Reverse-Complement Equivariant Networks for DNA Sequences

Vincent Mallet
Structural Bioinformatics Unit, Department of Structural Biology and Chemistry,
Institut Pasteur, CNRS UMR3528, C3BI, USR3756
Mines ParisTech, PSL University, Center for Computational Biology
vincent.mallet96@gmail.com

Jean-Philippe Vert
Google Research, Brain team, Paris
jpvert@google.com

Abstract

As DNA sequencing technologies keep improving in scale and cost, there is a growing need to develop machine learning models to analyze DNA sequences, e.g., to decipher regulatory signals from DNA fragments bound by a particular protein of interest. As a double helix made of two complementary strands, a DNA fragment can be sequenced as two equivalent, so-called *Reverse Complement* (RC) sequences of nucleotides. To take into account this inherent symmetry of the data in machine learning models can facilitate learning. In this sense, several authors have recently proposed particular RC-equivariant convolutional neural networks (CNNs). However, it remains unknown whether other RC-equivariant architectures exist, which could potentially increase the set of basic models adapted to DNA sequences for practitioners. Here, we close this gap by characterizing the set of all linear RC-equivariant layers, and show in particular that new architectures exist beyond the ones already explored. We further discuss RC-equivariant pointwise nonlinearities adapted to different architectures, as well as RC-equivariant embeddings of k-mers as an alternative to one-hot encoding of nucleotides. We show experimentally that the new architectures can outperform existing ones.

1 Introduction

Incorporating prior knowledge about the structure of data in the architecture of neural networks is a promising approach to design expressive models with good generalization properties. In particular, exploiting natural symmetries in the data can lead to models with fewer parameters to estimate than agnostic approaches. This is especially beneficial when the amount of available data is limited. A famous example of such an architecture is the convolutional neural network (CNN) for 1D sequences or 2D images, which is well adapted to problems which are invariant to translations in the data, while exploiting multiscale and local information in the signals. Motivated by the success of CNNs, there has been a fast-growing body of research in recent years to build the theoretical underpinnings and design architectures and efficient algorithms to systematically exploit symmetries and structures in the data [3].

A central idea that has emerged is to formalize the symmetries in data by a particular *group action* (e.g., the group of translations or rotations on images), and to create multilayer neural networks which, by design, "behave well" under the action of the group. This is captured formally by the concept of *equivariance*, which states that each equivariant layer should be designed to be subject to the group action (e.g., we should be able to "translate" or "rotate" the signal in each layer), and that when an input data is transformed by a particular group element, then its representation in an equivariant layer should also be transformed according to the same group element. While it is easy to see that convolutional layers in CNNs are equivariant to translations, Cohen and Welling [7] formalized the concept of group equivariance CNN (G-CNN) for more general groups and showed in particular how to design convolutional layers equivariant not only to translations but also to reflections and to a discrete set of rotations. Following this seminal work, the theoretical foundations of group equivariant neural networks were then expanded, going beyond regular representations [9], for more groups [2, 18, 37, 40], in less regular spaces [8, 10] or with more general results on their generality and universality [11, 13, 14, 22]. The main applications were developed with the groups of rotations in 2D and 3D, mostly to computer vision problems, but also in biology with histopathology [17, 23], medicine [41] and quantum chemistry [33].

In this paper, we explore and study the potential benefits of equivariant architectures for an important class of data, namely deoxyribonucleic acid (DNA) sequences. DNA is the major form of genetic material in most organisms, from bacteria to mammals, which encodes in particular all proteins that a cell can produce and which is transmitted from generation to generation. The study of DNA in humans and various organisms has led to tremendous progress in biology and medicine since the 1970s, when the first DNA sequencing technologies were invented, and the collapsing cost of sequencing in the last twenty years has accelerated the production of DNA sequences: there are for example about 2.8 billion sequences for a total length of $\sim 10^{13}$ nucleotides publicly available at the European Nucleotide Archive (ENA[1]). Unsurprisingly in such a data-rich field, machine learning-based approaches are increasingly used to analyze DNA sequences, e.g., in metagenomics to automatically predict the species present in an environment from randomly sequenced DNA fragments [26, 28, 36, 38] and to detect the presence of viral DNA in human samples [36], in functional genomics to predict the presence of protein binding sites or other regulatory elements in DNA sequences of interest [16, 24, 30, 35, 43, 44], to predict epigenetic modifications [25], or to predict the effect of variations in the DNA sequence on a phenotype of interest [1, 46].

Due to the sequential nature of DNA and the translation-equivariant nature of the questions addressed, many of these works are based on 1D CNN architectures, although recently transformer-based language models have also shown promising results on various tasks [6, 20, 42]. However, besides translation, DNA has an additional fundamental symmetry that has been largely ignored so far: the so-called reverse complement (RC) symmetry, due to the fact that DNA is made of two strands oriented in opposite direction and encoding complementary nucleotides. In other words, a given DNA segment can be sequenced as two RC DNA sequences, depending on which strand is sequenced; any predictive model for, e.g., DNA sequence classification should therefore be RC-invariant, which calls for RC-equivariant architectures. While strategies based on data augmentation and prediction averaging has been commonly used to handle the need for RC invariance [1, 32], one translation- and RC-equivariant CNN architecture has been proposed and led to promising results [4, 29, 34]. However, it remains unclear whether that architecture is the only one that can encode translation- and RC-equivariance, or if alternative models exist to complement the toolbox of users wishing to develop deep learning models for DNA sequences.

Using the general theory of equivariant representations, in particular steerable CNNs [9], we answer that question by characterizing the set of all linear translation- and RC-equivariant layers. We show in particular that new architectures exist beyond the ones already explored by [4, 29, 34], which in the language of equivariant CNNs only make use of the regular representation [7] while more general representations lead to different layers. We further discuss RC-equivariant pointwise nonlinearities adapted to different representations, as well as RC-equivariant embeddings of k-mers as an alternative to one-hot encoding of nucleotides. We test the new architecture on several protein binding prediction problems, and show experimentally that the new models can outperform existing ones, confirming the potential benefit of exploring the full set of RC-equivariant layers when manipulating DNA sequences with deep neural networks.

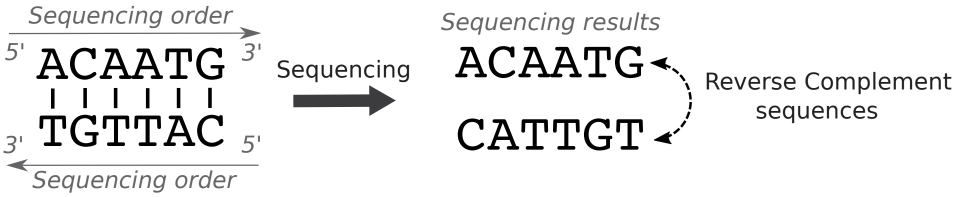

Figure 1: Illustration of the reverse-complement symmetry. Both DNA strands get sequenced in opposite directions resulting in redundant information.

2 Methods

2.1 Group action of translation and reverse complementarity on DNA sequence

DNA is a long polymer made of two intertwined strands, forming the well-known double-helical structure. Each strand is a non-symmetric polymer that can be described as an oriented chain of four possible monomers called nucleotides and denoted respectively $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$. The two strands are oriented in opposite directions, and their nucleotides face each other to form hydrogen bonds. They interact at each position in a deterministic way because only two nucleotides pairings can happen: (A,T) and (G,C). Thus, given a nucleotide sequence on one strand, we can deduce the so-called RC sequence of its corresponding strand by complementing each nucleotide and reversing the order (**Figure 1**). When a double-stranded DNA fragment is sequenced, the two strands are first separated and, typically, only one of them is randomly selected and is decrypted by the machine. This implies that any given DNA fragment can be equivalently described by two RC sequences of nucleotides. Moreover, several genomic learning tasks amount to a sequence annotation that does not depend on the strand. For example, a protein can bind a double-stranded DNA fragment, and both strands of the bound part can get sequenced. This motivates the search for equivariance to this RC-action for the prediction functions. Moreover, the sequencing often results in long sequences where the relevant parts of the sequence do not correlate with their position. The task of prediction over genomic sequences is thus largely translation equivariant, which explains why the community settled on the use of CNNs to train and predict on arbitrary length segments.

To formalize mathematically the translation and RC operations on DNA sequences, we first encode the raw genetic sequence as a signal function in $F_0 = \left\{f : \mathbb{Z} \to \{0,1\}^4\right\}$, as the one hot encoding of the nucleotide content for each integer position. Because of the finite length of this polymer, we assume that beyond a compact support this function takes a constant value of zero. The group $(\mathbb{Z},+)$ of translations acts naturally on this encoding by $T_u(f)(x) = f(x-u)$, for a translation $u \in \mathbb{Z}$, and the RC operations amounts to the following : $RC(f)(x) = \sigma(-1)[f(-x)]$, where $\sigma(-1)$ is the 4×4 permutation matrix that exchanges complementary bases A/T and C/G (while we denote by $\sigma(1)$ the 4×4 identity matrix). We notice that RC is a linear operation on F_0 that satisfies $RC^2 = I$, and thus that the RC operation is a group representation on F_0 for the group $\mathbb{Z}_2 = \{1,-1\}$ endowed with multiplication.

