Emergent Equivariance in Deep Ensembles

Jan E. Gerken [*1] **Pan Kessel** [*2]

Abstract

We show that deep ensembles become equivariant for all inputs and at all training times by simply using full data augmentation. Crucially, equivariance holds off-manifold and for any architecture in the infinite width limit. The equivariance is emergent in the sense that predictions of individual ensemble members are not equivariant but their collective prediction is. Neural tangent kernel theory is used to derive this result and we verify our theoretical insights using detailed numerical experiments.

1. Introduction

Deep ensembles are a standard workhorse of deep learning practitioners (Lakshminarayanan et al., 2017). They operate by averaging the prediction of several networks and therefore offer a straightforward way of estimating the uncertainty of the prediction. For example, deep ensembles are widely used in the medical domain such as in cancer cell detection in pathology or in protein folding in drug design as quantifying the confidence of the output is critical in these fields (Saib et al., 2020; Ruffolo et al., 2023).

The main message of this paper is that deep ensembles offer a novel and straightforward way to enforce equivariance with respect to symmetries of the data. Specifically, we show that upon full data augmentation, deep ensembles become equivariant *at all training steps and for any input* in the large width limit. While this statement would be trivial for a fully trained model and on the data manifold, our results are significantly more powerful in that they also hold off-manifold and even at initialization. A deep ensemble is thus indistinguishable from a fully equivariant network. It is important to emphasize that this manifest equivariance is emergent: while the prediction of the ensemble is equivariant, the predictions of its members are not. In particular, the ensemble members are not required to have an equivariant architecture.

We rigorously derive this surprising emergent equivariance by using the duality between neural networks and kernel machines in the large width limit (Neal, 1996; Lee et al., 2018; Yang, 2020). The neural tangent kernel (NTK) describes the evolution of deep neural networks during training (Jacot et al., 2018). In the limit of infinite width, the neural tangent kernel is frozen, i.e., it does not evolve during training and the training dynamics can be solved analytically. As a random variable over initializations, the output of the neural network after arbitrary training time follows a Gaussian distribution whose mean and covariance are available as closed form expressions (Lee et al., 2019). In this context, deep ensembles can be interpreted as a Monte-Carlo estimate of the corresponding expected network output. This insight allows us to theoretically analyze the effect of data augmentation throughout training and show that the deep ensemble is fully equivariant.

In practice, this emergent equivariance of deep ensemble cannot be expected to hold perfectly and exact equivariance will be broken, since neural networks are not infinitely wide and the expectation value over initalizations is estimated by Monte-Carlo. Furthermore, in the case of a continuous symmetry group, data augmentation cannot cover the entire group orbit and is thus approximate. We analyze the resulting breaking of equivariance and demonstrate empirically that deep ensembles nevertheless show a competetively high degree of equivariance even with a low number of ensemble members.

The main contributions of our work are:

- We prove in our main theorem 5.3 that infinitely wide deep ensembles are equivariant at all stages of training and any input if trained with full data augmentation using the theory of neural tangent kernels.
- We derive bounds for deviations from equivariance due to finite size as well as data augmentation for a continuous group.
- We empirically demonstrate the emergent equivariance

[*] Equal contribution [1] Department of Mathematical Sciences, Chalmers University of Technology and the University of Gothenburg, SE-412 96 Gothenburg, Sweden [2] Prescient Design, Genentech Roche, Basel, Switzerland. Correspondence to: Jan Gerken <gerken@chalmers.se>, Pan Kessel <pan.kessel@roche.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

in three settings: Ising model, FashionMNIST, and a high-dimensional medical dataset of histological slices.

2. Related Works

Deep Ensembles, Equivariance. There is a too extensive body of literature on both equivariance and deep ensembles to be summarized here, reflecting their central importance to modern deep learning. We refer to Gerken et al. (2023) and Ganaie et al. (2022) for reviews, respectively. The relation between manifest equivariance and data augmentation concerning model performance was studied by Gerken et al. (2022) and for training dynamics by Flinth & Ohlsson (2023).

Equivariance without architecture constraints. Equivariance can also be achieved by symmetrizing the network output over (an appropriately chosen subset of) the group orbit (Puny et al., 2021; Basu et al., 2023a;b). This approach is orthogonal to ours: instead of an ensemble of models, an ensemble of outputs is considered. Note that the memory footprint of the symmetrization depends on the size of the group orbit while, for deep ensembles, it depends on the number of ensemble members. Another architecture-agnostic method to reach equivariance is to homogenize inputs using a canonicalization network (Kaba et al., 2023; Mondal et al., 2023). Canonicalization and symmetrization lead to exact equivariance (up to possible discretization effects) while deep ensembles naturally allow for uncertainty estimation and increased robustness.

Neural Tangent Kernel. That Bayesian neural networks behave as Gaussian processes was first discovered by Neal (1996), this result was extended to deep neural networks by Lee et al. (2018). Neural tangent kernels (NTKs), which capture the evolution of wide neural networks under gradient descent training, were introduced by Jacot et al. (2018). The literature on this topic has since expanded considerably so that we can only cite some selected works, a review on the topic is given by Golikov et al. (2022). The NTK for CNNs was computed by Arora et al. (2019). Lee et al. (2019) used the NTK to show that wide neural networks trained with gradient descent become Gaussian processes and Yang (2020) introduced a comprehensive framework to study scaling limits of wide neural networks rigorously. This framework was used by Yang & Hu (2022) to find a parametrization suitable for scaling networks to large width. NTKs were used to study GANs (Franceschi et al., 2022), PINNs (Wang et al., 2022), backdoor attacks (Hayase & Oh, 2022) as well as pruning (Yang & Wang, 2023), amongst other applications. Corrections to the infinite-width limit, in particular in connection to quantum field theory, have been investigated as well (Huang & Yau, 2020; Yaida, 2020; Halverson et al., 2021; Erbin et al., 2022).

Data augmentation and kernel machines. Mroueh et al. (2015), Raj et al. (2017) and Mei et al. (2021) study properties of kernel machines using group-averaged kernels but they do not consider wide neural networks. Dao et al. (2019) use a Markov process to model random data augmentations and show that an optimal Bayes classifier in this context becomes a kernel machine. It is also shown that training on augmented data is equivalent to using an augmented kernel. Li et al. (2019) introduce new forms of pooling to improve kernel machines. As part of their analysis, they derive the analogous augmented kernel results as Dao et al. (2019) for the NTK at infinite training time. In contrast, we focus on the symmetry properties of the resulting (deep) ensemble of infinitely wide neural networks. In particular, we analyze the behavior of the ensemble at finite training time, show that their assumption of an "equivariant kernel" is satisfied under very mild assumptions on the representation (cf. Theorem 5.1), include equivariance on top of invariance and derive a bound for the invariance error accrued by approximating a continuous group with finitely many samples.

3. Deep Ensembles and Neural Tangent Kernels

In this section, we give a brief overview over deep ensembles and their connection to NTKs.

Deep Ensemble. Let $f_w : X \to \mathbb{R}$ be a neural network with parameters w which are initialized by sampling from the density p, i.e. $w \sim p$. For notational simplicity, we consider only scalar-valued networks in the main part of the paper unless stated otherwise. Our results however hold also for vector-valued networks. The output of the deep ensemble $\bar{f}_t$ of the network f_w is then defined as the expected value over initializations of the trained ensemble members

$$\bar{f}_t(x) = \mathbb{E}_{w\sim p}\left[f_{\mathcal{L}_t w}(x)\right] \,, \tag{1}$$

where the operator $\mathcal{L}_t$ maps the initial weight w to its corresponding value after t steps of gradient descent. In practice, the deep ensemble is approximated by a Monte-Carlo estimate of the expectation value using a finite number M of initializations

$$\bar{f}_t(x) \approx \hat{f}_t(x) = \frac{1}{M}\sum_{i=1}^{M} f_{\mathcal{L}_t w_i}(x) \,, \tag{2}$$

where $w_i \sim p$. This amounts to performing M training runs with different initializations and averaging the outputs of the resulting models. It is worthwhile to note that in the literature, the average $\hat{f}_t$ as defined in (2) is often referred to as the deep ensemble (Lakshminarayanan et al., 2017). In this work, we will however use the term deep ensemble to refer to the expectation value $\bar{f}_t$ of (1). Analogously, we refer to $\hat{f}_t$ as the MC estimate of the deep ensemble $\bar{f}_t$.

Relation to NTK. In the infinite width limit, a deep ensemble follows a Gaussian distribution described by the neural tangent kernel (Jacot et al., 2018)

$$\Theta(x, x') = \sum_{l=1}^{L} \mathbb{E}_{w\sim p}\left[\left(\frac{\partial f_w(x)}{\partial w^{(l)}}\right)^{\top} \frac{\partial f_w(x')}{\partial w^{(l)}}\right] , \tag{3}$$

where $w^{(l)}$ denotes the parameters of the l^{th} layer and we have assumed that the network has a total of L layers. Here, the width is taken to infinity, resulting in Gaussian distributions, whose mean and covariance over the initialization distribution is then studied. In general, Θ has additional axes for dimensions not taken to infinity, e.g. pixels in CNNs and output channels in MLPs, which we will keep implicit in most of the main part. In the following, we use the notation

$$\Theta_{ij} = \Theta(x_i, x_j) \tag{4}$$

for the Gram matrix, i.e. the kernel evaluated on two elements x_i and x_j of the training set

$$\mathcal{T} = (\mathcal{X}, \mathcal{Y}) = \{(x_i, y_i) \,|\, i = 1, \dots, |\mathcal{T}|\} . \tag{5}$$

Using the NTK, we can analytically calculate the distribution of ensemble members in the large width limit for a given input x at any training time t for learning rate η: networks trained with the MSE loss follow a Gaussian process distribution with mean function μ_t and covariance function Σ_t which are given in terms of the NTK by Lee et al. (2019)

$$\mu_t(x) = \Theta(x, x_i) \, \left[\Theta^{-1} T_t\right]_{ij} y_j \, , \tag{6}$$

$$\Sigma_t(x, x') = \mathcal{K}(x, x') + \Sigma_t^{(1)}(x, x') - (\Sigma_t^{(2)}(x, x') + \text{h.c.}) , \tag{7}$$

where $T_t = (\mathbb{I} - \exp(-\eta\Theta t))$ and all sums over the training set are implicit by the Einstein summation convention and we have defined

$$\Sigma_t^{(1)}(x, x') = \Theta(x, x_i) \left[\Theta^{-1} T_t \, \mathcal{K} T_t \, \Theta^{-1}\right]_{ij} \Theta(x_j, x') ,$$
$$\Sigma_t^{(2)}(x, x') = \Theta(x, x_i) \, \left[\Theta^{-1} T_t\right]_{ij} \mathcal{K}(x_j, x') ,$$

with the NNGP kernel

$$\mathcal{K}(x, x') = \mathbb{E}_{w\sim p}\left[f_w(x) \, f_w(x')\right] . \tag{8}$$

The Gram matrix of the NNGP is given by $\mathcal{K}_{ij} = \mathcal{K}(x_i, x_j)$. For Σ_t in a less compact notation, see (47) in Appendix A.

Remark 3.1. The function μ_t in (6) captures the mean output of networks trained on different initializations for time t. Therefore, it is just the expected ensemble output (1), $\bar{f}_t(x) = \mu_t(x)$. The variance of this quantity is given by the covariance function evaluated at identical arguments $\Sigma_t(x) := \Sigma_t(x, x)$.

In Appendix A we provide a brief review of NTK theory for readers unfamiliar with it.

In practice, the cost of inverting the Gram matrix is prohibitive. Therefore, one typically estimates the deep ensemble by (2) using M trained models with different random initalizations. Nevertheless, the dual NTK description allows us to reason about the properties of the exact deep ensemble. In the following, we will use this duality to theoretically investigate the effect of data augmentation on deep ensembles.

4. Equivariance and Data Augmentation

In this section, we summarize basics facts about representations of groups, equivariance, and data augmentation and establish our notation.

Representations of Groups. Groups abstractly describe symmetry transformations. In order to describe how a group transforms a vector, we use group representations. A (linear) representation of a group G is a map $\rho : G \to \mathrm{GL}(V)$ where V is a vector space and ρ is a group homomorphism, i.e. $\rho(g_1)\rho(g_2) = \rho(g_1 g_2)$ for all $g_1, g_2 \in G$. A representation is called orthogonal if $\rho(g^{-1}) = \rho(g)^{\top}$, i.e., if it has orthogonal representation matrices.

Equivariance. For learning tasks in which data x and labels y transform under group representations, the map $x \mapsto y$ has to be compatible with the symmetry group; this property is called equivariance. Formally, let $f : X \to Y$ denote a (possibly vector valued) model with input space X and output space Y on which the group G acts with representations ρ_X and ρ_Y, respectively. Then, f is equivariant with respect to the representations ρ_X and ρ_Y if it obeys

$$\rho_Y(g) f(x) = f(\rho_X(g) x) \qquad \forall x \in X, g \in G . \tag{9}$$

Similarly, a model f is invariant with respect to the representation ρ_X if it satisfies the above relation with ρ_Y being the trivial representation, i.e. $\rho_Y(g) = \mathbb{I}$ for all $g \in G$. Considerable work has been done to construct manifestly equivariant neural networks with respect to specific, practically important special cases of (9). It has been shown both empirically (e.g. in Thomas et al. (2018), Bekkers et al. (2018)) and theoretically (e.g. in Sannai et al. (2021), Elesedy & Zaidi (2021)) that equivariance can lead to better sample efficiency, improved training speed and greater robustness. A downside of equivariant architectures is that they need to be purpose-built for symmetry properties of the problem at hand since standard well-established architectures are mostly not equivariant.

Data Augmentation. An alternative approach to incorporate information about the symmetries of the data into the model is data augmentation. Instead of using the original training set $\mathcal{T}$, we use a set which is augmented by all elements of the group orbit, i.e.

$$\mathcal{T}_{\text{aug}} = \{(\rho_X(g)x, \rho_Y(g)y) | g \in G, (x, y) \in \mathcal{T}\} . \quad (10)$$

In stochastic gradient descent, we randomly draw a minibatch from this augmented training set to estimate the gradient of the loss. If the group has finite order, data augmentation has the immediate consequence that the action of any group element $g \in G$ on a training sample can be written as a permutation π_g of the indices of the augmented training set $\mathcal{T}_{\text{aug}}$, i.e.

$$\rho_X(g)x_i = x_{\pi_g(i)} \qquad \text{and} \qquad \rho_Y(g)y_i = y_{\pi_g(i)} , \quad (11)$$

where $i \in \{1, \dots, |\mathcal{T}_{\text{aug}}|\}$. Data augmentation has the advantage that it does not impose any restrictions on the architecture and is hence straightforward to implement. However, the symmetry is only learned and it can thus be expected that the model is only (approximately) equivariant towards the end of training and on the data manifold. Furthermore, the model cannot benefit from the restricted function space which the symmetry constraint specifies.

5. Emergent Equivariance for Large-Width Deep Ensembles

In this section, we prove that any large-width deep ensemble is emergently equivariant when data augmentation is used. After stating our assumptions, the sketch the proof in three steps.

Assumptions. We consider a finite group G with representations ρ_X and ρ_Y as well as data augmentation with respect to these representations, as discussed above. The case of continuous groups will be discussed subsequently. If the input or output have spatial axes a, the representations ρ_X and ρ_Y act via a representation ρ on that domain,

$$\rho_X(g)x_i^a = \tau_X(g)x^{\rho^{-1}(g)a} \quad (12)$$
$$\rho_Y(g)y_i^a = \tau_Y(g)y^{\rho^{-1}(g)a} . \quad (13)$$

The representations $\tau_{X,Y}$ are assumed to be orthogonal and act on the channel dimensions of the input and output. E.g. for rotations on images in the input, $\tau_X = \mathbb{I}$ and ρ is the fundamental representation of $SO(2)$ in terms of 2×2 rotation matrices. For graph neural networks, we consider orthogonal transformations of the node features, $\rho_X(g)x^v = \tau_X(g)x^v$ with node index v. Hence, in this case $\rho = \mathbb{I}$ is trivial. Our results hold for fairly general architectures consisting of convoluational, fully-connected, and flattening layers as well as local aggregation layers in graph neural networks trained on the MSE loss. To illustrate the underlying techniques of the proof, we will prove each step using the simple example of a MLP with a single channel dimension before stating the general results derived in the appendix.

