{\psi}_{\text{MAP}}))$
ADAM	$\boldsymbol{\psi}_{\text{MAP}} = \boldsymbol{\theta}_{\text{MAP}}$	7.2 ± 0.2	1929.8 ± 61.2
SGD	$\boldsymbol{\psi}_{\text{MAP}} = \boldsymbol{\theta}_{\text{MAP}}$	5.2 ± 0.2	1531.8 ± 14.2
	$\boldsymbol{\psi}_{\text{MAP}} = \frac{1}{2}\boldsymbol{\theta}_{\text{MAP}}$		6143.7 ± 60.8
	$\boldsymbol{\psi}_{\text{MAP}} = 2\boldsymbol{\theta}_{\text{MAP}}$		383.6 ± 3.3

Since the integral is the normalization constant of the Gaussian $\mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{MAP}}, \boldsymbol{H}(\boldsymbol{\theta}_{\text{MAP}}))$, we obtain the Laplace log-marginal likelihood (LML):

$$\log Z(\boldsymbol{\theta}_{\text{MAP}}) = -\mathcal{L}_{\text{MAP}}(\boldsymbol{\theta}_{\text{MAP}}) - \frac{d}{2}\log(2\pi) + \log\det \boldsymbol{H}(\boldsymbol{\theta}_{\text{MAP}}).$$

Notice that $\boldsymbol{H}(\boldsymbol{\theta})$ is a bilinear form, acting on the tangent vector $\boldsymbol{d}(\boldsymbol{\theta}_{\text{MAP}}) := (\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}})$. Under a reparametrization $\varphi : \boldsymbol{\theta} \mapsto \boldsymbol{\psi}$ with $\boldsymbol{\psi}_{\text{MAP}} = \varphi(\boldsymbol{\theta}_{\text{MAP}})$, the term inside the exponent in (15) transforms into

$$-\frac{1}{2}(\boldsymbol{J}(\varphi^{-1}(\boldsymbol{\psi}_{\text{MAP}}))\boldsymbol{d}(\varphi^{-1}(\boldsymbol{\psi}_{\text{MAP}})))^\top(\boldsymbol{J}^{-1}(\boldsymbol{\psi}_{\text{MAP}})^\top \boldsymbol{H}(\varphi^{-1}(\boldsymbol{\psi}_{\text{MAP}}))\boldsymbol{J}^{-1}(\boldsymbol{\psi}_{\text{MAP}}))$$
$$\boldsymbol{J}(\varphi^{-1}(\boldsymbol{\psi}_{\text{MAP}}))\boldsymbol{d}(\varphi^{-1}(\boldsymbol{\psi}_{\text{MAP}})),$$

due to the transformations of the tangent vector and the bilinear-Hessian. This simplifies into

$$\exp\left(-\frac{1}{2}\boldsymbol{d}(\varphi^{-1}(\boldsymbol{\psi}_{\text{MAP}}))^\top \boldsymbol{H}(\varphi^{-1}(\boldsymbol{\psi}_{\text{MAP}}))\boldsymbol{d}(\varphi^{-1}(\boldsymbol{\psi}_{\text{MAP}}))\right)$$

which always equals the original integrand in (15). Thus, the integral evaluates to the same value. Hence, the last two terms of $\log Z(\boldsymbol{\theta}_{\text{MAP}})$ transform into $-\frac{d}{2}\log(2\pi) + \log\det \boldsymbol{H}(\varphi^{-1}(\boldsymbol{\psi}_{\text{MAP}}))$ in ψ-coordinates. This quantity is thus invariant under reparametrization since it behaves like standard functions.

E.3.1 Experiment Setup

We use the toy regression dataset of size 150. Training inputs are sampled uniformly from $[0, 8]$, while training targets are obtained via $y = \sin x + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.3^2)$. The network is a 1-hidden layer TanH network trained for 1000 epochs.