To jointly consider translations and RC actions, we naturally consider the semi-direct product group $G = \mathbb{Z} \rtimes \mathbb{Z}_2$. Elements $g \in G$ can be written as $g = ts$ with $t \in \mathbb{Z}$, $s \in \mathbb{Z}_2$ and the group G acts on F_0 by the action π_0 defined by:

$$\forall ts \in G\,, \quad \forall (f,x) \in F_0 \times \mathbb{Z}\,, \quad \big(\pi_0(ts)f\big)(x) = \sigma(s)[f(s(x-t))]\,.$$

In other words, π_0 is the representation of G on F_0 induced by the representation σ of RC on $\mathbb{R}^4$ [9].

2.2 Features spaces of equivariant layers

Let us now describe the structure of intermediate layers of a neural network equivariant to translations and RC. Following the theory of steerable CNNs [9], we consider successive representations of the input DNA sequence in the following way:

Definition 1. *Given ρ a representation of $\mathbb{Z}_2$ on $\mathbb{R}^D$ for some $D \in \mathbb{N}^*$, a ρ-*feature space *is the set of signals $F = \{f : \mathbb{Z} \to \mathbb{R}^D\}$ endowed with the G group action π, known as the representation*

[1] As of May, 2021: https://www.ebi.ac.uk/ena

induced by ρ :

$$\forall ts \in G\,, \quad \forall (f,x) \in F \times \mathbb{Z}\,, \quad \big(\pi(ts)f\big)(x) = \rho(s)[f(s(x-t))]\,. \tag{1}$$

With this definition, we see in particular that the one-hot encoding input layer maps the input DNA sequence to a σ-feature space, and that the dimension (i.e., number of channels in the language of deep learning) and group action of ρ-feature space are fully characterized by the representation ρ. Interestingly, the theory of linear group representations allows us to characterize more precisely *all* such representations:

Theorem 1. *For any representation ρ of $\mathbb{Z}_2$ on $\mathbb{R}^D$, there exist $a, b \in \mathbb{N}$ such that $a + b = D$ and an invertible matrix $P \in GL(\mathbb{R}^D)$ such that*

$$\forall s \in \mathbb{Z}_2\,, \quad \rho(s) = P\,Diag(I_a, sI_b)P^{-1}\,.$$

In other words, combining Definition 1 and Theorem 1, we see that any ρ-feature space that we will use to build translation- and RC-equivariant layers is fully characterized by a triplet (P, a, b), which we call its *type*, and which characterizes both its dimension $D = a + b$ and the action of the group G by (1). By slight abuse of language, we also refer to (P, a, b) as the type of ρ.

Theorem 1 is a standard result of group theory, which explicits the decomposition of any representation ρ in terms of so-called irreducible representation, or *irreps*. In the case of $\mathbb{Z}_2$, there are exactly two irreps which act on $\mathbb{R}$, namely, $\rho_1(s) = 1$ and $\rho_{-1}(s) = s$. If ρ has type (P, a, b), then it means that it can be decomposed as a times $\rho_1(s)$ and b times $\rho_{-1}(s)$. In the particular case where P is the identity matrix, i.e., when we consider a type (I, a, b), then $\rho(s)$ is a diagonal matrix for any $s \in \mathbb{Z}_2$, and each channel of F is acted upon by a single irrep. In that case, we will call the channels of type "1" (resp. "-1") if they are acted upon by ρ_1 (resp. ρ_{-1}), and we will say that F is an "irrep feature space".

Now, let us introduce another special case. Since $\mathbb{Z}_2$ is finite of cardinality 2, let us consider the *regular representation* ρ_{reg} of $\mathbb{Z}_2$ on $\mathbb{R}^2$ defined by:

$$\rho_{reg}(1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \rho_{reg}(-1) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

One can easily check that ρ_{reg} is of type $(P_{reg}, 1, 1)$, where $P_{reg} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. It corresponds to a ρ-feature space with two channels, where the RC operations flips the two channels (and of course the sequence coordinates).

Let us now consider feature spaces of interest. In the input layer, nucleotides are one-hot encoded in a certain order, let us say $(\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C})$. As stated above, this input space is acted upon by σ, a $2-$cycle that swaps bases A/T and C/G. We see that we can rewrite $\sigma = (\rho_{reg} \oplus \rho_{reg}) := (\rho_{reg}^{\oplus 2})$, where $\oplus$ is the bloc-diagonal operation. Because ρ_{reg} is of type $(P_{reg}, 1, 1)$, we can diagonalize σ with $(P_{reg}^{\oplus 2})$ and the diagonal would be alternated +1 and -1 values. Thus, there exists a permutation Π such that σ is of type $(P, 2, 2)$, with $P = \Pi(P_{reg}^{\oplus 2})\Pi^{-1}$. These concepts are illustrated in Supplementary Section A.1

Interestingly, all RC-equivariant layers proposed so far in [4, 29, 34] follow a similar pattern: the channels go by pair, and the RC action amounts to flipping the channel values within a pair and reversing the sequence coordinates. In our formalism, this corresponds to channels of type (P, a, a), where $a \in \mathbb{N}^*$ is the number of pairs of channels, and where up to a permutation of channels the matrix P satisfies $P = \Pi(P_{reg}^{\oplus a})\Pi^{-1}$. Following [34], we will refer to these layers as *Reverse Complement Parameter Sharing* (RCPS) layers below.

This highlights the fact that translation- and RC-equivariant layers explored so far are equivariant according to Definition 1, but that there exists potentially many other equivariant layers, obtained in particular by allowing ρ-feature spaces of types (P, a, b) where $a \neq b$, on the one hand, and where P is not a direct sum of P_{reg}, on the other hand. We investigate such variants below.

2.3 Equivariant linear layers

While Definition 1 characterizes ρ-feature space in terms of structure and group action, an equivariant multilayer neural network is built by stacking ρ-feature spaces on top of each other and connecting them with equivariant layers. Cohen et al. [11, Theorem 2] gives us a general result about such equivariant mappings. Here, we apply this result to our specific data and group, and characterize the class of equivariant linear layers, i.e., the linear functions $\phi : F_n \to F_{n+1}$ that satisfy $\pi_{n+1}\phi = \phi\pi_n$, where π_n and π_{n+1} are respectively the group action on F_n and F_{n+1}.

Theorem 2. *Given two representations ρ_n and ρ_{n+1} of $\mathbb{Z}_2$, of respective types (P_n, a_n, b_n) and $(P_{n+1}, a_{n+1}, b_{n+1})$ with $a_n + b_n = D_n$ and $a_{n+1} + b_{n+1} = D_{n+1}$, and respective ρ_n- and ρ_{n+1}-feature spaces F_n and F_{n+1}, a linear map $\phi : F_n \to F_{n+1}$ is equivariant if and only if it can be written as a convolution:*

$$\forall (f, x) \in F_n \times \mathbb{Z}\,, \quad \phi(f)(x) = \sum_{y\in\mathbb{Z}} \kappa(y - x) f(y)\,, \tag{2}$$

where the kernel $\kappa : \mathbb{Z} \to \mathbb{R}^{D_{n+1}\times D_n}$ satisfies:

$$\forall x \in \mathbb{Z}\,, \quad \kappa(-x) = \rho_{n+1}(-1)\kappa(x)\rho_n(-1)\,, \tag{3}$$

or equivalently:

$$\forall x \in \mathbb{Z}\,, \quad \kappa(x) = P_{n+1} \begin{pmatrix} \alpha(x) & \beta(x) \\ \gamma(x) & \delta(x) \end{pmatrix} P_n^{-1}\,, \tag{4}$$

where $\alpha : \mathbb{Z} \to \mathbb{R}^{a_{n+1}\times a_n}$ and $\delta : \mathbb{Z} \to \mathbb{R}^{b_{n+1}\times b_n}$ are even, while $\beta : \mathbb{Z} \to \mathbb{R}^{a_{n+1}\times b_n}$ and $\gamma : \mathbb{Z} \to \mathbb{R}^{b_{n+1}\times a_n}$ are odd functions.

As stated in Cohen et al. [11], "Convolution is all you need" to define linear layers which are equivariant to our group. In addition, Theorem 2 characterizes all the convolution kernels that ensure equivariance through the two equivalent constraints (3) and (4).