Step 1: The representation ρ_X acting on the input space X induces a canonical transformation of the NTK and NNGP kernel

$$\Theta(x, x') \;\to\; \Theta(\rho_X(g)x, \rho_X(g)x') \quad (14)$$
$$\mathcal{K}(x, x') \;\to\; \mathcal{K}(\rho_X(g)x, \rho_X(g)x') . \quad (15)$$

For a representation ρ_X acting on the input space X, this canonical transformation induces a transformation of the output indices as specified by the following theorem:

Theorem 5.1 (Kernel transformation)**.** *Let G be a group and ρ_X a representation of G acting on the input space X as in (12). Then, the neural tangent kernel Θ, as defined in (3), as well as the NNGP kernel $\mathcal{K}$, as defined in (8), of a neural network satisfying the assumptions above transform according to*

$$\Theta(\rho_X(g)x, \rho_X(g)x') = \rho_K(g)\Theta(x, x')\rho_K^\top(g) , \quad (16)$$
$$\mathcal{K}(\rho_X(g)x, \rho_X(g)x') = \rho_K(g)\mathcal{K}(x, x')\rho_K^\top(g) , \quad (17)$$

for all $g \in G$ and $x, x' \in X$, where ρ_K is a transformation acting on the spatial dimensions of the kernels according to $\rho_K(g)K^a = K^{\rho^{-1}(g)a}$. If the kernels do not have spatial axes, $\rho_K = \mathbb{I}$.

Proof. See Appendix B.

Note that Theorem 5.1 states in particular that MLP-kernels are invariant since they do not have spatial axes. While this kernel invariance is shared by many standard kernels, such as RBF or linear kernels, this property is non-trivial for NTK and NNGP since they are not simply functions of the norm of the difference or inner product of the two input values x and x'. Furthermore, this result holds irrespective of whether a group is of finite or infinite order.

Step 2: Data augmentation allows to rewrite the group action as a permutation (see (11)). For the Gram matrix, acting with π_g is equivalent to multiplication by a permutation matrix $\Pi(g)$. Combining this with the invariance of the MLP-kernels derived above, we can shift a permutation from the first to the second index of the Gram matrix, i.e., for MLPs,

$$\begin{aligned}\Pi(g)\Theta(\mathcal{X}, \mathcal{X}) &= \Theta(\rho_X(g)\mathcal{X}, \mathcal{X}) && (18)\\ &= \Theta(\mathcal{X}, \rho_X^{-1}(g)\mathcal{X}) && (19)\\ &= \Theta(\mathcal{X}, \mathcal{X})(\Pi^{-1}(g))^\top && (20)\\ &= \Theta(\mathcal{X}, \mathcal{X})\Pi(g) , && (21)\end{aligned}$$

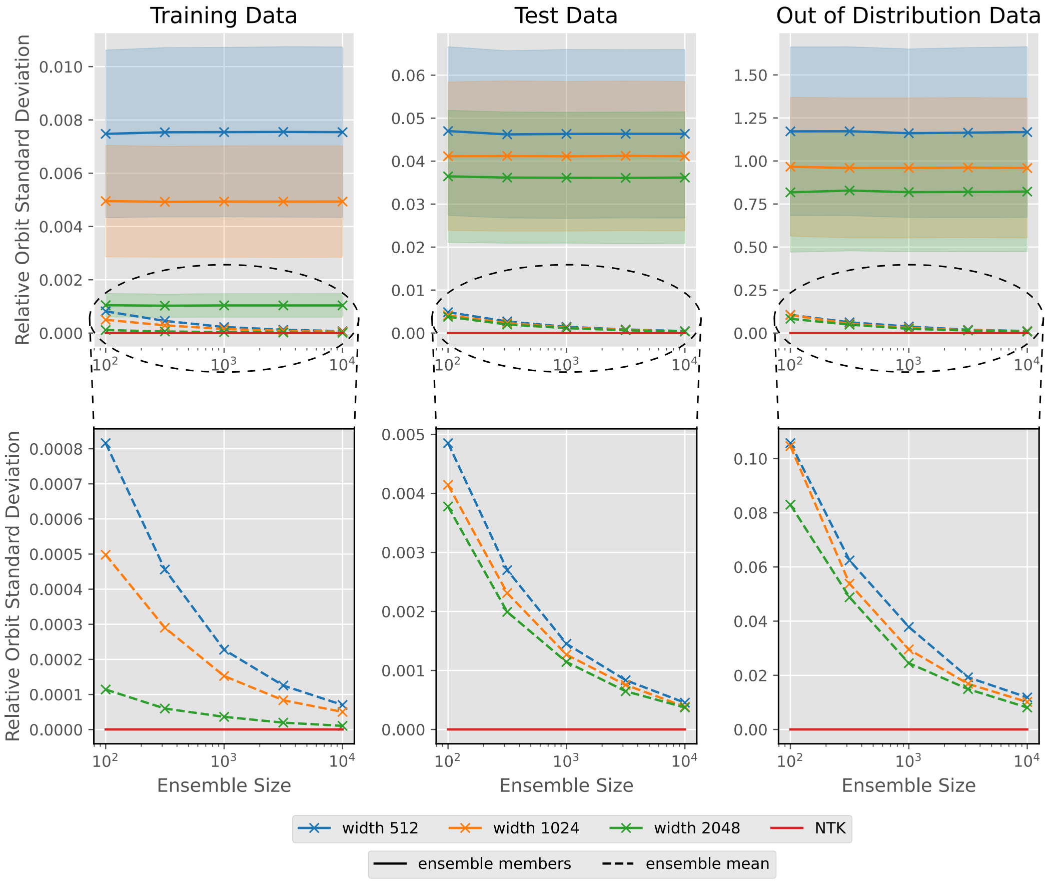

Figure 1. Invariance of predicted energies with respect to lattice rotations by 90°. Solid lines refer to predictions of individual ensemble members and their standard deviation, dashed lines refer to mean predictions of the ensemble. Zoom-ins in the second row show that the invariance of mean predictions converges to NTK invariance for large ensembles and network widths.

where we have used the Theorem 5.1 for the second equality. This property can be extended to more general architectures and analytical functions of the kernels as stated in the following lemma.

Lemma 5.2 (Shift of permutation)**.** *Data augmentation implies that the permutation group action Π commutes with any matrix-valued analytical function F involving the Gram matrices of the NNGP and NTK as well as their inverses:*

$$\begin{aligned} &\Pi(g)F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1}) \\ &= \rho_K(g) F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1}) \Pi(g) \rho_K^\top(g)\,. \end{aligned} \tag{22}$$

where $\Pi(g)$ denotes the group action in terms of training set permutations as a permutation matrix, see (11).

Proof. See Appendix B.

Step 3: Using Lemma 5.2, it can be shown that the deep ensemble is equivariant in the infinite width limit. Before stating the general theorem, we first illustrate the underlying reasoning by showing one particular consequence, i.e., that the mean of an MLP is invariant if the training labels $\mathcal{Y}$ are not transformed, i.e. $\rho_Y(g) = \mathbb{I}$. By (6), the output of the deep ensemble for transformed input $x \to \rho_X(g)x$ is given by

$$\begin{aligned} \bar{f}_t(\rho_X(g)\,x) &= \mu_t(\rho_X(g)\,x) && (23)\\ &= \Theta(\rho_X(g)\,x, \mathcal{X}) \left[\Theta^{-1}\,T_t\right] \mathcal{Y}\,. && (24) \end{aligned}$$

We can now use the invariance of MLP kernels from Theorem 5.1 and write the action of ρ_X on training samples $\mathcal{X}$ as a permutation Π. Together with $(\Pi^{-1})^\top = \Pi$, we obtain

$$\Theta(\rho_X(g)\,x, \mathcal{X}) \left[\Theta^{-1} T_t\right] \mathcal{Y} = \Theta(x, \mathcal{X}) \Pi(g) \left[\Theta^{-1} T_t\right] \mathcal{Y}\,.$$

Now we use Lemma 5.2 to commute the permutation past $\Theta^{-1}T_t$,

$$\Theta(x, \mathcal{X}) \Pi(g) \left[\Theta^{-1} T_t\right] \mathcal{Y} = \Theta(x, \mathcal{X}) \left[\Theta^{-1} T_t\right] \Pi(g) \mathcal{Y}\,.$$

Since the labels are invariant by assumption, $\Pi(g)\mathcal{Y} = \mathcal{Y}$ and therefore

$$\bar{f}_t(\rho_X(g)\,x) = \bar{f}_t(x)\,. \tag{25}$$

Using analogous reasoning, the following more general result can be derived:

Theorem 5.3 (Emergent Equivariance of Deep Ensembles)**.** *Under the assumptions stated above, the distribution of large-width ensemble members $f_w : X \to Y$ is equivariant with respect to the representations ρ_X and ρ_Y of the group G if data augmentation is applied. In particular, the ensemble is equivariant,*

$$\bar{f}_t(\rho_X(g)\,x) = \rho_Y(g)\,\bar{f}_t(x)\,, \tag{26}$$

for all $g \in G$. This result holds

1. *at any training time t,*
2. *for any element of the input space $x \in X$.*

Proof. See Appendix B.

We stress that this results holds even off the data manifold, i.e., for out-of-distribution data, and in the early stages of training as well as at initialization. As a result, it is not a trivial consequence of the training. Furthermore, we do not need to make any restrictions on the architectures of the ensemble members. In particular, the individual members will generically not be equivariant. However, their averaged prediction will be (at least in the large width limit). In this sense, the equivariance is emergent.

6. Limitations: Approximate Equivariance

In the following, we discuss the breaking of equivariance due to i) statistical fluctuations of the estimator due to the finite number of ensemble members, ii) continuous symmetry groups which do not allow for complete data augmentation, and iii) finite width corrections in NTK theory.

Finite Number of Ensemble Members. We derive the following bound for estimates of deep ensembles in the infinite width limit:

Lemma 6.1 (Bound for finite ensemble members)**.** *The deep ensemble $\bar{f}_t$ and its estimate $\hat{f}_t$ do not differ by more than threshold δ,*

$$|\bar{f}_t(x) - \hat{f}_t(x)| < \delta\,, \tag{27}$$

with probability $1 - \epsilon$ for ensemble sizes M that obey

$$M > -\frac{2\Sigma_t(x)}{\delta^2}\ln\left(\sqrt{\pi}\epsilon\right)\,. \tag{28}$$

We stress that the covariance Σ is known in closed form, see (7). As such, the right-hand-side can be calculated exactly. We note that we also derive a somewhat tighter bound in Appendix B which however necessitates to numerically solve for M.

Continuous Groups. For a continuous group G, consider a finite subgroup $A \subset G$ which is used for data augmentation. We quantify the discretization error of using A instead of G by

$$\epsilon = \max_{g\in G}\min_{g'\in A}\|\rho_X(g) - \rho_X(g')\|_{\mathrm{op}}\,. \tag{29}$$

Then, the invariance error of the mean (6) is bounded by ϵ:

Lemma 6.2 (Bound for continuous groups)**.** *Consider a deep ensemble of neural networks with Lipschitz continuous derivatives with respect to the parameters. For an approximation $A \subset G$ of a continuous symmetry group G with discretization error ϵ, the prediction of the ensemble trained on A deviates from invariance by*

$$|\bar{f}_t(x) - \bar{f}_t(\rho_X(g)\,x)| \le \epsilon\, C(x)\,, \qquad \forall g \in G\,,$$

where C is independent of g.

Random Augmentations. In practice, very often the augmentation is not performed over an entire subgroup A of the symmetry group as assumed in Lemma 6.2, but rather batches are augmented randomly. That is, A is not a subgroup, only a subset of G. In this case, the error (29) of using A rather than G for augmentation can be defined in terms of an expectation value over the distribution of the augmentations. The statement of Lemma 6.2 can then only be expected to hold in expectation. However, note that the solution of the training dynamics derived using NTKs in the infinite width limit assumes that the training set is the same in each epoch. Normally this assumption will be broken by random data augmentation. This effect cannot be controlled by Lemma 6.2.

Finite Width. Convergence of the ensemble output to a Gaussian distribution only holds in the infinite width limit. There has been substantial work on finite-width corrections to the NTK limit (Huang & Yau, 2020; Yaida, 2020; Halverson et al., 2021; Erbin et al., 2022) which could in principle be used to quantify the resulting violations of exact equivariance. This is however of significant technical difficulty and therefore beyond the scope of this work. In the experimental section, we nevertheless demonstrate that even finite-width ensembles show emergent equivariance to good approximation.

7. Experiments

In this section, we empirically study the emergent equivariance of finite width deep ensembles for several architectures (fully connected and convolutional), tasks (regression and classification), and application domains (computer vision and physics).

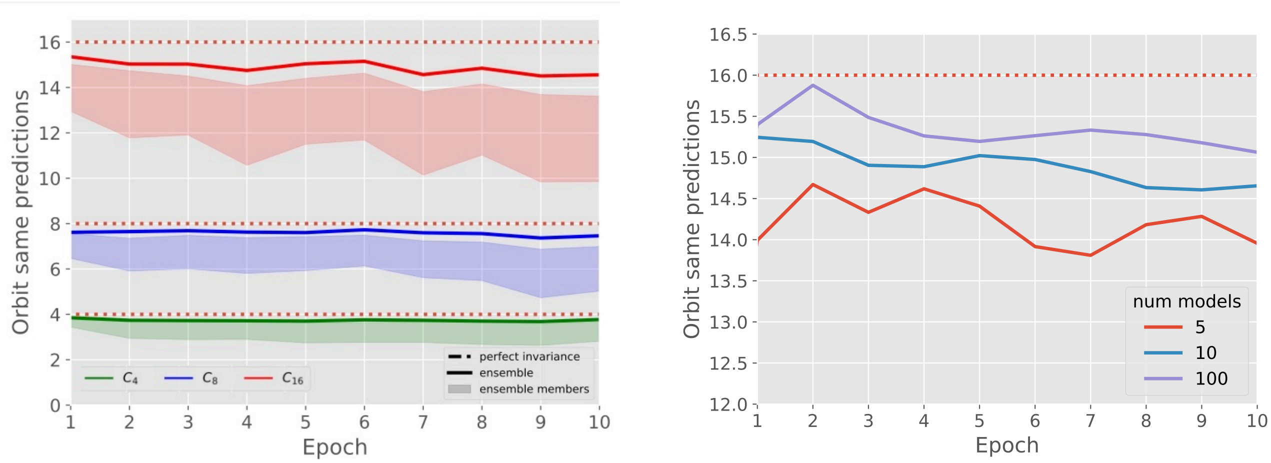

Figure 2. Emergent invariance for FashionMNIST **Left:** Number of out-of-distribution MNIST samples with the same prediction across a symmetry orbit for group orders 4 (green), 8 (blue), and 16 (red) versus training epoch. The models were trained on augmented FashionMNIST. Solid lines show the ensemble prediction. Shaded area is between the 25th and 75th quantile of the predictions of individual members of the ensemble. **Right:** Out of distribution invariance in the same setup as on the left-hand-side at group order 16. As the number of ensemble members increases, the prediction becomes more invariant, as expected.

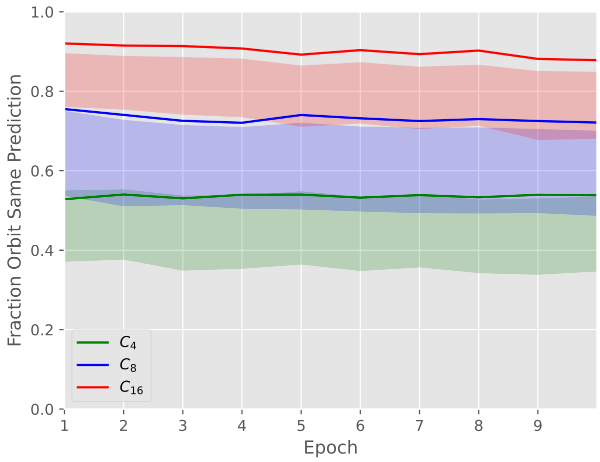

Figure 3. Equivariance extends to $SO(2)$ symmetry. Fraction of randomly sampled rotations that leave the prediction invariant is reported. Data augmentation with group order 4 (green), 8 (blue), 16 (red) is used. As expected, the equivariance increases with the group order.

Ising Model. We validate our analytical computations with experiments on a problem for which we can compute the NTK exactly: the two-dimensional Ising model on a 5x5 lattice with energy function $\mathcal{E} = -J\sum_{\langle i,j\rangle} s_i s_j$, with the spins $s_i \in \{+1, -1\}$, J a coupling constant and the sum runs over all adjacent spins. The energy of the Ising model is invariant under the cyclic group C_4 of rotations of the lattice by $90°$. We train ensembles of five different sizes with 100 to 10k members of fully-connected networks with hidden-layer widths 512, 1024 and 2048 to approximate the energy function using all rotations in C_4 as data augmentation. In this setting, we can compute the NTK exactly on the given training data using the JAX package `neural-tangents` (Novak et al., 2020). We verify that the ensembles converge to the NTK for large widths, see Appendix C.1.

To quantify the invariance of the ensembles, we measure the standard deviation of the predicted energy across the group orbit averaged over all datapoints of i) training set, ii) test set, and iii) out-of-distribution set. The latter is generated randomly drawing spins from a Gaussian distribution with mean zero and variance 400. For better interpretability, we divide by the mean of $\mathcal{E}$, so that for a *relative standard deviation (RSD)* across orbits of one, the deviation from invariance is as large as a typical ground truth energy. For an exactly equivariant model, we would obtain an RSD of zero.

Figure 1 shows that the deep ensemble indeed exhibit the expected emergent invariance. As expected, the NTK features very low RSD compatible with numerical error. The RSD of the mean predictions of the ensembles are larger but still very small and converge to the NTK results for large ensembles and network widths, cf. dashed lines in Figure 1. In contrast, the RSD computed for individual ensemble members is much higher and varies considerably between ensemble members, cf. solid lines in Figure 1. Even out of distribution, the ensemble means deviate from invariance only by about 0.8% for large ensembles and network widths, compared to 82% for individual ensemble members.

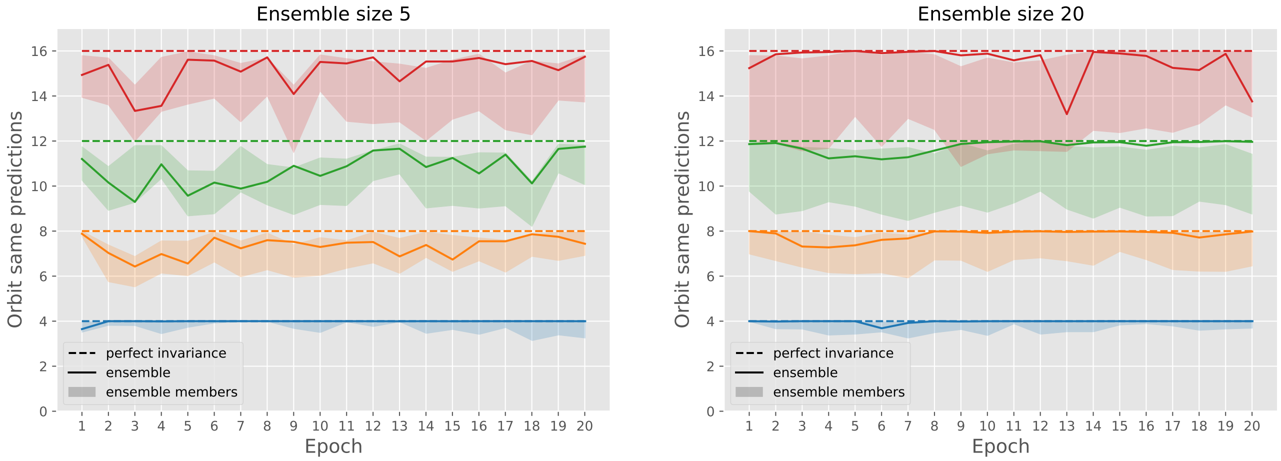

Figure 4. Ensemble invariance on OOD data for ensembles trained on histological data. Number of OOD samples with the same prediction across a symmetry orbit for group orders 4 (blue), 8 (orange), 12 (green) and 16 (red) versus training epoch. Even for ensemble size 5 (left), the ensemble predictions (solid line) are more invariant than the ensemble members (shaded region corresponding to 25th to 75th percentile of ensemble members). The effect is larger for ensemble size 20 (right).

	C_4	C_8	C_{16}
DeepEns+DA	3.85±0.12	**7.72±0.34**	**15.24±0.69**
only DA	3.41±0.18	6.73±0.24	12.77±0.71
E2CNN	**4±0.0**	**7.71±0.21**	**15.08±0.34**
Canon	**4±0.0**	**7.45±0.14**	12.41±0.85

Table 1. Deep Ensembles (ensemble size 100) show competitive degree of equivariance. Mean and standard deviation over training of orbit same prediction on the out-of-distribution MNIST validation set. All methods use roughly the same number of parameters. Best methods, taking into account statistical uncertainty, are shown in bold.

FashionMNIST. We train convolutional neural networks augmenting the original dataset (Xiao et al., 2017) by all elements of the group orbit of the cyclic group C_k, i.e., all rotations of the image by any multiple of $360/k$ degrees with $k = 4, 8, 16$ and choose ensembles of size $M = 5, 10, 100$. We then evaluate the *orbit same prediction (OSP)*, i.e., how many of the images in a given group orbit have on average the same classification result as the unrotated image. We evaluate the OSP metric both on the validation set of FashionMNIST as well as on various out-of-distribution (OOD) datasets. Specifically, we choose the validation sets of MNIST, grey-scaled CIFAR10 as well as images for which each pixel is drawn iid from $\mathcal{N}(0, 1)$. Figure 2 shows the OSP metric for OOD data from MNIST. The ensemble prediction becomes more invariant as the number of ensemble members increases. Furthermore, the ensemble prediction is significantly more invariant as the individual ensemble members, i.e., the invariance is emergent. As the group order k increases, more ensemble members are needed to achieve a high degree of invariance. Figure 3 illustrates that the deep ensembles can also capture continuous rotation symmetry. Specifically, we train using data augmentation with respect to various discrete C_k groups and check that they lead to increasing invariance with respect to $SO(2)$. For $k = 16$, over 90 percent of the orbit elements have the same prediction as the untransformed input establishing that the model is approximately invariant under the continuous symmetry as well, as is expected from Lemma 6.2. We also compare to a manifestly equivariant E2CNN (Cesa et al., 2022; Weiler & Cesa, 2019) and canonicalized model (Kaba et al., 2023). Interestingly, the manifest equivariance of these models is slightly broken for groups C_k with group order $k > 4$ due to interpolation artifacts. As result, finite deep ensembles are competitive with these manifestly equivariant models, see Table 1. Note also that using data augmentation without any ensembling leads to significantly less equivariant models. More details about the experiments as well as plots showing results for the other OOD datasets can be found in Appendix C.2.

Histological Data. A realistic task, where rotational invariance is of key importance, is the classification of histological slices. We trained ensembles of CNNs on the NCT-CRC-HE-100K dataset (Kather et al., 2018) which comprises of stained histological images of human colorectal cancer and normal tissue with a resolution of 224 × 224 pixels in nine classes.

As for our experiments on FashionMNIST, we verify that the ensemble is more invariant as a function of its input than the ensemble members by evaluating the OSP on OOD data. In order to arrive at a sample of OOD data on which the network makes non-constant predictions, we optimize the input of the untrained ensemble to yield balanced predictions of high confidence. Using this specifically generated dataset for each ensemble, we observe the same increase in invariance also outside of the training domain as predicted by our theoretical considerations, cf. Figure 4. For further results on validation data as well as examples of our OOD data see Appendix E

8. Conclusions

Equivariant neural networks are a central ingredient in many machine learning setups, in particular in the natural sciences. However, constructing manifestly invariant models can be difficult. Deep ensembles are an important tool which can straightforwardly boost the performance and estimate uncertainty of existing models, explaining their widespread use in practice.

In this work, using the theory of neural tangent kernels, we proved that infinitely wide ensembles show emergent equivariance when trained on augmented data. We furthermore discussed implications of finite width and ensemble size as well as the effect of approximating a continuous symmetry group. Experiments on several different datasets support our theoretical insights.

The extension of our proof to additional layers like attention, pooling or dropout is straightforward. In future work, it would be interesting to incorporate the effects of finite width corrections and include a more detailed model of data augmentation, for instance along the lines of Dao et al. (2019).

Impact statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

The work of J.G. is supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. P.K. wants to thank Shin Nakajima and Andreas Loukas for discussions, J.G. wants to thank Max Guillen and Philipp Misof for discussions.

References

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On Exact Computation with an Infinitely Wide Neural Net. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Basu, S., Katdare, P., Sattigeri, P., Chenthamarakshan, V., Driggs-Campbell, K., Das, P., and Varshney, L. R. Equivariant few-shot learning from pretrained models. *arXiv preprint arXiv:2305.09900*, 2023a.

Basu, S., Sattigeri, P., Ramamurthy, K. N., Chenthamarakshan, V., Varshney, K. R., Varshney, L. R., and Das, P. Equi-tuning: Group equivariant fine-tuning of pretrained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6788–6796, 2023b.

Bekkers, E. J., Lafarge, M. W., Veta, M., Eppenhof, K. A. J., Pluim, J. P. W., and Duits, R. Roto-Translation Covariant Convolutional Networks for Medical Image Analysis. In Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C., and Fichtinger, G. (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture Notes in Computer Science, pp. 440–448, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00928-1. doi: 10.1007/978-3-030-00928-1_50.

Cesa, G., Lang, L., and Weiler, M. A program to build E(N)-equivariant steerable CNNs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WE4qe9xlnQw.

Dao, T., Gu, A., Ratner, A., Smith, V., Sa, C. D., and Re, C. A Kernel Theory of Modern Data Augmentation. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 1528–1537. PMLR, May 2019.

Du, S. S., Hou, K., Póczos, B., Salakhutdinov, R., Wang, R., and Xu, K. Graph Neural Tangent Kernel: Fusing Graph Neural Networks with Graph Kernels. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., November 2019.

Elesedy, B. and Zaidi, S. Provably Strict Generalisation Benefit for Equivariant Models. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2959–2969. PMLR, July 2021.

Erbin, H., Lahoche, V., and Samary, D. O. Nonperturbative renormalization for the neural network-QFT correspondence. *Machine Learning: Science and Technology*, 3(1):015027, March 2022. ISSN 2632-2153. doi: 10.1088/2632-2153/ac4f69.

Flinth, A. and Ohlsson, F. Optimization Dynamics of Equivariant and Augmented Neural Networks. (arXiv:2303.13458), March 2023. doi: 10.48550/arXiv.2303.13458.

Franceschi, J.-Y., Bézenac, E. D., Ayed, I., Chen, M., Lamprier, S., and Gallinari, P. A Neural Tangent Kernel Perspective of GANs. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 6660–6704. PMLR, June 2022. doi: 10.48550/arXiv.2106.05566.

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., and Suganthan, P. N. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, October 2022. ISSN 09521976. doi: 10.1016/j.engappai.2022.105151.

Gerken, J. E., Carlsson, O., Linander, H., Ohlsson, F., Petersson, C., and Persson, D. Equivariance versus Augmentation for Spherical Images. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 7404–7421. PMLR, June 2022. doi: 10.48550/arXiv.2202.03990.

Gerken, J. E., Aronsson, J., Carlsson, O., Linander, H., Ohlsson, F., Petersson, C., and Persson, D. Geometric deep learning and equivariant neural networks. *Artificial Intelligence Review*, June 2023. ISSN 1573-7462. doi: 10.1007/s10462-023-10502-7.

Golikov, E., Pokonechnyy, E., and Korviakov, V. Neural Tangent Kernel: A Survey. (arXiv:2208.13614), August 2022. doi: 10.48550/arXiv.2208.13614.

Halverson, J., Maiti, A., and Stoner, K. Neural Networks and Quantum Field Theory. *Machine Learning: Science and Technology*, 2(3):035002, September 2021. ISSN 2632-2153. doi: 10.1088/2632-2153/abeca3.

Hayase, J. and Oh, S. Few-shot Backdoor Attacks via Neural Tangent Kernels. In *The Eleventh International Conference on Learning Representations*, September 2022. doi: 10.48550/arXiv.2210.05929.

Huang, J. and Yau, H.-T. Dynamics of Deep Neural Networks and Neural Tangent Hierarchy. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 4542–4551. PMLR, November 2020. doi: 10.48550/arXiv.1909.08156.

Jacot, A., Gabriel, F., and Hongler, C. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Kaba, S.-O., Mondal, A. K., Zhang, Y., Bengio, Y., and Ravanbakhsh, S. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pp. 15546–15566. PMLR, 2023.

Kather, J. N., Halama, N., and Marx, A. 100,000 histological images of human colorectal cancer and healthy tissue. April 2018. doi: 10.5281/zenodo.1214456.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep Neural Networks as Gaussian Processes. In *International Conference on Learning Representations*, February 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. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. doi: 10.1088/1742-5468/abc62b.

Li, Z., Wang, R., Yu, D., Du, S. S., Hu, W., Salakhutdinov, R., and Arora, S. Enhanced Convolutional Neural Tangent Kernels. *arXiv:1911.00809 [cs, stat]*, November 2019.

Mei, S., Misiakiewicz, T., and Montanari, A. Learning with invariances in random features and kernel models. In *Proceedings of Thirty Fourth Conference on Learning Theory*, pp. 3351–3418. PMLR, July 2021.

Mondal, A. K., Panigrahi, S. S., Kaba, S.-O., Rajeswar, S., and Ravanbakhsh, S. Equivariant adaptation of large pre-trained models. *arXiv preprint arXiv:2310.01647*, 2023.

Mroueh, Y., Voinea, S., and Poggio, T. A. Learning with Group Invariant Features: A Kernel Perspective. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Neal, R. M. *Bayesian Learning for Neural Networks*. Springer Science & Business Media, 1996. ISBN 978-1-4612-0745-0.

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 *Eighth International Conference on Learning Representations*, April 2020.

Puny, O., Atzmon, M., Ben-Hamu, H., Misra, I., Grover, A., Smith, E. J., and Lipman, Y. Frame averaging for invariant and equivariant network design. *arXiv preprint arXiv:2110.03336*, 2021.

Raj, A., Kumar, A., Mroueh, Y., Fletcher, T., and Schoelkopf, B. Local Group Invariant Representations via Orbit Embeddings. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1225–1235. PMLR, April 2017.

Ruffolo, J. A., Chu, L.-S., Mahajan, S. P., and Gray, J. J. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature communications*, 14 (1):2389, 2023.

Saib, W., Sengeh, D., Dlamini, G., and Singh, E. Hierarchical deep learning ensemble to automate the classification of breast cancer pathology reports by icd-o topography. *arXiv preprint arXiv:2008.12571*, 2020.

Sannai, A., Imaizumi, M., and Kawano, M. Improved generalization bounds of group invariant / equivariant deep networks via quotient feature spaces. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 771–780. PMLR, December 2021. doi: 10.48550/arXiv.1910.06552.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv:1802.08219 [cs]*, May 2018.

Wang, S., Yu, X., and Perdikaris, P. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, January 2022. ISSN 0021-9991. doi: 10.1016/j.jcp.2021.110768.

Weiler, M. and Cesa, G. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. URL `https://arxiv.org/abs/1911.08251`.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Yaida, S. Non-Gaussian processes and neural networks at finite widths. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, pp. 165–192. PMLR, August 2020.

Yang, G. Scaling Limits of Wide Neural Networks with Weight Sharing: Gaussian Process Behavior, Gradient Independence, and Neural Tangent Kernel Derivation. *arXiv:1902.04760 [cond-mat, physics:math-ph, stat]*, April 2020.