To illustrate this result, let us consider two RCPS feature spaces F_n and F_{n+1} of respective types $(\Pi_n(P_{reg}^{\oplus a_n})\Pi_n^{-1}, a_n, a_n)$ and $(\Pi_{n+1}(P_{reg}^{\oplus a_{n+1}})\Pi_{n+1}^{-1}, a_{n+1}, a_{n+1})$. Then, the channels in F_n and F_{n+1} go by pair, and if we consider a slice $\tilde{\kappa} : \mathbb{Z} \to \mathbb{R}^{2\times 2}$ of the convolution kernel κ describing how a pair of channels in F_n maps to a pair of channels in F_{n+1}, (3) gives the constraint:

$$\tilde{\kappa}(-x) := \begin{pmatrix} \tilde{\kappa}_{11}(-x) & \tilde{\kappa}_{12}(-x) \\ \tilde{\kappa}_{21}(-x) & \tilde{\kappa}_{22}(-x) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tilde{\kappa}(x) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \tilde{\kappa}_{22}(x) & \tilde{\kappa}_{21}(x) \\ \tilde{\kappa}_{12}(x) & \tilde{\kappa}_{11}(x) \end{pmatrix}.$$

We recover exactly the constraints of the RCPS filters first proposed by [34], proving as a consequence of Theorem 2 that RCPS convolution filters describe exactly *all* equivariant linear mappings between RCPS feature spaces.

Moreover, if we now consider any two feature spaces F_n and F_{n+1} of respective types (P_n, a_n, b_n) and $(P_{n+1}, a_{n+1}, b_{n+1})$, then Equation (4) tells us that up to multiplications by matrices P_{n+1} and P_n^{-1}, the kernel is expressed in terms of even and odd functions, which can be trivially implemented with parameter sharing. For example, to represent the even function α, one just need to parameterize the values of $\alpha(x)$ for $x \geq 0$, and complete the negative values by parameter sharing $\alpha(-x) = \alpha(x)$. Hence, the parameter sharing idea used in RCPS [34] extends to any equivariant linear map.

Instead of using (4) to parameterize equivariant convolution kernels, one may also directly write the constraints (3) for specific representations, and potentially save the need of multiplication by P_{n+1} and P_n^{-1} in (4). This is for example the case in RCPS layers [34], and more generally for channels acted upon by the regular representation; for the sake of completeness, we derive in Appendix A.4 the constraints to go from and to the regular representation or the irreps, and use them in our implementation.

2.4 Equivariant nonlinear layers

Besides equivariant linear layers, a crucial component needed for multilayer neural networks is the possibility to have equivariant nonlinear layers, such as nonlinear pointwise activation functions or batch normalization [19]. In this section, we discuss particular nonlinearities that are adapted to various equivariant layers.

Pointwise activations. Let us begin with pointwise transformations, that encompass most activation functions used in deep learning. Pointwise transformations are formally defined as follows: given a function $\theta : \mathbb{R} \to \mathbb{R}$ and a vector space $V = \mathbb{R}^A$ for some index set A, the pointwise extension of θ to V is the mapping $\bar{\theta}_V : V \to V$ defined by $\bar{\theta}_V(f)(a) = \theta(f(a))$, for any $(f, a) \in V \times A$. For a D-dimensional representation ρ of $\mathbb{Z}_2$ and a ρ-feature space F with G-group action π, we say that a pointwise extension $\bar{\theta}_F : F \to F$ is equivariant if it commutes with π, i.e., $\pi\bar{\theta}_F = \bar{\theta}_F\pi$. By definition of the group action (1), this is equivalent to saying that the pointwise extension $\bar{\theta}_{\mathbb{R}^D}$ of θ to $\mathbb{R}^D$ commutes with ρ. The following theorem gives an exhaustive characterization of a large class of equivariant pointwise extensions for any ρ-feature space:

Theorem 3. *Let ρ be a representation of $\mathbb{Z}_2$ and $\theta : \mathbb{R} \to \mathbb{R}$ be a continuous function with left and right derivatives at 0. Let F be a ρ-layer and $\bar{\theta}_F : F \to F$ be the point-wise extension of θ on this layer. Then $\bar{\theta}_F$ is equivariant if and only if at least one of the following cases holds:*

1. *θ is a linear function.*
2. *θ is an affine function, and $\rho(-1)\mathbf{1} = \mathbf{1}$.*
3. *θ is not an affine function, and there exists a permutation matrix Π, integers $a, b, c, d \in \mathbb{N}$, and scalars $(\lambda_1, \ldots, \lambda_a) \in (\mathbb{R}_+^*)^a$, such that ρ decomposes as*

$$\Pi^{-1}\rho(-1)\Pi = \bigoplus_{i=1}^{a} \begin{pmatrix} 0 & \lambda_i \\ \lambda_i^{-1} & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}^{\oplus b} \oplus (1)^{\oplus c} \oplus (-1)^{\oplus d}. \tag{5}$$

In that case,

- *Either $b = d = 0$ and $\forall i, \lambda_i = 1$ and θ is any function,*
- *Or $b = d = 0$ and $\exists i, \lambda_i \neq 1$ and θ is a leaky ReLu function.*[2]
- *Or $b + d > 0$ and $\forall i, \lambda_i = 1$ and θ is an odd function,*

The first case in Theorem 3 is of little interest, since pointwise linear functions are always equivariant to linear group actions. The second case essentially says that adding a constant to a pointwise linear function is only equivariant for representations ρ such that the sum of all rows of $\rho(-1)$ is equal to 1. This holds for example for the regular representation and the RCPS layers, but not for an irrep feature space of type (I, a, b) with $b > 0$, since in that case, some rows have a single "-1" entry. The most interesting case is the third one, since it describes what pointwise nonlinearities one can apply. The condition (5) on the decomposition of ρ essentially excludes all representations that have more than one nonzero value in at least one row of $\rho(-1)$. Among valid ρ's that decompose as (5), we see that the regular representation (corresponding to the first block in (5) with $\lambda_i = 1$)), used in RCPS, stands out as the only that allows *any* nonlinearity, besides of course invariant channels of type "+1" (third block in (5)). Replacing a "1" in the regular representation by a scalar $\lambda_i \neq 1$ (in the first block of (5), with $b = d = 0$) creates a valid representation ρ, however only leaky ReLu pointwise nonlinearities can be applied in that case. Another case of practical interest is the irrep feature space of type (I, c, d) for some $c > 0$ and $d > 0$. By Theorem 3, only odd nonlinearities are allowed in that case, such as the hyperbolic tangent function. Finally, one should keep in mind that other representations, which do not satisfy the conditions listed in Theorem 3, do not allow any equivariant nonlinear pointwise transform; this is for example the case of $\rho(-1) = \begin{pmatrix} 0 & -1/2 \\ -2 & 0 \end{pmatrix}$, which is a valid representation of $\mathbb{Z}_2$ but does neither meet the condition to accept affine activations (case 2), nor to accept nonlinear activations (case 3) because $\rho(-1)$ does not decompose according to (5).

Other activation functions Besides pointwise transformations from a ρ-feature space to itself characterized in Theorem 3, the set of nonlinear equivariant layer is tremendous and the design choices are endless. A first extension is to keep pointwise activation, but to allow different nonlinearities on different channels, e.g., by using any function on the "+1" channels and an odd function on the "-1" channels of an irrep feature space. Another relaxation is to use different input and output representations. While odd functions will not affect the field type, even functions will turn a field of type "-1" into a "+1" type. It is well known that any function decomposes into a sum of an odd and even function. Therefore, given ρ, a representation decomposed as in (5), any point-wise non-linearity can be used in a ρ-feature space by first decomposing it into its odd and even components and applying each component separately for the second and fourth blocks.

[2] A leaky ReLu function is $\theta(x) = \alpha_{sign(x)}x$ for some $(\alpha_+, \alpha_-) \in \mathbb{R}^2$.

Other possibilities exist and include creating new representations by tensorization, which amounts to taking pointwise products between different channels [13, 21, 37]. or using non point-wise activation layers, that act on several coupled dimensions, such as the ones used in [37]. For instance, we could apply the max function to paired channels. These possibilities are discussed in [39]

Batch normalization An equivariant batch normalization was introduced by [34]. It considers a feature map and its reverse complement as two instances, which is easy to do because the reverse complement feature map is already computed when using regular representation. We propose another batch normalization for irrep feature spaces that also gives the result we would have had if the batch contained all the reverse complement of its sequences. For the "+1" dimensions, it amounts to scaling as we would have the same values twice. For the "-1" dimensions, we enforce a zero mean and compute a variance estimate based on this constraint.