Yang, G. and Hu, E. J. Feature Learning in Infinite-Width Neural Networks. (arXiv:2011.14522), July 2022. doi: 10.48550/arXiv.2011.14522.

Yang, H. and Wang, Z. On the Neural Tangent Kernel Analysis of Randomly Pruned Neural Networks. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pp. 1513–1553. PMLR, April 2023.

A. Introduction to neural tangent kernels

In this appendix, we will give a brief review of the theory of neural tangent kernels (NTKs) for readers who are not familiar with it. For a more comprehensive review, see e.g. Golikov et al. (2022).

A.1. The empirical NTK

To understand how the NTK arises in the training dynamics of neural networks, consider a neural network $f_w : X \to \mathbb{R}$ with parameters w. Under continuous gradient descent, the parameters are updated according to

$$\frac{\partial w}{\partial t} = -\eta \frac{\partial \mathcal{L}(f_w(\mathcal{X}), \mathcal{Y})}{\partial w}\,, \tag{30}$$

where t is the training time, η is the learning rate and $\mathcal{L}(f_w(\mathcal{X}), \mathcal{Y})$ is the loss function which depends on the training set predictions $f_w(\mathcal{X})$ and the training labels $\mathcal{Y}$. Since $\mathcal{L}$ depends on w only through $f_w(\mathcal{X})$, we can use the chain-rule to rewrite (30),

$$\frac{\partial w}{\partial t} = -\eta \sum_{i=1}^{|\mathcal{T}|} \frac{\partial f_w(x_i)}{\partial w} \frac{\partial \mathcal{L}(f_w(\mathcal{X}), \mathcal{Y})}{\partial f_w(x_i)}\,. \tag{31}$$

Similarly, f_w depends on t only through w, so we obtain

$$\frac{\partial f_w(x)}{\partial t} = \left(\frac{\partial f_w(x)}{\partial w}\right)^\top \frac{\partial w}{\partial t}\,. \tag{32}$$

And hence, using (31),

$$\frac{\partial f_w(x)}{\partial t} = -\eta \sum_{i=1}^{|\mathcal{T}|} \left(\frac{\partial f_w(x)}{\partial w}\right)^\top \frac{\partial f_w(x_i)}{\partial w} \frac{\partial \mathcal{L}(f_w(\mathcal{X}), \mathcal{Y})}{\partial f_w(x_i)} = -\eta \sum_{i=1}^{|\mathcal{T}|} \Theta^w(x, x_i) \frac{\partial \mathcal{L}(f_w(\mathcal{X}), \mathcal{Y})}{\partial f_w(x_i)}\,, \tag{33}$$

where we have introduced the *empirical neural tangent kernel*

$$\Theta^w(x, x') = \left(\frac{\partial f_w(x)}{\partial w}\right)^\top \frac{\partial f_w(x')}{\partial w}\,. \tag{34}$$

This quantity depends on the parameters and hence evolves during training. As can be seen in (33), it is the NTK which induces the complicated non-linear evolution in the training dynamics since the derivative of the loss with respect to the predictions is independent of the parameters. Since it depends on the parameters, we can think of the empirical NTK at initialization as a random variable over the initialization distribution.

A.2. Infinite width limit

At infinite width, the dynamics (33) simplify dramatically. Firstly, it is known for a long time that the preactivations at initialization become a mean-zero Gaussian process (GP) as the width of the layers tend to infinity (Neal, 1996). The covariance function of this GP is known as the *neural network gaussian process (NNGP) kernel* $\mathcal{K}(x, x')$. Therefore, in particular

$$f_{w_0}(x) \sim \mathcal{N}(0, \mathcal{K}(x, x)) \qquad \forall x \in X \tag{35}$$

at initialization. The NNGP kernel can be computed recursively layer-by-layer with the recursion relations given in the proof of Theorem 5.1 in Appendix B below.

Secondly, it was realized more recently (Jacot et al., 2018) that in the infinite width limit, the empirical NTK (34) converges in probability to its expectation value and therefore becomes a deterministic quantity

$$\Theta^w(x, x') \xrightarrow{\text{width}\to\infty} \mathbb{E}_w\left[\left(\frac{\partial f_w(x)}{\partial w}\right)^\top \frac{\partial f_w(x')}{\partial w}\right] = \Theta(x, x')\,, \tag{36}$$

which is the definition of the NTK we used above in (3). This limiting quantity can be computed again recursively layer-by-layer.

In (Jacot et al., 2018), the authors introduced a slightly different parametrization of neural network layers. For fully connected layers of input width n, instead of using

$$z^{(\ell)}(x) = W f^{(\ell)}(x) \qquad \text{with} \qquad W_{ij} \sim \mathcal{N}\Big(0, \frac{1}{n}\Big) \tag{37}$$

they suggest to use

$$z^{(\ell)}(x) = \frac{1}{\sqrt{n}} W f^{(\ell)}(x) \qquad \text{with} \qquad W_{ij} \sim \mathcal{N}(0,1)\,. \tag{38}$$

Note that at initialization, the distribution of $z^{(\ell)}(x)$ is the same in both parametrizations. During training, the derivatives with respect to W will however be rescaled in (38). An important result of (Jacot et al., 2018) was that under mild assumptions on the nonlinearity, in the infinite width limit the NTK does not only become a deterministic, but also constant throughout training when using the parametrization (38).

A.3. Training dynamics

In the parametrization (38), the training dynamics (33) therefore simplify dramatically when taking the layer widths to infinity. For the MSE loss, (33) becomes

$$\frac{\partial f_t(x)}{\partial t} = -\eta \sum_{i=1}^{|\mathcal{T}|} \Theta(x, x_i)(f_t(x_i) - y_i) = -\eta\, \Theta(x, \mathcal{X})(f_t(\mathcal{X}) - \mathcal{Y})\,, \tag{39}$$

where we have introduced the notation f_t for the neural network with parameters from training time t, Θ is the deterministic and constant NTK (36) and we have used the compact notation for summations over the training data employed in many places of this paper.

The differential equation (39) can be solved analytically in two steps. Since the right-hand side depends on f_t evaluated on the training set, whereas the left-hand side depends on f_t evaluated at an arbitrary point, we first solve (39) evaluated on the training set,

$$\frac{\partial f_t(\mathcal{X})}{\partial t} = -\eta\, \Theta(\mathcal{X}, \mathcal{X})(f_t(\mathcal{X}) - \mathcal{Y})\,. \tag{40}$$

The solution to this equation is, in terms of the initial training-set predictions $f_0(\mathcal{X})$, given by

$$f_t(\mathcal{X}) = e^{-\eta\Theta(\mathcal{X},\mathcal{X})t}(f_0(\mathcal{X}) - \mathcal{Y}) + \mathcal{Y}\,. \tag{41}$$

Next, we can plug this solution back into (39) to obtain an equation for $f_t(x)$,

$$\frac{\partial f_t(x)}{\partial t} = -\eta\, \Theta(x, \mathcal{X}) e^{-\eta\Theta(\mathcal{X},\mathcal{X})t}(f_0(\mathcal{X}) - \mathcal{Y})\,. \tag{42}$$

The right-hand side depends on t only through the exponential factor. Integration therefore yields a solution for $f_t(x)$ which we write in terms of the initial prediction $f_0(x)$

$$f_t(x) = \Theta(x, \mathcal{X})\Theta(\mathcal{X}, \mathcal{X})^{-1}(e^{-\eta\Theta(\mathcal{X},\mathcal{X})t} - \mathbb{I})(f_0(\mathcal{X}) - \mathcal{Y}) + f_0(x)\,. \tag{43}$$

This completely solves the training dynamics and we can predict the output at an arbitrary test point after an arbitrary amount of training time.

The prediction at time t is still a random variable of the initialization distribution. However, the initialization only enters via f_0 which in the infinite width limit is a GP as noted above. Therefore, f_t is as a linear combination of GPs itself a GP. Since the mean function of f_0 is identically zero (35), it is straightforward to compute the mean function μ_t of f_t

$$\mu_t(x) = \mathbb{E}[f_t(x)] = \Theta(x, \mathcal{X})\Theta(\mathcal{X}, \mathcal{X})^{-1}(\mathbb{I} - e^{-\eta\Theta(\mathcal{X},\mathcal{X})t})\mathcal{Y}\,. \tag{44}$$

This is just the expression given above in (6). Similarly, one can compute the covariance function Σ_t of f_t

$$\begin{aligned}
\Sigma_t(x, x') &= \mathbb{E}[(f_t(x) - \mu_t(x))(f_t(x') - \mu_t(x'))] && (45)\\
&= \mathbb{E}\Big[\Big(\Theta(x, \mathcal{X})\Theta(\mathcal{X}, \mathcal{X})^{-1}(e^{-\eta\Theta(\mathcal{X},\mathcal{X})t} - \mathbb{I})f_0(\mathcal{X}) + f_0(x)\Big)\\
&\quad \times \Big(f_0(\mathcal{X})^\top (e^{-\eta\Theta(\mathcal{X},\mathcal{X})t} - \mathbb{I})\Theta(\mathcal{X}, \mathcal{X})^{-1}\Theta(\mathcal{X}, x') + f_0(x')\Big)\Big] && (46)\\
&= \mathcal{K}(x, x') - \mathcal{K}(x, \mathcal{X})\Theta(\mathcal{X}, \mathcal{X})^{-1}(\mathbb{I} - e^{-\eta\Theta(\mathcal{X},\mathcal{X})t})\Theta(\mathcal{X}, x')\\
&\quad - \Theta(x, \mathcal{X})\Theta(\mathcal{X}, \mathcal{X})^{-1}(\mathbb{I} - e^{-\eta\Theta(\mathcal{X},\mathcal{X})t})\mathcal{K}(\mathcal{X}, x')\\
&\quad + \Theta(x, \mathcal{X})\Theta(\mathcal{X}, \mathcal{X})^{-1}(\mathbb{I} - e^{-\eta\Theta(\mathcal{X},\mathcal{X})t})\mathcal{K}(\mathcal{X}, \mathcal{X})(\mathbb{I} - e^{-\eta\Theta(\mathcal{X},\mathcal{X})t})\Theta(\mathcal{X}, \mathcal{X})^{-1}\Theta(\mathcal{X}, x')\,, && (47)
\end{aligned}$$

where we have used that the expectation value of f_0 vanishes and that the covariance function of f_0 is the NNGP $\mathcal{K}$. The final expression is (7) from above. Since the predictions on arbitary test points are a GP, by providing explicit expressions for the mean- and convariance functions, we have determined the statistics of the predictions entirely.

B. Proofs

Theorem 5.1 (Kernel transformation)**.** *Let G be a group and ρ_X a representation of G acting on the input space X as in (12). Then, the neural tangent kernel Θ, as defined in (3), as well as the NNGP kernel $\mathcal{K}$, as defined in (8), of a neural network satisfying the assumptions above transform according to*

$$\Theta(\rho_X(g)x, \rho_X(g)x') = \rho_K(g)\Theta(x, x')\rho_K^\top(g)\,, \tag{16}$$

$$\mathcal{K}(\rho_X(g)x, \rho_X(g)x') = \rho_K(g)\mathcal{K}(x, x')\rho_K^\top(g)\,, \tag{17}$$

for all $g \in G$ and $x, x' \in X$, where ρ_K is a transformation acting on the spatial dimensions of the kernels according to $\rho_K(g)K^a = K^{\rho^{-1}(g)a}$. If the kernels do not have spatial axes, $\rho_K = \mathbb{I}$.

Proof. We will prove the transformation properties by induction over the layer, using the forward equations for the kernels of the different layer types considered. In this prescription, both NNGP and NTK are defined recursively per layer. In the first layer, they are simply given by

$$\mathcal{K}_1^{a,a'}(x, x') = x_a^\top x'_{a'} \tag{48}$$

$$\Theta_1^{a,a'}(x, x') = \mathcal{K}_1^{a,a'}(x, x')\,, \tag{49}$$

where a and a' both denote a spatial axis, e.g. in two dimensions, a is a multi-index $a = (h, w)$ and for graphs, a is the node index. The a, a' axes can be absent for inputs without spatial axes.

The forward equations which account for the nonlinearities are given by

$$\Lambda_\ell^{a,a'}(x, x') = \begin{pmatrix} \mathcal{K}_\ell^{a,a}(x, x) & \mathcal{K}_\ell^{a,a'}(x, x') \\ \mathcal{K}_\ell^{a',a}(x', x) & \mathcal{K}_\ell^{a',a'}(x', x') \end{pmatrix} \tag{50}$$

$$\mathcal{K}_\ell^{a,a'}(x, x') = \mathbb{E}_{(u,v)\sim\mathcal{N}(0,\Lambda_{\ell-1}^{a,a'}(x,x'))}[\sigma(u)\sigma(v)] \tag{51}$$

$$\dot{\mathcal{K}}_\ell^{a,a'}(x, x') = \mathbb{E}_{(u,v)\sim\mathcal{N}(0,\Lambda_{\ell-1}^{a,a'}(x,x'))}[\sigma'(u)\sigma'(v)] \tag{52}$$

$$\Theta_\ell^{a,a'}(x, x') = \dot{\mathcal{K}}_\ell^{a,a'}(x, x')\Theta_{\ell-1}^{a,a'}(x, x')\,, \tag{53}$$

where σ is the nonlinearity and σ' its derivative. Note that throughout, we drop numerical prefactors which depend on the prefactors in the layer definitions and initialization variances and are irrelevant for our argument.

For fully connected layers, the forward equation for the kernels is given by

$$\mathcal{K}_\ell(x, x') = \mathcal{K}_{\ell-1}(x, x') \tag{54}$$

$$\Theta_\ell(x, x') = \mathcal{K}_\ell(x, x') + \Theta_{\ell-1}(x, x')\,. \tag{55}$$

For convolutional layers, the forward equation for the kernels is given by (Arora et al., 2019)

$$\mathcal{K}_\ell^{a,a'}(x, x') = \sum_{\tilde{a}} \mathcal{K}_{\ell-1}^{a+\tilde{a},a'+\tilde{a}}(x, x') \tag{56}$$

$$\Theta_\ell^{a,a'}(x, x') = \mathcal{K}_\ell^{a,a'}(x, x') + \sum_{\tilde{a}} \Theta_{\ell-1}^{a+\tilde{a},a'+\tilde{a}}(x, x')\,. \tag{57}$$

For flattening layers, the forward equation for the kernels is given by (Novak et al., 2020)

$$\mathcal{K}_\ell^{a,a'}(x, x') = \sum_{\tilde{a}} \mathcal{K}_{\ell-1}^{\tilde{a},\tilde{a}}(x, x') \tag{58}$$

$$\Theta_\ell(x, x') = \mathcal{K}_\ell^{a,a'}(x, x') + \sum_{\tilde{a}} \Theta_{\ell-1}^{\tilde{a},\tilde{a}}(x, x')\,. \tag{59}$$

In graph neural networks we consider graphs with node features $x_a \in \mathbb{R}^n$ at node a. In a local aggregation layer, we sum the node features over a neighborhood $\mathcal{N}(a)$ of a,

$$z_\ell^a(x) = \sum_{\tilde{a}\in\mathcal{N}(a)} z_{\ell-1}^{\tilde{a}}(x)\,. \tag{60}$$

For local aggregation layers, the forward equations are given by (Du et al., 2019)

$$\mathcal{K}_\ell^{a,a'}(x,x') = \sum_{\tilde{a}\in\mathcal{N}(a)}\sum_{\tilde{a}'\in\mathcal{N}(a')}\mathcal{K}_{\ell-1}^{\tilde{a},\tilde{a}'}(x,x') \tag{61}$$

$$\Theta_\ell^{a,a'}(x,x') = \mathcal{K}_\ell^{a,a'}(x,x') + \sum_{\tilde{a}\in\mathcal{N}(a)}\sum_{\tilde{a}'\in\mathcal{N}(a')}\Theta_{\ell-1}^{\tilde{a},\tilde{a}'}(x,x')\,. \tag{62}$$

In a global aggregation layer, we sum over the entire graph instead. Correspondingly, for global aggregation layers, the sums in (61) and (62) run over the entire node set.

The kernels for the entire network are then given by the kernels of the last layer L,

$$\mathcal{K}^{a,a'}(x,x') = \mathcal{K}_L^{a,a'}(x,x') \qquad \text{and} \qquad \Theta^{a,a'}(x,x') = \Theta_L^{a,a'}(x,x')\,. \tag{63}$$

The presence or absence of spatial indices of the kernels depends on if the final network layer has spatial dimensions or not. If additional channels are present in the final layer (as e.g. in multi-class classification), the NTK is proportional to the unit matrix in those dimensions (Jacot et al., 2018). The fully general NTK therefore has the structure

$$\Theta^{\alpha a,\alpha' a'}(x,x') = \Theta^{a,a'}(x,x')\delta^{\alpha\alpha'}\,, \tag{64}$$

where δ is the Kronecker symbol.

Base case Using the definition (12) of ρ_X, the NNGP is equivariant by orthogonality of τ_X,

$$\begin{aligned}
\mathcal{K}_1^{a,a'}(\rho_X(g)x,\rho_X(g)x') &= x^\top_{\rho(g)^{-1}a}\tau_X^\top(g)\tau_X(g)x'_{\rho(g)^{-1}a'} = x^\top_{\rho(g)^{-1}a}x'_{\rho(g)^{-1}a'} &\quad (65)\\
&= \mathcal{K}_1^{\rho(g)^{-1}a,\rho(g)^{-1}a'}(x,x') = (\rho_K(g)\mathcal{K}_1(x,x')\rho_K^\top(g))^{a,a'}\,. &\quad (66)
\end{aligned}$$

In the first layer, NNGP and NTK are equal according to (49), so also the NTK transforms as (66).

Induction step Assume that

$$\begin{aligned}
\mathcal{K}_{\ell-1}(\rho_X(g)x,\rho_X(g)x') &= \rho_K(g)\mathcal{K}_{\ell-1}(x,x')\rho_K^\top(g) &\quad (67)\\
\Theta_{\ell-1}(\rho_X(g)x,\rho_X(g)x') &= \rho_K(g)\Theta_{\ell-1}(x,x')\rho_K^\top(g)\,. &\quad (68)
\end{aligned}$$

with ρ_K trivial if no spatial indices are present in layer $\ell-1$. For the nonlinearities, we have

$$\begin{aligned}
\mathcal{K}_\ell^{a,a'}(\rho_X(g)x,\rho_X(g)x') &= \mathbb{E}_{(u,v)\sim\mathcal{N}(0,\Lambda_{\ell-1}^{a,a'}(\rho_X(g)x,\rho_X(g)x'))}[\sigma(u)\sigma(v)] &\quad (69)\\
&= \mathbb{E}_{(u,v)\sim\mathcal{N}(0,\Lambda_{\ell-1}^{\rho^{-1}(g)a,\rho^{-1}(g)a'}(x,x'))}[\sigma(u)\sigma(v)] &\quad (70)\\
&= \mathcal{K}_\ell^{\rho^{-1}(g)a,\rho^{-1}(g)a'}(x,x') &\quad (71)\\
&= (\rho_K(g)\mathcal{K}_\ell(x,x')\rho_K^\top(g))^{a,a'} &\quad (72)
\end{aligned}$$

and similarly for $\dot{\mathcal{K}}_\ell$. For the NTK, we have

$$\begin{aligned}
\Theta_\ell^{a,a'}(\rho_X(g)x,\rho_X(g)x') &= \dot{\mathcal{K}}_\ell^{a,a'}(\rho_X(g)x,\rho_X(g)x')\Theta_{\ell-1}^{a,a'}(\rho_X(g)x,\rho_X(g)x') &\quad (73)\\
&= (\rho_K(g)\dot{\mathcal{K}}_\ell(x,x')\rho_K^\top(g))^{a,a'}(\rho_K(g)\Theta_{\ell-1}(x,x')\rho_K^\top(g))^{a,a'} &\quad (74)\\
&= (\rho_K(g)\dot{\mathcal{K}}_\ell(x,x')\Theta_{\ell-1}(x,x')\rho_K^\top(g))^{a,a'} &\quad (75)\\
&= (\rho_K(g)\Theta_\ell^{a,a'}(x,x')\rho_K^\top(g))^{a,a'}\,. &\quad (76)
\end{aligned}$$

For fully connected layers, the induction steps for $\mathcal{K}_\ell$ and Θ_ℓ are implied immediately by (54) and (55) and the induction assumptions.

For convolutional layers, the induction step for $\mathcal{K}_\ell$ is given by

$$\begin{aligned}
\mathcal{K}_\ell^{a,a'}(\rho_X(g)x,\rho_X(g)x') &= \sum_{\tilde{a}}\mathcal{K}_{\ell-1}^{a+\tilde{a},a'+\tilde{a}}(\rho_X(g)x,\rho_X(g)x') &\quad (77)\\
&= \sum_{\tilde{a}}\mathcal{K}_{\ell-1}^{\rho^{-1}(g)(a+\tilde{a}),\rho^{-1}(g)(a'+\tilde{a})}(x,x') &\quad (78)\\
&= \sum_{\tilde{a}}\mathcal{K}_{\ell-1}^{\rho^{-1}(g)a+\tilde{a},\rho^{-1}(g)a'+\tilde{a}}(x,x') &\quad (79)\\
&= \mathcal{K}_\ell^{\rho^{-1}(g)a,\rho^{-1}(g)a'}(x,x') &\quad (80)\\
&= (\rho_K(g)\mathcal{K}_\ell(x,x')\rho_K^\top(g))^{a,a'}\,. &\quad (81)
\end{aligned}$$

The induction step for the NTK in convolutional layers proceeds along the same lines.

For flattening layers, the induction step for $\mathcal{K}_\ell$ is given by

$$\begin{aligned}\mathcal{K}_\ell(\rho_X(g)x,\rho_X(g)x') &= \sum_{\tilde{a}} \mathcal{K}_\ell^{\tilde{a},\tilde{a}}(\rho_X(g)x,\rho_X(g)x') \qquad (82)\\ &= \sum_{\tilde{a}} \mathcal{K}_\ell^{\rho^{-1}(g)\tilde{a},\rho^{-1}(g)\tilde{a}}(x,x') \qquad (83)\\ &= \sum_{\tilde{a}} \mathcal{K}_\ell^{\tilde{a},\tilde{a}}(x,x') = \mathcal{K}_\ell(x,x')\,. \qquad (84)\end{aligned}$$

The induction step for the NTK of flattening layers proceeds along the same lines.

For local aggregation layers in graph neural networks, the induction step for $\mathcal{K}_\ell$ is given by

$$\begin{aligned}\mathcal{K}_\ell^{a,a'}(\rho_X(g)x,\rho_X(g)x') &= \sum_{\tilde{a}\in\mathcal{N}(a)}\sum_{\tilde{a}'\in\mathcal{N}(a')} \mathcal{K}_{\ell-1}^{\tilde{a},\tilde{a}'}(\rho_X(g)x,\rho_X(g)x') \qquad (85)\\ &= \sum_{\tilde{a}\in\mathcal{N}(a)}\sum_{\tilde{a}'\in\mathcal{N}(a')} \mathcal{K}_{\ell-1}^{\tilde{a},\tilde{a}'}(x,x') \qquad (86)\\ &= \mathcal{K}_\ell^{a,a'}(x,x') \qquad (87)\\ &= (\rho_K(g)\mathcal{K}_\ell(x,x')\rho_K^\top(g))^{a,a'}\,, \qquad (88)\end{aligned}$$

where we have used that $\rho = \mathbb{I}$ in this case. The induction step for the NTK of local aggregation layers proceeds along the same lines.

□

Corollary B.1. *By redefining $x' \to \rho_X^{-1}(g)x'$, the transformation properties of NNGP and NTK in Theorem 5.1 can equivalently be written as*

$$\begin{aligned}\mathcal{K}(\rho_X(g)x,x') &= \rho_K(g)\mathcal{K}(x,\rho_X^{-1}(g)x')\rho_K^\top(g) \qquad (89)\\ \Theta(\rho_X(g)x,x') &= \rho_K(g)\Theta(x,\rho_X^{-1}(g)x')\rho_K^\top(g)\,. \qquad (90)\end{aligned}$$

Lemma B.2. *Data augmentation implies that*

(a) $\Theta_{\pi_g(i),\,j} = \rho_K(g)\Theta_{i,\,\pi_g^{-1}(j)}\rho_K^\top(g)$,

(b) $\Theta^{-1}_{\pi_g(i),\,j} = \rho_K(g)\Theta^{-1}_{i,\,\pi_g^{-1}(j)}\rho_K^\top(g)$,

(c) $\mathcal{K}_{\pi_g(i),j} = \rho_K(g)\mathcal{K}_{i,\,\pi_g^{-1}(j)}\rho_K^\top(g)$,

(d) $\mathcal{K}^{-1}_{\pi_g(i),\,j} = \rho_K(g)\mathcal{K}^{-1}_{i,\,\pi_g^{-1}(j)}\rho_K^\top(g)$,

and analogous results hold for any power of Θ, Θ^{-1}, $\mathcal{K}$ and $\mathcal{K}^{-1}$, respectively.

Proof. **(a):** By data augmentation, it follows that

$$\begin{aligned}\Theta_{\pi_g(i),\,j} &= \Theta(x_{\pi_g(i)},x_j) \qquad (91)\\ &= \Theta(\rho_X(g)x_i,x_j) \qquad (92)\\ &= \rho_K(g)\Theta(x_i,\rho_X(g)^{-1}x_j)\rho_K^\top(g) \qquad (93)\\ &= \rho_K(g)\Theta(x_i,x_{\pi_g^{-1}(j)})\rho_K^\top(g) \qquad (94)\\ &= \rho_K(g)\Theta_{i,\,\pi_g^{-1}(j)}\rho_K^\top(g)\,, \qquad (95)\end{aligned}$$

where we used Corollary B.1 in (93). For any power $N \in \mathbb{N}$ of the kernel, it holds therefore that

$$\begin{aligned}
\left[\Theta^N\right]_{\pi_g(i),\, j} &= \Theta_{\pi_g(i),\, l} \left[\Theta^{N-1}\right]_{lj} && (96)\\
&= \rho_K(g)\Theta_{i,\, \pi_g^{-1}(l)}\rho_K^\top(g) \left[\Theta^{N-1}\right]_{lj} && (97)\\
&\overset{l\mapsto\pi_g(l)}{=} \rho_K(g)\Theta_{il}\rho_K^\top(g) \left[\Theta^{N-1}\right]_{\pi_g(l)j} && (98)\\
&= \rho_K(g)\Theta_{il}\rho_K^\top(g)\Theta_{\pi(l)m} \left[\Theta^{N-2}\right]_{mj} && (99)\\
&= \rho_K(g)\Theta_{il}\rho_K^\top(g)\rho_K(g)\Theta_{l\pi_g^{-1}(m)}\rho_K^\top(g) \left[\Theta^{N-2}\right]_{mj} && (100)\\
&= \rho_K(g)\Theta^2_{i\pi_g^{-1}(m)}\rho_K^\top(g) \left[\Theta^{N-2}\right]_{mj} && (101)\\
&= \cdots = \rho_K(g) \left[\Theta^N\right]_{i,\, \pi_g^{-1}(j)} \rho_K^\top(g)\,. && (102)
\end{aligned}$$

Here, the contraction of spatial axes between adjacent kernels is implicit and in (100), we have redefined these summation variables over spatial axes to absorb the action of $\rho_K^\top \rho_K$.

(b): We start from the equality

$$\Theta(\mathcal{X}, \rho_X(g)\mathcal{X})_{il} \left[\Theta(\mathcal{X}, \rho_X(g)\mathcal{X})\right]^{-1}_{lj} = \delta_{ij}\,, \tag{103}$$

where we have used the following notation for the Gram matrix $\Theta(\mathcal{X}, \mathcal{X})_{ij} := \Theta_{ij}$ and G acts sample-wise on the dataset, $(\rho_X(g)\mathcal{X})_i = \rho_X(g)x_i$. By data augmentation, this can be rewritten as

$$\Theta(\mathcal{X}, \mathcal{X})_{i,\, \pi_g(l)} \left[\Theta(\mathcal{X}, \rho_X(g)\mathcal{X})\right]^{-1}_{lj} = \delta_{ij}\,. \tag{104}$$

We now relabel the summation variable $l \to \pi_g^{-1}(l)$ and obtain

$$\Theta(\mathcal{X}, \mathcal{X})_{il} \left[\Theta(\mathcal{X}, \rho_X(g)\mathcal{X})\right]^{-1}_{\pi_g^{-1}(l),\, j} = \delta_{ij}\,. \tag{105}$$

By uniqueness of the inverse matrix, it thus follows that

$$\Theta(\mathcal{X}, \mathcal{X})^{-1}_{lj} = \left[\Theta(\mathcal{X}, \rho_X(g)\mathcal{X})\right]^{-1}_{\pi_g^{-1}(l),\, j} \qquad \iff \qquad \Theta(\mathcal{X}, \mathcal{X})^{-1}_{\pi_g(l),\, j} = \left[\Theta(\mathcal{X}, \rho_X(g)\mathcal{X})\right]^{-1}_{lj}\,. \tag{106}$$

Similarly, we can start from the expression

$$\left[\Theta(\rho_X(g)^{-1}\mathcal{X}, \mathcal{X})\right]^{-1}_{il} \Theta(\rho_X^{-1}(g)\mathcal{X}, \mathcal{X})_{lj} = \delta_{ij}\,. \tag{107}$$

By data augmentation, this can be rewritten as

$$\left[\Theta(\rho_X(g)^{-1}\mathcal{X}, \mathcal{X})\right]^{-1}_{il} \Theta(\mathcal{X}, \mathcal{X})_{\pi_g^{-1}(l),\, j} = \delta_{ij}\,. \tag{108}$$

Relabeling the summation variable $l \to \pi_g(l)$, we obtain

$$\left[\Theta(\rho_X^{-1}(g)\mathcal{X}, \mathcal{X})\right]^{-1}_{i,\pi_g(l)} \Theta(\mathcal{X}, \mathcal{X})_{lj} = \delta_{ij}\,. \tag{109}$$

By uniqueness of the inverse matrix, it follows again that

$$\Theta(\mathcal{X}, \mathcal{X})^{-1}_{il} = \left[\Theta(\rho_X^{-1}(g)\mathcal{X}, \mathcal{X})\right]^{-1}_{i,\pi_g(l)} \qquad \iff \qquad \Theta(\mathcal{X}, \mathcal{X})^{-1}_{i,\pi_g^{-1}(l)} = \left[\Theta(\rho_X^{-1}(g)\mathcal{X}, \mathcal{X})\right]^{-1}_{il}\,. \tag{110}$$

Combining the results (106) and (110), the statement of the lemma follows immediately:

$$\begin{aligned}
\Theta^{-1}_{\pi_g(i),j} = \left[\Theta(\mathcal{X}, \mathcal{X})\right]^{-1}_{\pi_g(i),j} &\overset{(106)}{=} \left[\Theta(\mathcal{X}, \rho_X(g)\mathcal{X})\right]^{-1}_{ij} && (111)\\
&= \rho_K(g) \left[\Theta(\rho_X^{-1}(g)\mathcal{X}, \mathcal{X})\right]^{-1}_{ij} \rho_K^\top(g) && (112)\\
&\overset{(110)}{=} \rho_K(g)\Theta(\mathcal{X}, \mathcal{X})^{-1}_{i,\pi_g^{-1}(j)}\rho_K^\top(g) && (113)\\
&= \rho_K(g)\Theta^{-1}_{i,\pi_g^{-1}(j)}\rho_K^\top(g)\,, && (114)
\end{aligned}$$

where ρ_K is unaffected by the inverse since we invert the Gram matrix along the training sample axes i, j. The proof for any power $(\Theta^{-1})^N$ of the inverse Gram matrix follows in complete analogy to the proof of the same result for the Gram matrix Θ.