K-mers. Instead of the standard one-hot encoding of individual nucleotides as input layer, we propose to one-hot encode k-mers for $k \geq 1$, i.e., overlapping blocks of k consecutive nucleotides. This technique is known to improve performance in several tasks [27, 28]. In order to implement it into an equivariant network, we need to know how the group acts on the k-mers space, made of 4^k elements. The simplest idea is to pair the index of the channels of two RC k-mers. Because some k-mers are their own reverse complement, the canonical way to do so is to have a representation that is a blend of "+1" irrep and regular representation. An alternative is to make the regular representation act on the k-mers instead by redundantly encoding these k-mers into paired dimensions. This is the strategy we follow in our implementation, to be more coherent with the usual input group action.

3 Experiments

We assess the performance of various equivariant architectures on a set of three binary prediction and four sequence prediction problems used by Zhou et al. [45] to assess the performance of RCPS networks. The binary classification problems aim to predict if a DNA sequence binds to three transcription factors (TFs), based on genome-wide binarized TF-ChIP-seq data for Max, Ctcf and Spi1 in the GM12878 lymphoblastoid cell-line [34]. The sequence prediction problems aim to predict TF binding at the base-pair resolution, using genome-wide ChIP-nexus profiles of four TFs-Oct4, Sox2, Nanog and Klf4 in mouse embryonic stem cells. For a more detailed explanation of the experimental setup, please refer to Zhou et al. [45]. We report "significant" differences in performance below when the P-value of a Wilcoxon signed rank test is smaller than 0.05.

Models. We build over the work of Zhou et al. [45] for both the binary and the sequence prediction problems. They benchmarked an equivariant RCPS architecture and a corresponding non-equivariant model, with the same number of filters and trained with data augmentation, which we respectively refer to as "RCPS" and "Standard" models below. The data augmentation scheme for the "Standard" model consists in adding to the training set the reverse complement sequences of all training sequences, which is a natural procedure to let the model "learn" the equivariance without encoding it explicitly in the architecture of the network. We checked empirically that data augmentation significantly improves the performance of non-equivariant models (Appendix A.6.1). In addition, we extend the RCPS architecture with one-hot encoding of k-mers as input layers, which we refer to as "Regular" below. Finally, we add to the comparison a new equivariant network where each RCPS layer is replaced by an (I, a, b) layer with the same number of filters, which we call "Irrep" below. We also use k-mers and vary the ratio $a/(a + b)$ in this model. We combine the regular and "+1" dimensions with *ReLu* activations and the "-1" dimensions with a *tanh* activation.

Influence of hyperparameters in equivariant models To assess the impact of different hyperparameters in the family of equivariant models we propose (k-mer length for Irrep and Regular, $a/(a+b)$ ratio for Irrep), we train equivariant models with different combinations of hyperparameters on the training set and assess their performance on the validation set, repeating the process ten times with different random seeds. We assess the performance of each run in terms of Area under the Receiver Operator Characteristic (AuROC), and show in Figure 2 the average performance reached by all runs with a given ratio $a/(a + b) \in \{0, 1/4, 1/2, 3/4, 1\}$ (left) and with a given $k \in \{1, 2, 3, 4\}$ (right). We see a clear asymmetry in the performance as a function of $a/(a + b)$, with poor performance when $a = 0$ and optimal performance for $a = 0.75$, significantly better than all other ratios tested.

This confirms that exploring different irreps may be valuable. As for the k-mer length, setting $k = 3$ gives the best performance and significantly outperforms all other values of k tested. This confirms that going beyond one-hot encoding of nucleotides in equivariant architectures can be beneficial.

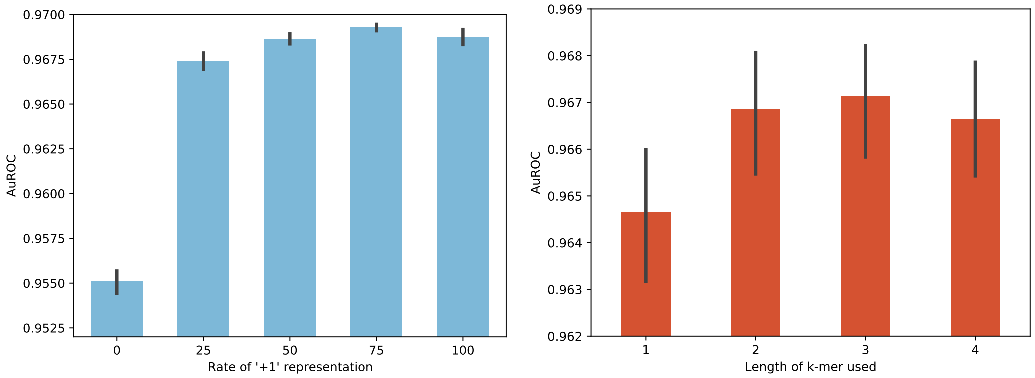

Figure 2: Average AuROC performance across four TFs and 10 random seeds for the Irrep model as a function of $a/(a+b)$ (*left*, also averaged over k values) and for the Irrep and Regular models as a function of k (*right*, also averaged over $a/(a+b)$ values for Irrep).

Binary task. We then compare the test set performance of three different models for the binary classification task: 1) Standard, 2) RCPS, and 3) the best Irrep or Regular equivariant model, where hyperparameters are selected based on the AuROC on the validation set, which we denote as "Best Equivariant". Figure 3 (left) shows the performance of each model on each TF task and overall. As already observed by [34], the equivariant RCPS architecture has a strong lead over the Standard, non-equivariant model in spite of data augmentation. Interestingly, we see that Best Equivariant is significantly better than RCPS on all tasks, and that the performance gain from RCPS to Best equivariant is of the same order as the performance gain from Standard to RCPS. This demonstrates that the family of equivariant architectures we introduce in this paper can lead to significant improvement over existing architectures.

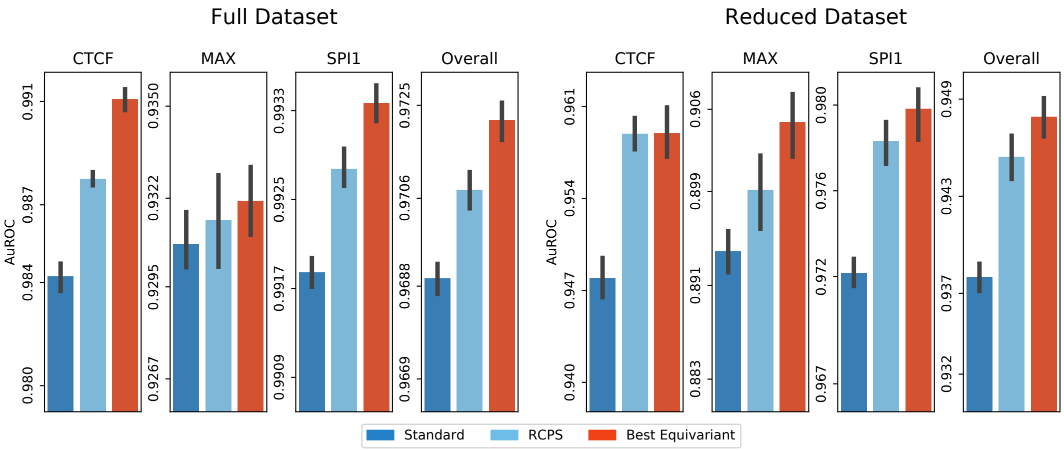

Figure 3: AuROC performance of the three different models (Standard, RCPS and Best equivariant after hyperparameter selection on the validation set) on the three binary classification problems CTCF, MAX and SPI1, as well as their average. Error bars correspond to an estimate of the standard error on 10 repeats with different random seeds. The left plot is the performance on the full datasets, while the right plot shows the performance where models are trained on a subset of 1,000 sequences only (notice the differences of AuROC values on the vertical axis in both plots).

Reduced models. Since equivariant architectures are meant to be particularly beneficial in the low-data regime [15], we further assess the performance of the three models on the same binary classification problems but with only 1,000 sequences used to train the models, and show the results on Figure 3 (right). Overall, the performances are worse than in the full-data regime (Figure 3, left), which confirms that this is a regime where more data helps. We also see that the relative order of the three different methods remains overall the same, with Best Equivariant outperforming RCPS, which itself outperforms Standard. Interestingly, the gaps between the best and worse models widens in the low-data regime, showing that the prior is more useful in this setting. More precisely, there is a large gap of about 1% between Best Equivariant and Standard in the low data regime, compared to a gap of about 0.3% on the full dataset. We also investigated whether equivariant models converge faster to their solutions, but found not noticeable difference (Appendix A.6.2).