(c): The proof for the NNGP follows in close analogy to the one for the NTK, see (a):

$$\begin{aligned}\mathcal{K}_{\pi_g(i),j} &= \mathcal{K}(x_{\pi_g(i)}, x_j) = \mathcal{K}(\rho_X(g)x_i, x_j) && (115)\\ &= \rho_K(g)\mathcal{K}(x_i, \rho_X^{-1}(g)x_j)\rho_K^\top(g) = \rho_K(g)\mathcal{K}(x_i, x_{\pi_g^{-1}(j)})\rho_K^\top(g) && (116)\\ &= \rho_K(g)\mathcal{K}_{i,\,\pi_g^{-1}(j)}\rho_K^\top(g)\,. && (117)\end{aligned}$$

The proof for any power of the NNGP again follows in complete analogy to (a).

(d): Since the transformation properties of Θ and $\mathcal{K}$ under G are completely identical, the proof follows the steps of (b) verbatim with the replacement $\Theta \to \mathcal{K}$. Similarly for any power of $\mathcal{K}$. □

Using this result, we can then show the following lemma as stated in the main part:

Lemma B.3 (Shift of permutation)**.** *Data augmentation implies that the permutation group action Π commutes with any matrix-valued analytical function F involving the Gram matrices of the NNGP and NTK as well as their inverses:*

$$\begin{aligned}&\Pi(g)F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})\\ &= \rho_K(g)F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})\Pi(g)\rho_K^\top(g)\,. && (22)\end{aligned}$$

where $\Pi(g)$ denotes the group action in terms of training set permutations as a permutation matrix, see (11).

Proof. As the matrix-valued function F is analytic, it has the following series expansion

$$F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})_{ij} = \sum_{n=1}^{\infty}\sum_{P_n} c_{P_n}\; P_n(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})_{ij}\,, \qquad (118)$$

where the inner sum is over all order n polynomials involving Θ and $\mathcal{K}$ as well as their inverses and c_{P_n} are coefficients.

By Lemma B.2, for any such polynomial P_n we have

$$P_n(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})_{\pi_g(i)j} = \rho_K(g)P_n(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})_{i\pi_g^{-1}(j)}\rho_K^\top(g)\,. \qquad (119)$$

Applying this result to the series expansion above implies

$$\begin{aligned}\left[\Pi(g)F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})\right]_{ij} &= F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})_{\pi_g(i)j} && (120)\\ &= \rho_K(g)F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})_{i\pi_g^{-1}(j)}\rho_K^\top(g) && (121)\\ &= \rho_K(g)\left[F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})(\Pi^{-1}(g))^\top\right]_{ij}\rho_K^\top(g) && (122)\\ &= \rho_K(g)\left[F(\Theta, \Theta^{-1}, \mathcal{K}, \mathcal{K}^{-1})\Pi(g)\right]_{ij}\rho_K^\top(g)\,. && (123)\end{aligned}$$

□

Theorem 5.3 (Emergent Equivariance of Deep Ensembles)**.** *Under the assumptions stated above, the distribution of large-width ensemble members $f_w : X \to Y$ is equivariant with respect to the representations ρ_X and ρ_Y of the group G if data augmentation is applied. In particular, the ensemble is equivariant,*

$$\bar{f}_t(\rho_X(g)\,x) = \rho_Y(g)\,\bar{f}_t(x)\,, \qquad (26)$$

for all $g \in G$. This result holds

1. *at any training time t,*
2. *for any element of the input space $x \in X$.*

Proof. The mean function of the output distribution on a test sample x after training time t is according to (6) given by

$$\begin{aligned}\mu(\rho_X(g)x) &= \Theta(\rho_X(g)x, \mathcal{X})[\Theta^{-1}T_t]\mathcal{Y}) && (124)\\ &= \rho_K(g)\Theta(x, \rho_X^{-1}(g)\mathcal{X})\rho_K^\top(g)[\Theta^{-1}T_t]\mathcal{Y} && (125)\\ &= \rho_K(g)\Theta(x, \mathcal{X})\rho_K^\top(g)\Pi(g)[\Theta^{-1}T_t]\mathcal{Y} && (126)\\ &= \rho_K(g)\Theta(x, \mathcal{X})[\Theta^{-1}T_t]\rho_K^\top(g)\Pi(g)\mathcal{Y} && (127)\\ &= \rho_K(g)\Theta(x, \mathcal{X})[\Theta^{-1}T_t]\rho_K^\top(g)\rho_Y(g)\mathcal{Y}\,, && (128)\end{aligned}$$

On a label with spatial index a and channel index α, ρ_Y acts according to (13). Furthermore, the index structure of the NTK will match the index structure of the labels since we use the network outputs to predict the labels. If the labels carry a channel index, then Θ is proportional to the unit matrix in this index, as mentioned in (64) and hence the representation τ_Y commutes all the way to the left. Finally, the action of ρ on the spatial indices of the labels (if present) is the same as the action of ρ_K, so we obtain

$$\begin{aligned}
\mu(\rho_X(g)x) &= \tau_Y(g)\rho_K(g)\Theta(x,\mathcal{X})[\Theta^{-1}T_t]\rho_K^\top(g)\rho_K(g)\mathcal{Y} && (129)\\
&= \tau_Y(g)\rho_K(g)\Theta(x,\mathcal{X})[\Theta^{-1}T_t]\mathcal{Y} && (130)\\
&= \tau_Y(g)\rho_K(g)\mu(x) && (131)\\
&= \rho_Y(g)\mu(x)\,. && (132)
\end{aligned}$$

The covariance function transforms according to

$$\Sigma_t(\rho_X(g)x,\rho_X(g)x') = \mathcal{K}(\rho_X(g)x,\rho_X(g)x') + \Sigma_t^{(1)}(\rho_X(g)x,\rho_X(g)x') - (\Sigma_t^{(2)}(\rho_X(g)x,\rho_X(g)x') + \text{h.c.})\,. \qquad (133)$$

The transformation of $\mathcal{K}$ is given by Theorem 5.1. The transformation of $\Sigma_t^{(1)}$ is given by

$$\begin{aligned}
&\Sigma_t^{(1)}(\rho_X(g)x,\rho_X(g)x')\\
&= \Theta(\rho_X(g)x,\mathcal{X})\,(\Theta(\mathcal{X},\mathcal{X}))^{-1}\,T_t\,\mathcal{K}(\mathcal{X},\mathcal{X})\,T_t\,(\Theta(\mathcal{X},\mathcal{X}))^{-1}\,\Theta(\mathcal{X},\rho_X(g)x') && (134)\\
&= \rho_K(g)\Theta(x,\mathcal{X})\rho_K^\top(g)\Pi(g)\,(\Theta(\mathcal{X},\mathcal{X}))^{-1}\,T_t\,\mathcal{K}(\mathcal{X},\mathcal{X})\,T_t\,(\Theta(\mathcal{X},\mathcal{X}))^{-1}\,\Pi^\top(g)\rho_K(g)\Theta(\mathcal{X},x')\rho_K^\top(g) && (135)\\
&= \rho_K(g)\Theta(x,\mathcal{X})\,(\Theta(\mathcal{X},\mathcal{X}))^{-1}\,T_t\,\mathcal{K}(\mathcal{X},\mathcal{X})\,T_t\,(\Theta(\mathcal{X},\mathcal{X}))^{-1}\,\rho_K^\top(g)\Pi(g)\Pi^\top(g)\rho_K(g)\Theta(\mathcal{X},x')\rho_K^\top(g) && (136)\\
&= \rho_K(g)\Theta(x,\mathcal{X})\,(\Theta(\mathcal{X},\mathcal{X}))^{-1}\,T_t\,\mathcal{K}(\mathcal{X},\mathcal{X})\,T_t\,(\Theta(\mathcal{X},\mathcal{X}))^{-1}\,\Theta(\mathcal{X},x')\rho_K^\top(g) && (137)\\
&= \rho_K(g)\Sigma_t^{(1)}(x,x')\rho_K^\top(g)\,. && (138)
\end{aligned}$$

Similarly for $\Sigma_t^{(2)}$. In total, the covariance function transforms according to

$$\Sigma_t(\rho_X(g)x,\rho_X(g)x') = \rho_K(g)\Sigma_t(x,x')\rho_K^\top(g)\,. \qquad (139)$$

Since the covariance function is also proportional to the unit matrix in possible channel dimensions, adding a transformation of τ_Y on the left and $\tau_Y^\top$ on the right does not change the expression. Therefore

$$\Sigma_t(\rho_X(g)x,\rho_X(g)x') = \rho_Y(g)\Sigma_t(x,x')\rho_Y^\top(g)\,. \qquad (140)$$

Together with (132), this implies that the output distribution is equivariant w.r.t. ρ_Y for any training time t and for any input x. □

B.1. Finite Number of Ensemble Members

Lemma B.4. *The probability that the deep ensemble $\bar{f}_t$ and its estimate $\hat{f}_t$ differ by more than a given threshold δ is bounded by*

$$\mathbb{P}\left[|\hat{f}_t(x) - \bar{f}_t(x)| > \delta\right] \leq \sqrt{\frac{2}{\pi}}\,\frac{\sigma_x}{\delta}\exp\left(-\frac{\delta^2}{2\sigma_x^2}\right)\,, \qquad (141)$$

where we have defined

$$\sigma_x^2 := \mathit{Var}(\hat{f}_t)(x) = \frac{\Sigma_t(x)}{M} \qquad (142)$$

with the output variance $\Sigma_t(x) = \Sigma_t(x,x)$ defined in (7).

Proof. The probability of such deviations is given by

$$\mathbb{P}\left[|\hat{f}_t(x) - \bar{f}_t(x)| > \delta\right] = \frac{2}{\sqrt{2\pi}\sigma_x}\int_\delta^\infty \exp\left(-\frac{t^2}{2\sigma_x^2}\right)\mathrm{d}t \qquad (143)$$

We now change the integration variable to $\tau = \frac{t}{\sigma_x\sqrt{2}}$ and obtain

$$\mathbb{P}\left[|\hat{f}_t(x) - \bar{f}({}_tx)| > \delta\right] = \frac{2}{\sqrt{\pi}}\int_{\frac{\delta}{\sqrt{2}\sigma_x}}^\infty \exp(-\tau^2)\mathrm{d}\tau \leq \frac{1}{\sqrt{\pi}}\,\frac{\sqrt{2}\sigma_x}{\delta}\int_{\frac{\delta}{\sqrt{2}\sigma_x}}^\infty (2\tau)\,\exp(-\tau^2)\mathrm{d}\tau\,, \qquad (144)$$

where we have used that $1 \leq \frac{2\tau}{2\min(\tau)}$ for $\tau \geq \min(\tau)$ to obtain the last inequality. The integral can be straightforwardly evaluated by rewriting the integrand as a total derivative and we thus obtain

$$\mathbb{P}\left[|\hat{f}_t(x) - \bar{f}_t(x)| > \delta\right] \leq \sqrt{\frac{2}{\pi}}\,\frac{\sigma_x}{\delta}\,\exp\left(-\frac{\delta^2}{2\sigma_x^2}\right)\,. \tag{145}$$

□

We stress that this result holds for any Monte-Carlo estimator and we therefore suspect that it could be well-known. For most MC estimators, it is however of relatively little use as the variance Σ is not known in closed form — in stark contrast to the deep ensemble, see (7), considered in this paper. This could explain why we were not able to locate this result in the literature.

For the deep ensemble, we can therefore exactly determine the necessary number of ensemble size to stay within a certain threshold δ with a given probability $1-\epsilon$. For this, one has to set the right-hand-side of the derived expression to this confidence ϵ and solve for the necessary ensemble size M. However, this equation appears to have no closed-from solution and needs to be solved numerically. We advise the reader to do so if need for a tight bound arises. For the presentation in the main part, we however wanted to derive a closed-form solution for M and thus had to rely on a looser bound which implies the following statement:

Lemma B.5 (Bound for finite ensemble members)**.** *The deep ensemble $\bar{f}_t$ and its estimate $\hat{f}_t$ do not differ by more than threshold δ,*

$$|\bar{f}_t(x) - \hat{f}_t(x)| < \delta\,, \tag{27}$$

with probability $1-\epsilon$ for ensemble sizes M that obey

$$M > -\frac{2\Sigma_t(x)}{\delta^2}\ln\left(\sqrt{\pi}\epsilon\right)\,. \tag{28}$$

Proof.

$$\mathbb{P}\left[|\hat{f}_t(x) - \bar{f}_t(x)| > \delta\right] < \frac{1}{\sqrt{\pi}}\,\frac{1}{z}\,\exp\left(-z^2\right) \leq \frac{1}{\sqrt{\pi}}\,\exp\left(-z^2\right) \overset{!}{<} \epsilon \tag{146}$$

with $z = \frac{\delta}{\sqrt{2}\sigma_x}$ and where we assume that M is chosen sufficiently large such that $z \geq 1$. This implies that

$$z^2 > -\ln\!\left(\sqrt{\pi}\epsilon\right) \qquad \Leftrightarrow \qquad M > -\frac{2\Sigma_t(x)}{\delta^2}\ln\!\left(\sqrt{\pi}\epsilon\right). \tag{147}$$

□

B.2. Continuous Groups

Lemma B.6 (Bound for continuous groups)**.** *Consider a deep ensemble of neural networks with Lipschitz continuous derivatives with respect to the parameters. For an approximation $A \subset G$ of a continuous symmetry group G with discretization error ϵ, the prediction of the ensemble trained on A deviates from invariance by*

$$|\bar{f}_t(x) - \bar{f}_t(\rho_X(g)\,x)| \leq \epsilon\, C(x)\,, \qquad \forall g \in G\,,$$

where C is independent of g.

Proof. As described in the main text, we consider a finite subgroup $A \subset G$ which we use for data augmentation (instead of using the continuous group G). The discretization error for the representation ρ_X is given by

$$\epsilon = \max_{g\in G}\,\min_{g'\in A}\,\|\rho_X(g) - \rho_X(g')\|_{\mathrm{op}}\,. \tag{148}$$

This implies that for any $g \in G$, we can find a $g' \in A$ such that

$$\|\rho_X(g)x_i - x_{\pi_{g'}(i)}\| = \|\rho_X(g)x_i - \rho_X(g')x_i\| \leq \|\rho_X(g) - \rho_X(g')\|_{\mathrm{op}}\,\|x_i\| < \epsilon\|x_i\|\,, \tag{149}$$

where we have used data augmentation (11) over A.

We can then calculate the difference of the prediction at any test point x and its transformation:

$$|\bar{f}_t(x) - \bar{f}_t(\rho_X(g)x)| = |\mu_t(x) - \mu_t(\rho_X(g)x)| \tag{150}$$

$$= |(\Theta(x,x_i) - \Theta(\rho_X(g)x, x_i))\,\Theta^{-1}_{ij}\,(\mathbb{I} - \exp(-\eta\Theta t))_{jk}\,y_k| \tag{151}$$

From the Lemma 5.2, it follows that

$$\Theta(x,x_i)\,\Theta^{-1}_{ij}\,(\mathbb{I}-\exp(-\eta\Theta t))_{jk}\,y_k = \Theta(x,x_i)\,\Theta^{-1}_{ij}\,(\mathbb{I}-\exp(-\eta\Theta t))_{jk}\,y_{\pi_{g'}(k)} \tag{152}$$

$$= \Theta(x,x_{\pi^{-1}_{g'}(i)})\,\Theta^{-1}_{ij}\,(\mathbb{I}-\exp(-\eta\Theta t))_{jk}\,y_k \tag{153}$$

Thus the difference can be rewritten as follows

$$|\bar{f}_t(x)-\bar{f}_t(\rho_X(g)x)| = |(\Theta(x,x_{\pi^{-1}_{g'}(i)})-\Theta(\rho_X(g)x,x_i))\,\Theta^{-1}_{ij}\,(\mathbb{I}-\exp(-\eta\Theta t))_{jk}\,y_k| \tag{154}$$

$$= |(\Theta(x,x_{\pi^{-1}_{g'}(i)})-\Theta(x,\rho_X^{-1}(g)x_i))\,\Theta^{-1}_{ij}\,(\mathbb{I}-\exp(-\eta\Theta t))_{jk}\,y_k| \tag{155}$$

It is convenient to define

$$\Delta\Theta(x',x,\bar{x}) \equiv |\Theta(x',x)-\Theta(x',\bar{x})| \tag{156}$$

which can be bounded as follows

$$\Delta\Theta(x',x,\bar{x}) = \left|\sum_{l=1}^{L}\mathbb{E}_{w\sim p}\left[\left(\frac{\partial f_w(x')}{\partial w^{(l)}}\right)^{\top}\left(\frac{\partial f_w(x)}{\partial w^{(l)}}-\frac{\partial f_w(\bar{x})}{\partial w^{(l)}}\right)\right]\right| \tag{157}$$

$$\leq \|x-\bar{x}\|\sum_{l=1}^{L}\mathbb{E}_{w\sim p}\left[\left|\left(\frac{\partial f_w(x')}{\partial w^{(l)}}\right)^{\top}\cdot L(w^{(l)})\right|\right] \tag{158}$$

$$\equiv \|x-\bar{x}\|\,\hat{C}(x)\,, \tag{159}$$

where $L(w^{(l)})$ is the Lipschitz constant of $\partial_{w^{(l)}}f_w$ and we emphasize that the norm is with respect to the input space. Using this expression, we can bound the difference of the means (155) by using the triangle inequality

$$|\bar{f}_t(x)-\bar{f}_t(\rho_X(g)x)| \leq \hat{C}(x)\sqrt{\sum_i \|x_{\pi^{-1}_{g'}(i)}-\rho_X(g)^{-1}x_i\|^2}\,\sqrt{\sum_i(\sum_{j,k}\Theta^{-1}_{ij}\,(\mathbb{I}-\exp(-\eta\Theta t))_{jk}\,y_k])^2}$$

$$\leq \epsilon\,\hat{C}(x)\sqrt{\sum_i \|x_i\|^2}\,\sqrt{\sum_i(\sum_{j,k}\Theta^{-1}_{ij}\,(\mathbb{I}-\exp(-\eta\Theta t))_{jk}\,y_k])^2} \equiv \epsilon C(x)\,.$$

Note that this result suggests that one should choose the discretization carefully to achieve as tight of a bound as possible. □

C. Experiments

In this section, we provide further details about our experiments.

C.1. Ising Model

Training details The energy function of the Ising model can be written as

$$\mathcal{E} = -\frac{J}{\mathrm{vol}(L)}\sum_{i\in L}E(i)\,, \tag{160}$$

where J is a coupling constant which we set to one for convenience and vol(L) denotes the number of lattice sites. The local energy $E(i)$ is given by[1]

$$E(i) = \sum_{j\in\mathcal{N}(i)} s_i s_j\,, \tag{161}$$

where $\mathcal{N}(i)$ denotes the neighbors of i along the lattice axes. The expectation value of $\mathcal{E}$ vanishes and its standard deviation is 2 for uniform sampling of spins in $\{+1,-1\}$.

The energy of the Ising model is invariant under rotations of the lattice by 90°, since the local energy (161) stays invariant if the neighborhood is rotated and the sum in (160) is just reshuffled. We train a fully-connected network with one hidden layer and a ReLU activation on 128 samples augmented with full C_4 orbits to 512 training samples. To obtain a sufficient training signal, we train the networks with a squared error loss on the local energies (161). We train for 100k steps of full-batch gradient descent with learning rate 0.5 for network widths 128, 512 and 1024 and learning rate 1.0 for network width 2048.

[1] Usually, one only sums over pairs of spins. Our prescription differs from that convention by an irrelevant factor of two and makes the local energy exactly equivariant under rotations of the lattice by 90°.

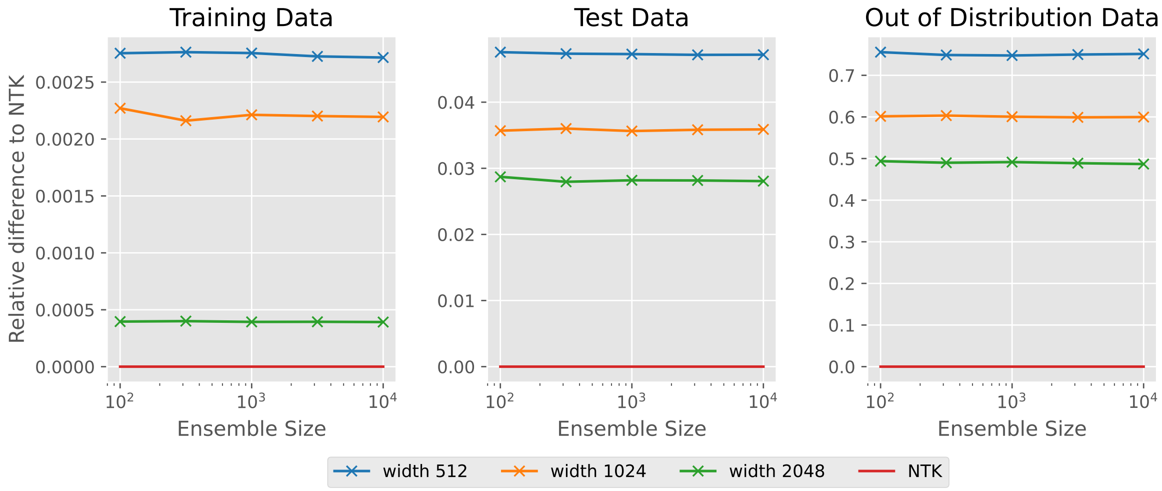

Figure 5. Difference in relative predicted total energy $\mathcal{E}$ between the ensembles and the NTK on the training data, in-distribution test data and out of distribution.

Ensemble-convergence to the NTK We verify that the ensembles converge to the NTK for large widths by computing the difference in total energy $\mathcal{E}$ between the mean ensemble prediction and the predicted mean of the NTK, cf. Figure 5. To make the numbers easily interpretable, we plot the relative difference, where we divide by the standard deviation of the ground truth energy, 2, which gives a typical value for $\mathcal{E}$. We perform the comparisons on the training data, in-distribution test data and out of distribution data. As expected, agreement is highest on the training data and lowest out of distribution, but in each case, ensembles with higher-width hidden layer generate mean predictions closer to the NTK. Beyond ensemble size 1000, the estimate of the expectation value over initializations in the NTK seems to be accurate enough that no further fluctuations can be seen in the plots.

C.2. Rotated FashionMNIST

Ensemble architecture As ensemble members, we use a simple convolutional neural network with two convolutional layers of kernel size 3 and 6 as well as 16 channels respectively. Both convolutional layers are followed by a relu non-linearity as well as 2×2 max-pooling. This is then followed by layers fully-connected of size $(400, 120)$, $(120, 84)$, and $(84, 10)$ of which the first two are fed into relu non-linearities. We choose ensembles of size $M = 5, 10, 100$.

OOD data We use the validation set of greyscaled and rescaled CIFAR10, the validation set of MNIST, as well as a dataset generated by images with pixels drawn iid from $N(0, 1)$ as OOD data. We also evaluate the invariance on the validation set of FMNIST, i.e., on in-distribution data. Please refer to the corresponding Figure 9, 10, and 11 contained in this appendix for the results.

Data augmentation We augment the original dataset by all elements of the group orbit of the cyclic group C_k, i.e., all rotations of the image by any multiple of $360/k$ degrees and ensure that each epoch contains all element of the group orbit in each epoch to closely align the experiments with our theoretical analysis. However, in exploratory analysis, we did not observe a notable difference when applying random group elements in each training step. For the cyclic group C_k, we choose group orders $k = 4, 8, 16$.

Training details We use the ADAM optimizer with the standard learning rate of pytorch lightning, i.e., 1e-3. We train for 10 epochs on the augmented dataset. We evaluate the metrics after each epoch on both the in-distribution and the out-of-distribution data. The ensembles achieve a test accuracy on the augmented datasets of between 88 to 91 percent depending on the chosen group order and ensemble size.

OSP metric: To obtain the orbit same prediction, we measure

$$\sum_{g \in G} \mathbb{I}(\text{argmax}_\alpha f^\alpha(\rho_X(g)x), \text{argmax}_\alpha f^\alpha(x)) , \tag{162}$$

where $\mathbb{I}$ denotes the indicator function. This corresponds to the number of elements in the orbit that have the same predicted class as the transformed data sample x. The orbit same prediction (OSP) of a dataset $\mathcal{D}$ is then this number averaged over all elements in the dataset. Note that the OSP has minimal value 1 as the identity is always part of the orbit.

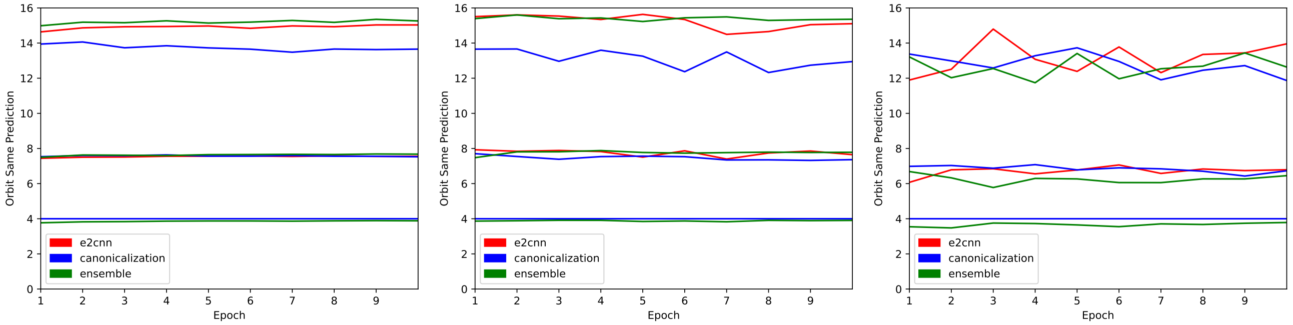

Figure 6. Comparison of various equivariance methods on in-distribution FMNIST (left), out-of-distribution MNIST (middle), out-of-distribution CIFAR10 (right). Deep ensembles are approximately equivariant due to finite-size ensembles and finite width (see discussion in main text). Canonicalization and E2CNN also do not show perfect equivariance for group orders $k > 4$ because of interpolation artifacts, see, for example, discussion in (Kaba et al., 2023).

Continuous rotations: We analyze the generalization properties to the full two-dimensional rotation group $SO(2)$ for deep ensembles trained with data augmentation using the finite cyclic group C_k. To this end, we define the continuous orbit same prediction as:

$$\frac{1}{\mathrm{Vol}(SO(2))} \int_{SO(2)} \mathrm{d}g\, \mathbb{I}(\mathrm{argmax}_\alpha f^\alpha(\rho_X(g)x), \mathrm{argmax}_\alpha f^\alpha(x)) \,, \tag{163}$$

where $\mathrm{d}g$ denotes the Haar measure. This continuous orbit same prediction thus corresponds to the percentage of elements in the orbit that are classified the same way as the untransformed element. We estimate this quantity by Monte-Carlo. The results of our analysis are shown in Figure 7 and clearly establish that for sufficiently high group order of the cyclic group used for data augmentation, the ensemble is approximately invariant with respect to the continuous symmetry as well. In particular, it is signficantly more invariant as its ensemble members. Interestingly, this is competitive with a model that is using canonicalization (Kaba et al., 2023) with respect to C_k and the same network architecture as its predictor network.

Comparison to other methods: For the deep ensemble, we use ten ensemble members with the same convolututional architecture as outlined above. For canonicalization, we use the same convolutional architecture as for the ensemble members and the same architecture for the canonicalization network as in the original publication (Kaba et al., 2023). For E2CNN, we follow the official MNIST example. We adjust hyperparameters such that all methods use roughly the same number of parameters as the deep ensemble. As a result, all methods have roughly the same number of parameters. Figure 6 demonstrates that all methods lead to a comparable degree of equivariance. Note that interpolation effects seem to hurt the performance of canonicalization more dramatically as compared to E2CNN. This is to be expected as canonicalization works by predicting a rotation and then undoing the rotation. This leads to another compounded source of discretization errors.

D. Cross Product

Training We train ensembles of two hidden-layer fully-connected networks to predict the cross-product $x \times y$ in $\mathbb{R}^3$ given two vectors x and y. This task is equivariant with respect to rotations $R \in \mathrm{SO}(3)$,

$$Rx \times Ry = R(x \times y) \,. \tag{164}$$

The training data consists of 100 vector pairs with components sampled from $\mathcal{N}(0, 1)$, the validation data consists of 1000 such pairs. For out of distribution data, we sample from a Poisson distribution with mean 0.5. We train using 10-fold data augmentation, i.e. we sample 10 rotation matrices from SO(3) and rotate the training data with these matrices, resulting in 1000 training vector pairs. We train for 50 epochs using the Adam optimizer and reach validation RMSEs of about 0.3 with exact performance depending on layer width and ensemble size.

Orbit MSE To evaluate how equivariant the ensembles trained with data augmentation are on a given dataset, we sample 100 rotation matrices from SO(3) and augment each input vector pair with their 100 rotated versions. Then, we predict the cross products on this enlarged dataset and rotate the predicted vectors back using the inverse rotations. Finally, we measure the MSE across the 100 back-rotated predictions against the unrotated prediction. The orbit MSE is averaged over the last five epochs.

The results of our experiments on the cross-product are shown in Figure 13. As above, we evaluate the orbit MSE on each ensemble member individually (solid lines and shaded region corresponding to ± one standard deviation) and for the ensemble output (dashed lines). This is true on training-, test- and out of distribution data. Also in this equivariant task is the ensemble mean about an order of magnitude more equivariant than the ensemble members. As expected from our theory, the ensemble becomes more equivariant for larger ensembles and wider networks.

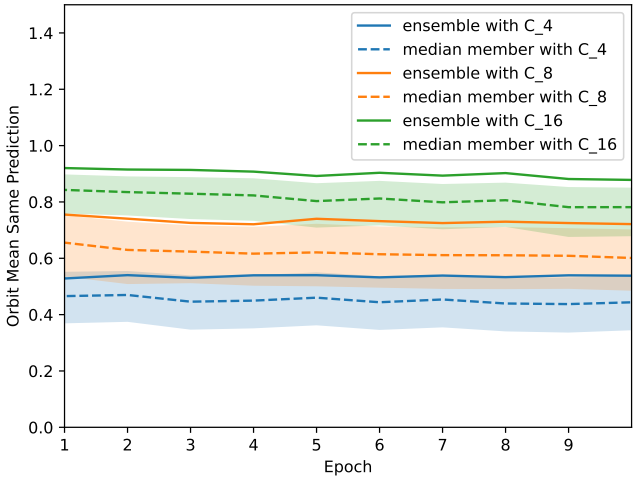

Figure 7. Mean orbit same prediction over $SO(2)$ group orbits. Solid lines show the ensemble prediction while dotted lines show the median of the ensemble members. Error band denotes the 75th and 25th percentile. As the group order k of the cyclic group C_k used for data augmentation increases, the mean orbit same prediction over $SO(2)$ increases. For $k = 16$, over 90 percent of the orbit elements have the same prediction as the untransformed input establishing that the model is approximately invariant under the continuous symmetry as well. The invariance of the ensemble is again emergent in the sense that it is above the 75th percentile of the ensemble members.

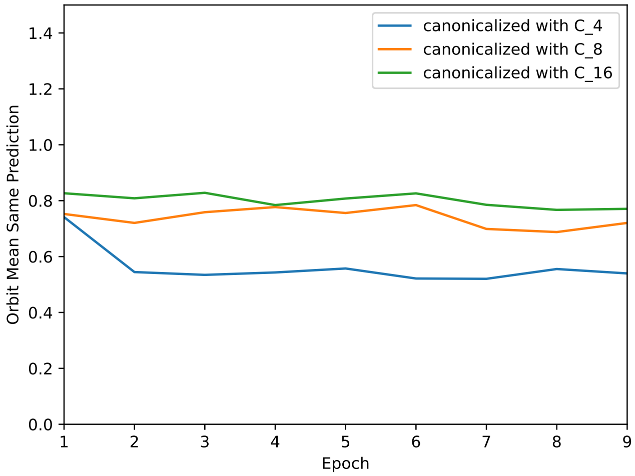

Figure 8. Mean orbit same prediction over $SO(2)$ group orbits for a model canonicalized with respect to C_k. As the group order k of the cyclic group C_k used for data augmentation increases, the mean orbit same prediction over $SO(2)$ increases.