On post-hoc models. Zhou et al. [45] introduced the so-called *post-hoc* model, another equivariant method obtained by averaging the predictions of a Standard model over a sequence and its reverse-complement, and showed that it is competitive with and often outperforms RCPS. The post-hoc model only requires training and storing one network, but aggregates two predictions for each sequence at inference time. Because of that, the good performance of post-hoc may be due in part to the aggregation step common to all ensemble models [12]. To decipher the respective contributions of the network architecture, on the one hand, and of the aggregation of predictions, on the other hand, we add to the comparison an ensemble of two Standard models trained with different random seeds (*Ensemble Standard*) and an ensemble of two equivariant Irrep models (*Ensemble Irrep*) and present the results in Figure 4. We see that Ensemble Irrep strongly outperforms Best Equivariant, and both post-hoc and Ensemble Standard widely outperform the Standard architecture. This confirms that ensembling equivariant or non-equivariant models through post-hoc of ensemble aggregation is always useful (at the cost of increased computational time). We see that Ensemble Standard is not significantly different from post-hoc Standard on CTCF and SPI1, but that post-hoc Standard is better on MAX, suggesting that most of the benefits of post-hoc Standard indeed comes from the ensembling effect. Regarding the impact of the architecture for a given budget of predictions, we saw earlier than Best equivariant significantly outperforms Standard when a single prediction per test sequence is allowed, and see now that Ensemble Irrep strongly outperforms both post-hoc and Ensemble Standard when two predictions are allowed, thus confirming the benefit of equivariant architectures in all settings. We also see that a single Best equivariant models outperforms post-hoc and Ensemble Standard, indicating that enforcing equivariance throughout the network is not only faster but also more more accurate than averaging a non-equivariant model over group transformed inputs.

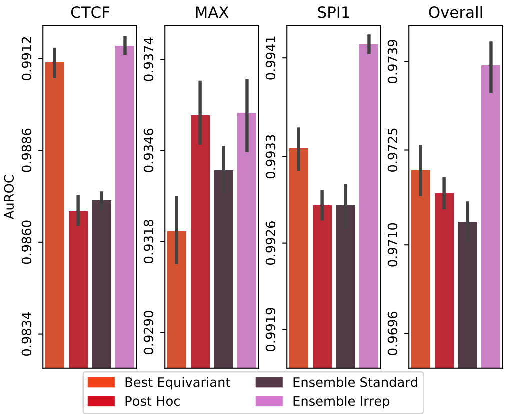

Figure 4: AuROC performance on the three binary classification problems, for the Best Equivariant model, the post-hoc Standard model, and an ensemble of two Standard or Irrep models. Error bars correspond to an estimate of the standard error on 10 repeats with different random seeds.

Profile task. We now compare the performance of different models on the profile prediction tasks. To limit the carbon footprint of this study, and based on the influence of hyperparameters on the binary task (Figure 2), we only test two equivariant models in addition to Standard and RCPS: a Regular model with $k = 3$, and an Irrep model with $k = 3$ and $a/(a + b) = 75\%$. We also assess the performance of post-hoc Standard (the best model in [45]), and an ensemble of two models of the best performing equivariant model. Figure 5 shows the performance of all models in terms of Spearman correlation between the target profile and the predicted ones, on the full dataset (left) or a reduced experiment with only 1,000 training sequences (right).

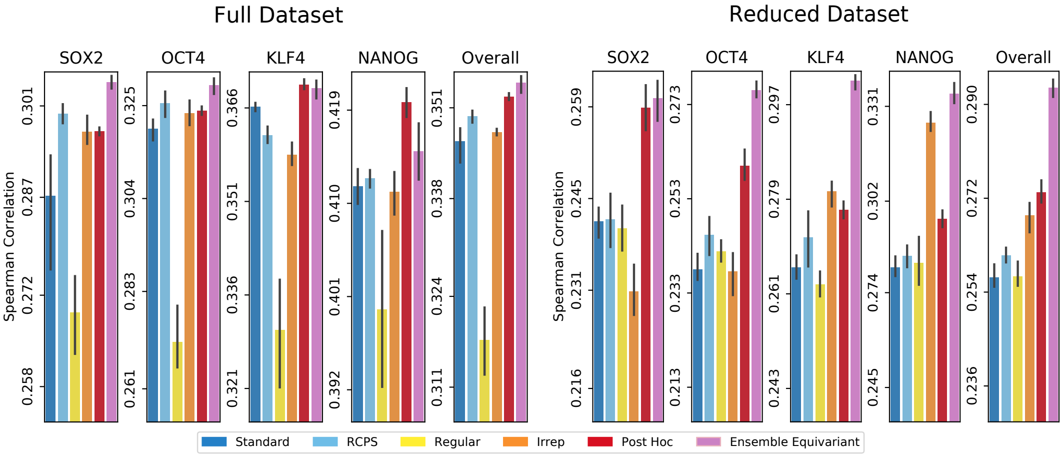

Figure 5: Spearman Correlation between true and predicted profiles by different methods for four data sets.

First of all, we observe as before that in the low-data regime, the gap between standard and equivariant networks grows in favor of equivariant ones. We also observe, surprisingly, that Irrep, which outperformed RCPS on the binary task, now underperforms it. A possible explanation could be that since this task aims to annotate an individual nucleotide, encoding the nucleotide level information using k-mers makes the signal blurry and decreases performance. However, in the reduced setting, Irrep performs better again. These results indicate that for now the best model should be chosen empirically on a validation set. Finally, despite good performance of post-hoc Standard, the ensemble equivariant model once again performs better for the same computational cost at inference.

Experiment settings and computational cost. All experiments were run on a single GPU (either a GTX1080 or a RTX6000), with 20 CPU cores. The binary classification experiments were shorter to train. To limit our carbon footprint, we chose to run more experiments on this task, e.g., for hyperparameter tuning and to reduce the number of replicates for the profile task. The total runtimes of each of those tasks were approximately of a week.

4 Conclusion

In this paper, we addressed the problem of including the RC symmetry prior in neural networks. Leveraging the framework of equivariant networks, in particular steerable CNNs, we deepened existing methods by unraveling the whole space of linear layers and pointwise nonlinearities that are translation and RC-equivariant. We also investigated the links between the linear representations and the non-linear layers of neural networks, exposing the special role of the regular representation in equivariant networks. Finally, we implemented new linear and nonlinear equivariant layers and make all these equivariant layers available in Keras [5] and Pytorch [31]. [3] We then explored empirically how this larger equivariant functional space behaves in terms of learning. Our best results improve the state of the art performance of equivariant networks, showing that new equivariant architectures can have practical benefits. In the future we plan to test more deeply the newly proposed architectures on prediction tasks involving double-stranded DNA, such as DNA-protein binding prediction, epigenetics or metagenomics. On the theoretical side, we characterized equivariant pointwise nonlinearities that preserve the layer type, but more general nonlinear transforms (e.g., not pointwise, or changing the layer type) remain to be fully characterized.

[3] code available at `https://github.com/Vincentx15/Equi-RC`

Acknowledgments and Disclosure of Funding

V.M. is recipient of a doctoral fellowship from the INCEPTION project [PIA/ANR-16-CONV-0005] and benefits from support from the CRI through Ecole Doctorale FIRE - Programme Bettencourt. We thank Marie Dechelle, Jacques Boitreaud, Carlos G. Oliver and Guillaume Bouvier for reviewing the manuscript. We thank Avanti Shrikumar, Hannah Zhou and Anshul Kundaje for helpful discussions and sharing their code.

Conflict of Interest: None declared.

References

[1] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

[2] B. Anderson, T. S. Hy, and R. Kondor. Cormorant: Covariant molecular neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. Technical Report 2104.13478, arXiv, 2021.

[4] R. C. Brown and G. Lunter. An equivariant Bayesian convolutional network predicts recombination hotspots and accurately resolves binding motifs. *Bioinformatics*, 35(13):2177–2184, 2019.

[5] F. Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[6] J. Clauwaert and W. Waegeman. Novel transformer networks for improved sequence labeling in genomics. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 2021.

[7] T. Cohen and M. Welling. Group equivariant convolutional networks. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR.

[8] T. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2019.

[9] T. S. Cohen and M. Welling. Steerable CNNs. In *International Conference on Learning Representations (ICLR)*, 2017.

[10] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. Spherical CNNs. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[11] T. S. Cohen, M. Geiger, and M. Weiler. A General Theory of Equivariant CNNs on Homogeneous Spaces. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[12] T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, page 1–15, Berlin, Heidelberg, 2000. Springer-Verlag.

[13] N. Dym and H. Maron. On the Universality of Rotation Equivariant Point Cloud Networks. *arXiv preprint arXiv:2010.02449*, 2020.

[14] C. Esteves. Theoretical aspects of group equivariant neural networks. *arXiv preprint arXiv:2004.05154*, 2020.

[15] F. Fuchs, D. Worrall, V. Fischer, and M. Welling. SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1970–1981. Curran Associates, Inc., 2020.

[16] M. Ghandi, D. Lee, M. Mohammad-Noori, and M. A. Beer. Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Comput Biol*, 10(7):e1003711, 2014.

[17] S. Graham, D. Epstein, and N. Rajpoot. Dense steerable filter cnns for exploiting rotational symmetry in histology images. *IEEE Transactions on Medical Imaging*, 39(12):4124–4136, 2020.

[18] E. Hoogeboom, J. W. T. Peters, T. S. Cohen, and M. Welling. Hexaconv. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[20] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, Feb. 2021.

[21] R. Kondor. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. *arXiv preprint arXiv:1803.01588*, 2018.

[22] R. Kondor and S. Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755. PMLR, 2018.

[23] M. W. Lafarge, E. J. Bekkers, J. P. Pluim, R. Duits, and M. Veta. Roto-translation equivariant convolutional networks: Application to histopathology image analysis. *Medical Image Analysis*, 68:101849, 2021.

[24] D. Lee, D. U. Gorkin, M. Baker, B. J. Strober, A. L. Asoni, A. S. McCallion, and M. A. Beer. A method to predict the impact of regulatory variants from DNA sequence. *Nature genetics*, 47 (8):955, 2015.

[25] J. J. Levy, A. J. Titus, C. L. Petersen, Y. Chen, L. A. Salas, and B. C. Christensen. MethylNet: an automated and modular deep learning approach for DNA methylation analysis. *BMC bioinformatics*, 21:108, 2020.

[26] Q. Liang, P. W. Bible, Y. Liu, B. Zou, and L. Wei. DeepMicrobes: taxonomic classification for metagenomics with deep learning. *NAR genom. bioinform.*, 2:lqaa009, Mar. 2020.

[27] W. Liang. Segmenting DNA sequence into words based on statistical language model. *Nature Precedings*, pages 1–1, 2012.

[28] R. Menegaux and J.-P. Vert. Continuous embeddings of DNA sequencing reads and application to metagenomics. *Journal of Computational Biology*, 26(6):509–518, 2019.

[29] K. Onimaru, O. Nishimura, and S. Kuraku. Predicting gene regulatory regions with a convolutional neural network for processing double-strand genome sequence information. *PloS one*, 15 (7):e0235748, 2020.

[30] M. Oubounyt, Z. Louadi, H. Tayara, and K. T. Chong. DeePromoter: Robust promoter predictor using deep learning. *Frontiers in genetics*, 10:286, 2019.

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[32] D. Quang and X. Xie. Factornet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods*, 166:40–47, 2019.

[33] K. T. Schütt, O. T. Unke, and M. Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021.

[34] A. Shrikumar, P. Greenside, and A. Kundaje. Reverse-complement parameter sharing improves deep learning models for genomics. *bioRxiv*, page 103663, 2017.

[35] G. D. Stormo. DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, 2000.

[36] A. Tampuu, Z. Bzhalava, J. Dillner, and R. Vicente. ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples. *PloS one*, 14:e0222271, 2019.

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

[38] K. Vervier, P. Mahé, M. Tournoud, J.-B. Veyrieras, and J.-P. Vert. Large-scale machine learning for metagenomics sequence classification. *Bioinformatics*, 32:1023–1032, 2016.

[39] M. Weiler and G. Cesa. General $e(2)$-equivariant steerable cnns. *arXiv preprint arXiv:1911.08251*, 2019.

[40] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *Advances in Neural Information Processing Systems*, pages 10381–10392, 2018.

[41] M. Winkels and T. S. Cohen. Pulmonary nodule detection in ct scans with equivariant cnns. *Medical image analysis*, 55:15–26, 2019.

[42] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontañón, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. Big Bird: Transformers for Longer Sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[43] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford. Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics*, 32(12):i121–i127, 2016.

[44] H. Zhang, C.-L. Hung, M. Liu, X. Hu, and Y.-Y. Lin. NCNet: Deep learning network models for predicting function of non-coding DNA. *Frontiers in genetics*, 10:432, 2019.

[45] H. Zhou, A. Shrikumar, and A. Kundaje. Towards a better understanding of reverse-complement equivariance for deep learning models in regulatory genomics. *bioRxiv*, 2020.11.04.368803, 2020.

[46] J. Zhou and O. G. Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

A Appendix

A.1 Illustration of group actions

This section is intended to provide a visual, more intuitive understanding of the different group actions on the tensors of our network. We begin with a visualization of the group action for the input space. We exemplify it over the sequence GGACT, whose reverse complement is AGTCC. The sequence is one hot encoded as explained in the main text and the group action over $\mathbb{Z}_2$ consist in flipping the tensor along the spatial axis and swapping the channels pairwise.

$$\begin{tikzcd}[ampersand replacement=\&]
\begin{array}{c}A\\C\\G\\T\end{array}\begin{bmatrix}0&0&1&0&0\\0&0&0&1&0\\1&1&0&0&0\\0&0&0&0&1\end{bmatrix} \arrow[r, bend left, "\pi(-1)"] \& \begin{bmatrix}1&0&0&0&0\\0&0&0&1&1\\0&1&0&0&0\\0&0&1&0&1\end{bmatrix}\begin{array}{c}A\\C\\G\\T\end{array} \arrow[l, bend left, "\pi(-1)\circ\pi(-1)=I"]
\end{tikzcd}$$

Now we illustrate the actions of other representations, on an example tensor $\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}$ with two channels (of type a or b) and three positions; this could typically be the representation of an input sequence of length 3 in an intermediate layer of dimention 2. Choosing the canonical representations of type $(I,2,0)$, $(I,0,2)$ and $(I,1,1)$ respectively, we get the following group actions (for clarity we add the channel type, a or b, near each matrix row):

$$\begin{tikzcd}[ampersand replacement=\&]
\begin{array}{c}a\\a\end{array}\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix} \arrow[r, bend left, "\pi(-1)"] \& \begin{bmatrix}3&2&1\\6&5&4\end{bmatrix}\begin{array}{c}a\\a\end{array} \arrow[l, bend left, "\pi(-1)\circ\pi(-1)=I"]
\end{tikzcd}$$

$$\begin{tikzcd}[ampersand replacement=\&]
\begin{array}{c}b\\b\end{array}\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix} \arrow[r, bend left, "\pi(-1)"] \& \begin{bmatrix}-3&-2&-1\\-6&-5&-4\end{bmatrix}\begin{array}{c}b\\b\end{array} \arrow[l, bend left, "\pi(-1)\circ\pi(-1)=I"]
\end{tikzcd}$$

$$\begin{tikzcd}[ampersand replacement=\&]
\begin{array}{c}a\\b\end{array}\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix} \arrow[r, bend left, "\pi(-1)"] \& \begin{bmatrix}3&2&1\\-6&-5&-4\end{bmatrix}\begin{array}{c}a\\b\end{array} \arrow[l, bend left, "\pi(-1)\circ\pi(-1)=I"]
\end{tikzcd}$$

Finally, when using different values for P, we can get other group actions. As mentioned in the main text, by choosing $(P_{reg},1,1)$, where $P_{reg}=\begin{bmatrix}1&1\\1&-1\end{bmatrix}$, we get the regular representation that flips the input channel. We also provide an example of the group action for a general P matrix, by choosing $(P_{general}, 1, 1)$, where $P_{general} = \begin{bmatrix} 1 & 3 \\ 1 & -1 \end{bmatrix}$, we get a representation on the fibers $\rho_{general} = \begin{bmatrix} -0.5 & 1.5 \\ 0.5 & 0.5 \end{bmatrix}$

$$\begin{tikzcd}
\textit{Reg} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \arrow[r, bend left, "\pi(-1)"] & \begin{bmatrix} 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix} \textit{Reg} \arrow[l, bend left, "\pi(-1) \circ \pi(-1) = I"]
\end{tikzcd}$$

$$\begin{tikzcd}
\textit{General} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \arrow[r, bend left, "\pi(-1)"] & \begin{bmatrix} 5.5 & 6.5 & 7.5 \\ 2.5 & 3.5 & 4.5 \end{bmatrix} \textit{General} \arrow[l, bend left, "\pi(-1) \circ \pi(-1) = I"]
\end{tikzcd}$$

Over the course of these examples, we have limited ourselves to the case where the input tensor had only three nucleotides and two channels, but this is coincidental. The representation with arbitrary P can mix an arbitrary number of channels together with the group action.