E. Histological Slices

Training The NCT-CRC-HE-100K dataset (Kather et al., 2018) comprises 100k stained histological images in nine classes. In order to make the task more challenging, we only use 10k randomly selected samples, train on 11/12$^{\text{th}}$ of this subset and validate on the remaining 1/12$^{\text{th}}$. We trained ensembles of CNNs with six convolutional layers of kernel size 3 and 6, 16, 26, 36, 46 and 56 output channels, followed by a kernel size 2, stride 2 max pooling operation and three fully connected layers of 120, 84 and 9 output channels. The models had 123k parameters each. We trained the ensembles with the Adam optimizer using a learning rate of 0.001 on batches of size 16. In our training setup, ensemble members reach a validation accuracy of about 96% after 20 epochs, cf. Figure 15.

Invariance on in-distribution data As for our experiments on FashionMNIST, we verify that the ensemble is more invariant as a function of its input than the ensemble members. On training- and validation data this is to be expected since the ensemble predictions have a higher accuracy than the predictions of individual ensemble members. The invariance results on validation data are depicted in Figure 14.

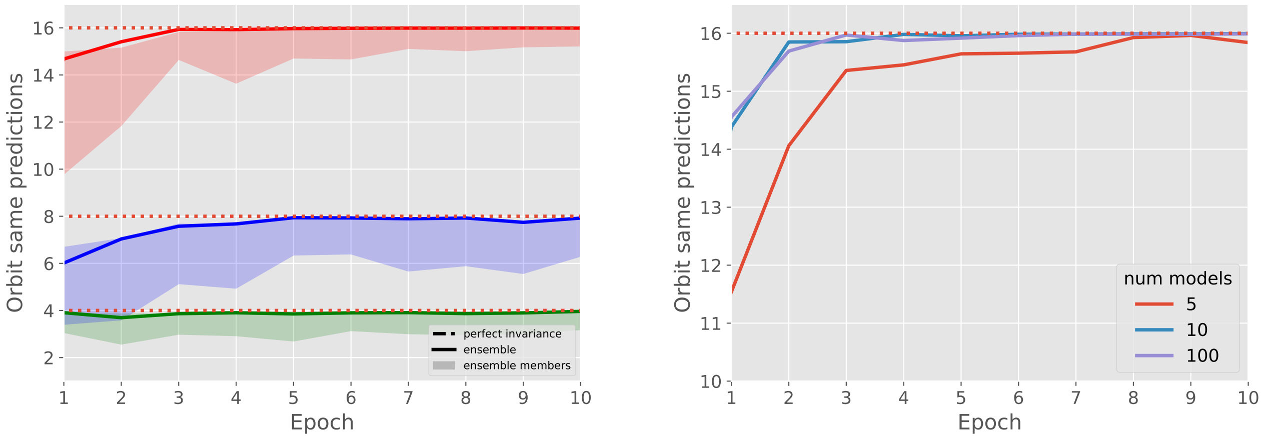

Figure 9. Same as Figure 2 but for OOD images with pixels drawn iid from $N(0, 1)$.

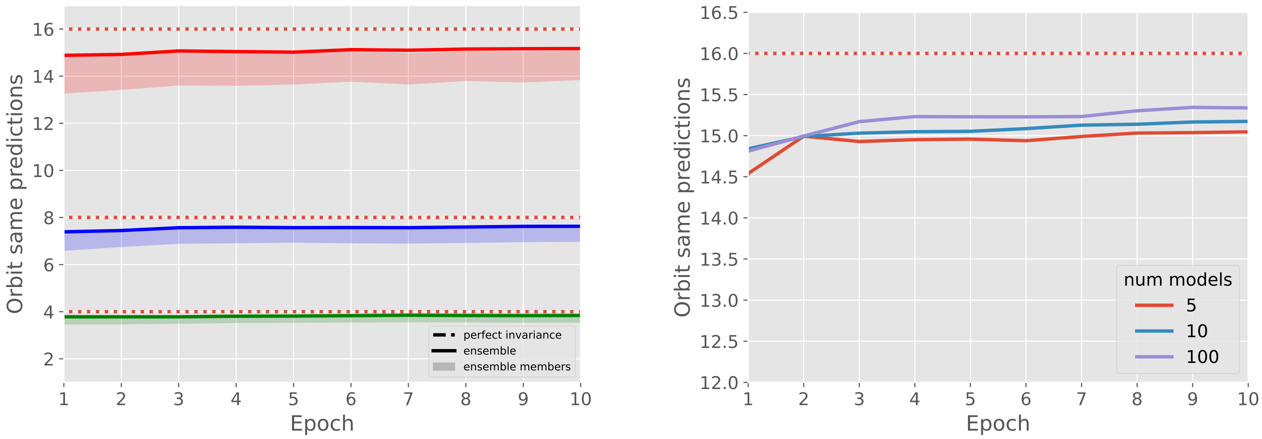

Figure 10. Same as Figure 2 but for FMNIST, i.e., in-distribution data.

OOD data In order to arrive at a sample of OOD data on which the network makes non-constant predictions, we optimize the input of the untrained ensemble using the Adam optimizer to yield predictions of high confidence ($> 99\%$), starting from 100 random normalized images for each class. We optimize only the 5×5 lowest frequencies in the Fourier domain to obtain samples which can be rotated without large interpolation losses, yielding samples as depicted in Figure 16.

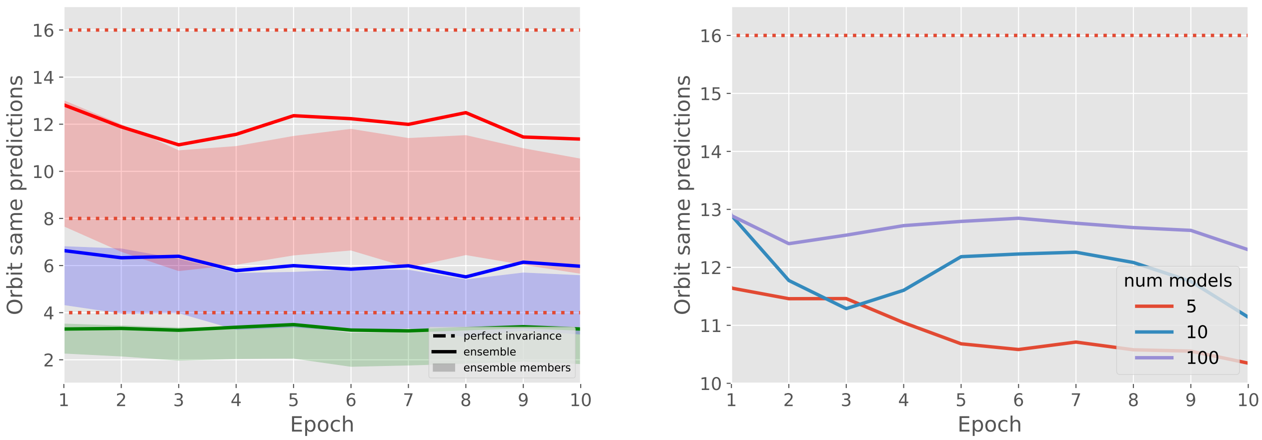

Figure 11. Same as Figure 2 but for rescaled and greyscaled CIFAR10 OOD data.

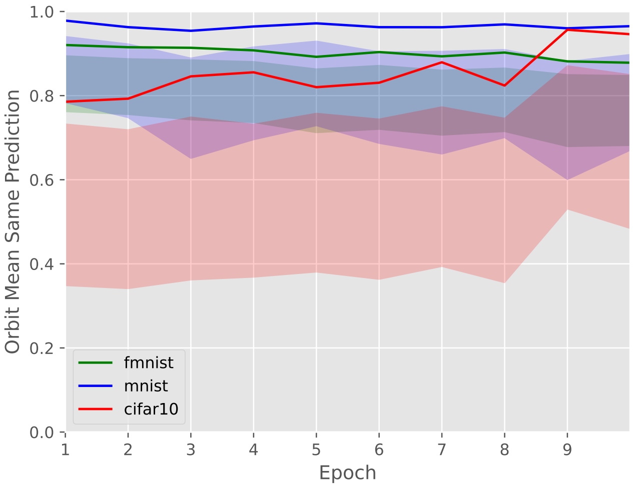

Figure 12. Equivariance extends to $SO(2)$ symmetry. Percentage of randomly sampled rotations that leave the prediction invariant is reported. Data augmentation with group order 16 is used.

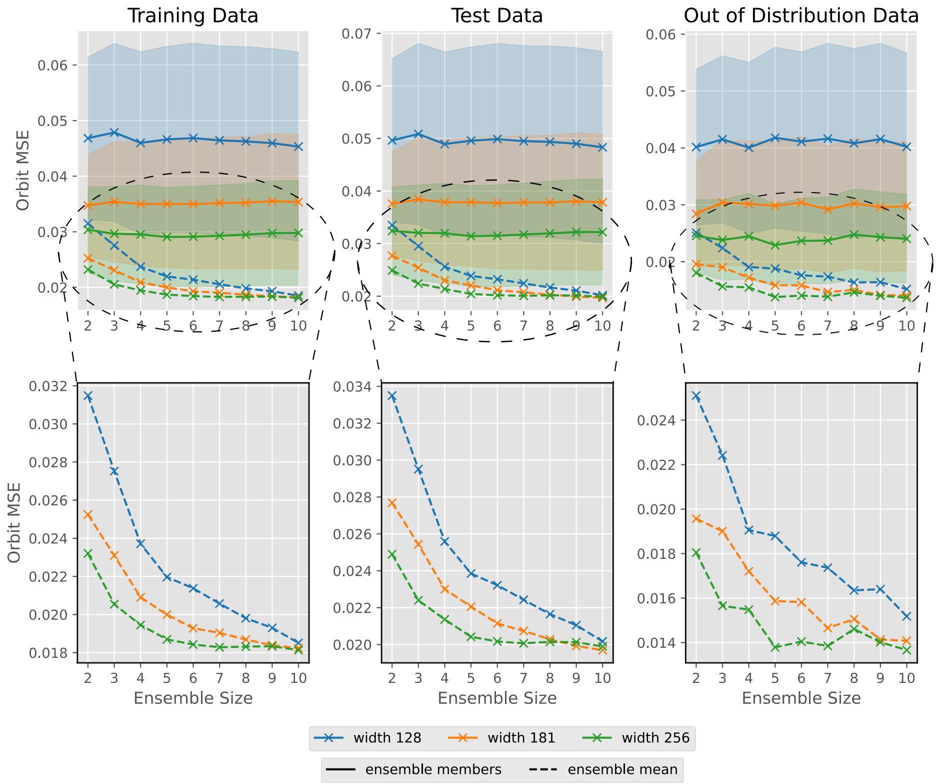

Figure 13. Emerging equivariance of ensembles predicting the cross-product. Plotted is the MSE of predictions across a random 100-element subset of the symmetry orbit of SO(3) versus ensemble size. Solid lines refer to the orbit MSE for individual ensemble members with shaded regions corresponding to ± one standard deviation, dashed lines refer to the ensemble prediction. Shown are evaluations on the training- (left), test- (middle) and out of distribution data (right). The lower row shows zoom-ins on the ensemble predictions.

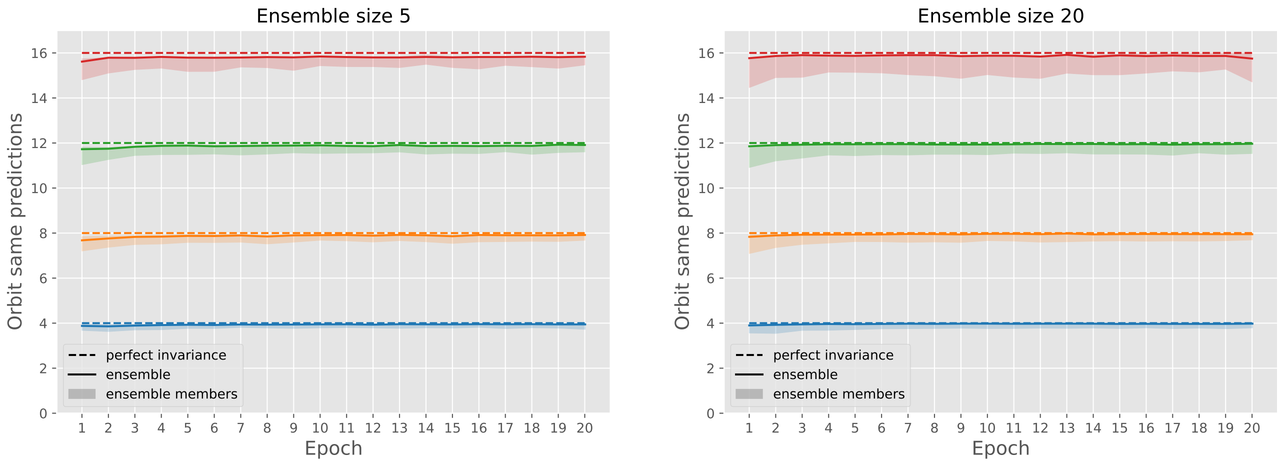

Figure 14. Ensemble invariance on validation data for ensembles trained on histological data. Number of validation samples with the same prediction across a symmetry orbit for group orders 4 (blue), 8 (orange), 12 (green) and 16 (red) versus training epoch for ensemble sizes 5 (left) and 20 (right). The ensemble predictions (solid line) are more invariant than the ensemble members (shaded region corresponding to 25^{th} to 75^{th} percentile of ensemble members). The effect is larger for ensemble size 20 (right).

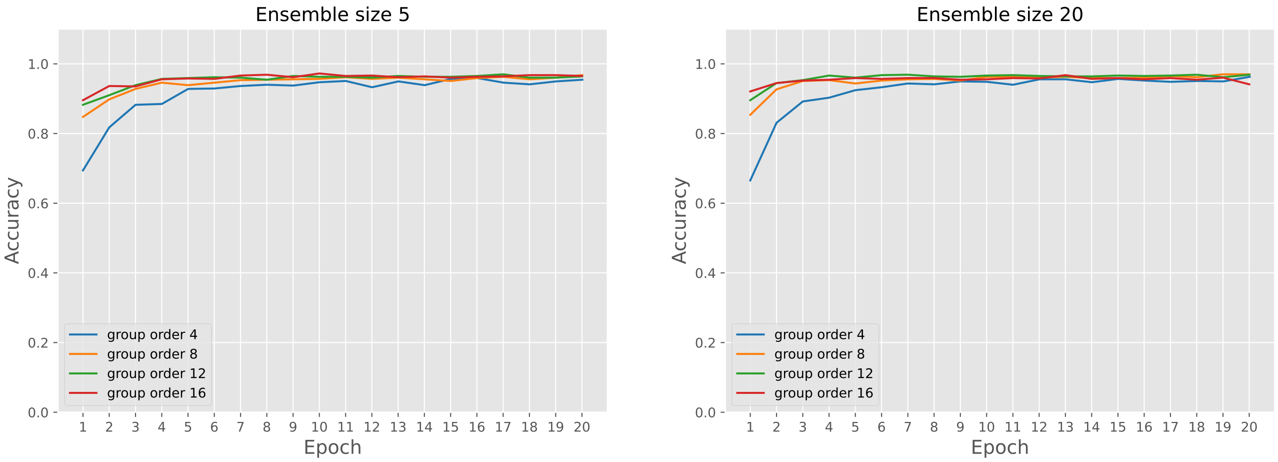

Figure 15. Validation accuracy versus training time for ensemble of size 5 (left) and 20 (right) trained on histological data.

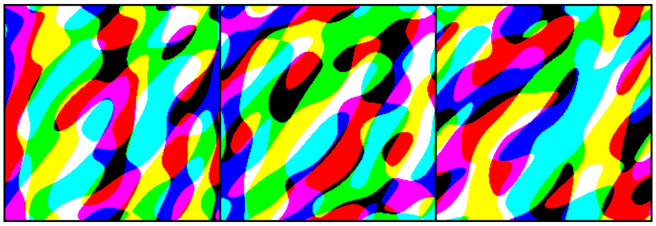

Figure 16. Three OOD data samples for the histology ensemble.