A.2 Proof of Theorem 1

Proof. The irreducible representations (irreps) of the 2-elements group $\mathbb{Z}_2$ are the 1-dimensional trivial and sign representations, given respectively by $\rho_1(s) = 1$ and $\rho_1(s) = s$. Any representation ρ_n can be decomposed as a direct sum of irreps, and since each irrep is 1-dimensional this means that there exists an invertible matrix P such that $P\rho_n(s)P^{-1}$ is diagonal, with diagonal terms either equal to 1 or equal to s. If we denote by a_n (resp. b_n) the number of diagonal terms equal to 1 (resp. s), then Theorem 1 follows. □

A.3 Proof of Theorem 2

Proof. Cohen et al. [11, Theorem 3.3] gives a general result about linear equivariant mapping. We first show that this result can be applied here, to show that these linear mappings are exactly the ones written as (2) and (3). For sake of clarity, we then provide a fully self-contained proof of the same result.

Let us first show that (2) and (3) correspond to a particular case of Cohen et al. [11, Theorem 3.3]. Under the notations of [11], our group is $G = \mathbb{Z} \rtimes \mathbb{Z}_2$, a locally compact, semi-direct product group. We choose $H = H_1 = H_2 = \mathbb{Z}_2$, making the coset space $G/H = \mathbb{Z}$. Since our group is a semi direct product group, we have $h_1(x, s) = s$. The spaces F_n that we have considered are signals in $\mathbb{R}^D$ over the coset space, acted upon by the representation induced by ρ. Equivalently, they are sections of the associated vector bundle for the trivial case of a product group. Therefore, these F_n exactly coincide with the setting of Cohen et al. [11, Theorem 3.3] and $\{\phi : F_n \to F_{n+1} | \pi_{n+1}\phi = \phi\pi_n\}$ is exactly $\mathcal{H}$. Then, by [11, Theorem 3.3], $\phi : F_n \to F_{n+1}$ is equivariant if and only if it can be written as a convolution:

$$\forall (f, x) \in F_n \times \mathbb{Z}\,, \quad \phi(f)(x) = \sum_{y \in \mathbb{Z}} \kappa(y - x) f(y)\,, \tag{2}$$

where the kernel $\kappa : \mathbb{Z} \to \mathbb{R}^{D_{n+1} \times D_n}$ satisfies:

$$\forall x \in \mathbb{Z}\, s \in \mathbb{Z}_2, , \quad \kappa(sx) = \rho_{n+1}(s)\kappa(x)\rho_n(s^{-1})\,. \tag{6}$$

Using that for $s \in \mathbb{Z}_2$, $s^{-1} = s$, and the triviality of this equation for $s = 1$, we get that (6) is equivalent to (3)

For sake of clarity and completeness, we now provide a more explicit and self-contained proof for (2) and (3), that follows the one of [40, Theorem 2] in our specific setting. We first notice that any linear mapping $\phi; F_n \to F_{n+1}$ can be written as

$$\forall (f,x) \in F_n \times \mathbb{Z}\,, \quad \phi(f)(x) = \sum_{y\in\mathbb{Z}} k(x,y) f(y)\,,$$

for some function $k : \mathbb{Z}^2 \to \mathbb{R}^{d_{n+1}\times d_n}$. For any $g = ts \in G$, the action of G on F_{n+1} gives:

$$\begin{aligned}\forall (f,x) \in F_n \times \mathbb{Z}\,, \quad \pi_{n+1}(g)\phi(f)(x) &= \rho_{n+1}(s)\phi(f)(s(x-t)) \\ &= \rho_{n+1}(s) \sum_{y\in\mathbb{Z}} k(s(x-t),y) f(y)\,.\end{aligned} \tag{7}$$

Similarly, the action of G on F_n followed by ϕ gives:

$$\begin{aligned}\forall (f,x) \in F_n \times \mathbb{Z}\,, \quad \phi(\pi_n(g)f)(x) &= \sum_{y\in\mathbb{Z}} k(x,y)\pi_n(g)f(y) \\ &= \sum_{y\in\mathbb{Z}} k(x,y)\rho_n(s)f(s(y-t)) \\ &= \sum_{y\in\mathbb{Z}} k(x,sy+t)\rho_n(s)f(y)\end{aligned} \tag{8}$$

where we made the change of variable $y \mapsto sy+t$ to get the last equality. ϕ is equivariant if and only if, for any $g \in G$, $\phi \circ \pi_n(g) = \pi_{n+1}(g) \circ \phi$, which from (7) and (8) is equivalent to:

$$\forall (f,x) \in F_n \times \mathbb{Z}\,, \quad \rho_{n+1}(s) \sum_{y\in\mathbb{Z}} k(s(x-t),y) f(y) = \sum_{y\in\mathbb{Z}} k(x,sy+t)\rho_n(s)f(y)\,. \tag{9}$$

For any $y_0 \in \mathbb{Z}$ and $v \in \mathbb{R}^{D_n}$, let us apply this equality to the function $f \in F_n$ given by $f(y_0) = v$ and $f(y) = 0$ for $y \neq y_0$:

$$\forall (x,y_0,v) \in \mathbb{Z}\times\mathbb{Z}\times\mathbb{R}^{D_n}\,, \quad \rho_{n+1}(s)k(s(x-t),y_0)v = k(x,sy_0+t)\rho_n(s)v\,.$$

Since this must hold for any $v \in \mathbb{R}^{D_n}$ this necessarily implies:

$$\forall (x,y_0) \in \mathbb{Z}^2\,, \quad \rho_{n+1}(s)k(s(x-t),y_0) = k(x,sy_0+t)\rho_n(s)\,.$$

With the change of variable $y = s(y_0 - t)$, this is equivalent to:

$$\forall (x,y) \in \mathbb{Z}^2\,, \quad \rho_{n+1}(s)k(s(x-t),s(y-t)) = k(x,y)\rho_n(s)\,,$$

which itself is equivalent to

$$\forall (x,y) \in \mathbb{Z}^2\,, \quad k(s(x-t),s(y-t)) = \rho_{n+1}(s)k(x,y)\rho_n(s)\,, \tag{10}$$

where we used the fact that $\rho_{n+1}(s)^2 = \rho_{n+1}(s^2) = I$ for any $s \in \mathbb{Z}_2$. This must hold in particular for $s = 1$ and $t = x$, which gives:

$$\forall (x,y) \in \mathbb{Z}^2\,, \quad k(0,y-x) = k(x,y)\,,$$

i.e., k is necessarily translation invariant in the sense that there must exist a function $\kappa : \mathbb{Z} \to \mathbb{R}^{D_{n+1}\times D_n}$ such that

$$\forall (x,y) \in \mathbb{Z}^2\,, \quad k(x,y) = \kappa(y-x)\,.$$

From (10) we see that κ must satisfy

$$\forall (x,y) \in \mathbb{Z}^2\,, \quad \kappa(s(y-x)) = \rho_{n+1}(s)\kappa(y-x)\rho_n(s)\,,$$

which boils down to the following constraint, after observing that the constraint is always true for $s = 1$ and is therefore only nontrivial for $s = -1$:

$$\forall x \in \mathbb{Z}\,, \quad \kappa(-x) = \rho_{n+1}(-1)\kappa(x)\rho_n(-1)\,. \tag{11}$$

At this point, we have therefore shown that an equivariant linear function must have an expansion of the form

$$\forall (f,x) \in F_n \times \mathbb{Z}\,, \quad \phi(f)(x) = \sum_{y\in\mathbb{Z}} \kappa(y-x)f(y)\,,$$

where κ must satisfy (11). Conversely, such a linear layer trivially satisfies (9), and is therefore equivariant. This proves (2) and (3).

To prove (4), we simply rewrite (3) using Theorem 1:

$$\forall x \in \mathbb{Z}\,, \quad \kappa(-x) = P_{n+1}\text{Diag}(I_{a_{n+1}}, -I_{b_{n+1}})P_{n+1}^{-1}\kappa(x)P_n\text{Diag}(I_{a_n}, -I_{b_n})P_n^{-1}\,. \tag{12}$$

Thus writing the matrix $K = P_{n+1}^{-1}\kappa(x)P_n$ by blocs of sizes $a_{n+1} \times a_n$, $a_{n+1} \times b_n$, $b_{n+1} \times a_n$ and $b_{n+1} \times b_n$, we have :

$$\begin{aligned}(12) \iff K(-x) &= \text{Diag}(I_{a_{n+1}}, -I_{b_{n+1}})K(x)\text{Diag}(I_{a_n}, -I_{b_n}) \\ \iff \begin{bmatrix}\alpha(-x) & \beta(-x) \\ \gamma(-x) & \delta(-x)\end{bmatrix} &= \begin{bmatrix}\alpha(x) & -\beta(x) \\ -\gamma(x) & \delta(x)\end{bmatrix}\end{aligned}$$

This gives us the equivalence (3) $\iff$ (12) $\iff$ (4). □

A.4 Resolution of the constraint for other basis

To go from an arbitrary representation (P, a, b) to another, we can write an odd/even kernel and change of basis. One may also solve the constraints (3) for specific representations, and save the need of multiplication by P_{n+1} and P_n^{-1} in (4). In this section, we solve the constraint in other basis, to go from one kind of representation (irrep or regular) to another. We just substitute the correct representation and see what constrained kernel it gives. The irrep and regular representations are in a basis such that they write as :

$$\rho_{irrep} = \begin{bmatrix}I_a & 0 \\ 0 & -I_b\end{bmatrix}, \quad \rho_{reg} = \begin{bmatrix}0 & 0 & \dots & 1 \\ \vdots & & & \vdots \\ 0 & 1 & \dots & 0 \\ 1 & 0 & \dots & 0\end{bmatrix}.$$

We get the following table of constraints :

F_n \ F_{n+1}	'irrep'	'regular'
'irrep'	$\begin{bmatrix}\alpha(-x) & \beta(-x) \\ \gamma(-x) & \delta(-x)\end{bmatrix} = \begin{bmatrix}\alpha(x) & -\beta(x) \\ -\gamma(x) & \delta(x)\end{bmatrix}$	$[\kappa_{j,a}(-x), \kappa_{j,b}(-x)] = [\kappa_{n-j,a}(x), -\kappa_{n-j,b}(x)]$
'regular'	$\begin{bmatrix}\kappa_{a,j}(-x) \\ \kappa_{b,j}(-x)\end{bmatrix} = \begin{bmatrix}\kappa_{a,n-j}(x) \\ -\kappa_{b,n-j}(x)\end{bmatrix}$	$\kappa_{i,j}(-x) = -\kappa_{n-i,n-j}(x)$ [34]

A.5 Proof of Theorem 3

With a slight abuse of notations, in this section we denote the matrix $\rho(-1)$ simply by $\rho \in \mathbb{R}^{D\times D}$, and for any $\theta : \mathbb{R} \to \mathbb{R}$ we define $\tilde{\theta}(x) := \theta(x) - \theta(0)$. We start with three technical lemmas, before proving Theorem 3.

Lemma 4. *Let $h : \mathbb{R} \to \mathbb{R}$ be a continuous function with left and right derivatives at 0. If there exists $A \in \mathbb{R}$ with $|A| > 1$ such that*

$$\forall x \in \mathbb{R}\,, \quad h(x) = Ah(A^{-1}x)\,, \tag{13}$$

then h is a leaky ReLu function, i.e., there exists $(\alpha_-, \alpha_+) \in \mathbb{R}^2$ such that

$$\forall x \in \mathbb{R}\,, \quad h(x) = \begin{cases}\alpha_- x & \text{if } x \le 0\,, \\ \alpha_+ x & \text{if } x \ge 0\,.\end{cases}$$

In addition, if $A < -1$, then $\alpha_- = \alpha_+$, i.e., h is linear.

Proof. Equation (13) implies $h(0) = 0$ and

$$\forall x \in \mathbb{R}^*\,, \quad \frac{h(x)}{x} = \frac{h(A^{-1}x)}{A^{-1}x}\,,$$

which by simple induction gives more generally:

$$\forall (x,n) \in \mathbb{R}^* \times \mathbb{N}\,, \quad \frac{h(x)}{x} = \frac{h(A^{-n}x)}{A^{-n}x}\,. \tag{14}$$

The right-hand side of (14) for $n = 2k$ converges to $h'_{sign(x)}(0)$ when $k \to +\infty$, which by unicity of the limit must be equal to the left-hand side. As a result, for any $x \in \mathbb{R}$, $h(x) = h'_{sign(x)}(0)x$, i.e., h is a leaky ReLu function with $\alpha_s = h'_s(0)$ for $s \in \{-,+\}$. If in addition $A < -1$, then (14) for $n = 2k+1$ converges to $h'_{-sign(x)}(0)$ when $k \to +\infty$. By unicity of the limit, this implies $h'_-(0) = h'_+(0)$, i.e., $\alpha_- = \alpha_+$. □

Lemma 5. *Under the assumptions of Theorem 3, if $\bar{\theta}_F$ is equivariant and if there exists $(i,j) \in [1,D]^2$ such that $\rho_{ij} \notin \{-1,0,1\}$, then necessarily $\tilde{\theta}$ is a leaky ReLu function.*

Proof. For any (i,j), applying the equivariance constraint $\theta(\rho x)_i = \rho\theta(x)_i$ to the vector $x = ae_j$, for any $a \in \mathbb{R}$, gives the equation:

$$\forall a \in \mathbb{R}\,, \quad \theta(a\rho_{ij}) = \rho_{ij}\theta(a) + (\sum_{k\neq j}\rho_{ik})\theta(0)\,.$$

If $|\rho_{ij}| > 1$, we can rewrite it as

$$\forall a \in \mathbb{R}\,, \quad \theta(a) = \rho_{ij}\theta(a\rho_{ij}^{-1}) + (\sum_{k\neq j}\rho_{ik})\theta(0)\,,$$

and if $0 < |\rho_{ij}| < 1$ we can rewrite it as

$$\forall a \in \mathbb{R}\,, \quad \theta(a) = \rho_{ij}^{-1}\theta(a\rho_{ij}) - \rho_{ij}^{-1}(\sum_{k\neq j}\rho_{ik})\theta(0)\,.$$

In both cases, this is an equation of the form

$$\forall a \in \mathbb{R}\,, \quad \theta(a) = A\theta(A^{-1}a) + B\,,$$

where $|A| > 1$. Subtracting to this equation the same equation written for $a = 0$ gives

$$\forall a \in \mathbb{R}\,, \quad \tilde{\theta}(a) = A\tilde{\theta}(A^{-1}a)\,. \tag{15}$$

By Lemma 4, $\tilde{\theta}$ is a leaky ReLu function.

□

Lemma 6. *Under the assumptions of Theorem 3, if $\bar{\theta}_F$ is equivariant and if there exists at least one row in ρ with at least two nonzero entry, then necessarily θ is an affine function.*

Proof. Let us suppose that ρ contains at least a row i with two nonzero entries, say $\rho_{ij} \neq 0$ and $\rho_{ik} \neq 0$. Then taking $x = x_je_j + x_ke_k$ with $x_j, x_k \in \mathbb{R}$, the equivariance constraint for the i-th dimension gives

$$\forall x_j, x_k \in \mathbb{R}\,, \quad \theta(\rho_{ij}x_j + \rho_{ik}x_k) = \rho_{ij}\theta(x_j) + \rho_{ik}\theta(x_k) + C\theta(0)\,,$$

with $C = \sum_{p\notin\{j,k\}}\rho_{ip}$. Subtracting to this equation the same equation written for $x_j = x_k = 0$ allows us to remove the constant term and get

$$\forall x_j, x_k \in \mathbb{R}\,, \quad \tilde{\theta}(\rho_{ij}x_j + \rho_{ik}x_k) = \rho_{ij}\tilde{\theta}(x_j) + \rho_{ik}\tilde{\theta}(x_k)\,. \tag{16}$$

We now prove that $\tilde{\theta}$ is necessarily a leaky ReLu function, i.e., that there exist $(\alpha_+, \alpha_-) \in \mathbb{R}^2$ such that $\tilde{\theta}(x) = \alpha_{sign(x)}x$, with potentially $\alpha_+ \neq \alpha_-$. By Lemma 5 this is true if $|\rho_{ij}| \neq 1$ or $|\rho_{ik}| \neq 1$, so we focus on the case $|\rho_{ij}| = |\rho_{ik}| = 1$, which we decompose in two subcases. First, if $\rho_{ij} = \rho_{ik} = s$ with $s \in \{-1, 1\}$, then taking $x_j = x_k = a$ in (16) gives $\tilde{\theta}(2sa) = 2s\tilde{\theta}(a)$, for any $a \in \mathbb{R}$. Second, if $\rho_{ij} = -\rho_{ik} = 1$ (resp. $\rho_{ij} = -\rho_{ik} = 1$), then taking $x_j = 2a$ and $x_k = a$ (resp.

$x_j = a$ and $x_k = 2a$) gives $\tilde{\theta}(2a) = 2\tilde{\theta}(a)$. In both subcases, by Lemma 4, $\tilde{\theta}$ must be a leaky ReLu function.

Knowing that $\tilde{\theta}$ is a leaky ReLu function with coefficients α_+ and α_-, in order to prove that θ is necessarily an affine function (i.e., that $\tilde{\theta}$ is linear), we need to show that $\alpha_+ = \alpha_-$. For that purpose, let us first suppose that ρ_{ij} and ρ_{ik} are both positive or both negative. Then there exists a pair $(x_j, x_k) \in \mathbb{R}^2$ such that $x_j > 0$, $x_k < 0$ and $\rho_{ij}x_j + \rho_{ik}x_k < 0$. Similarly, if ρ_{ij} and ρ_{ik} are of different signs, say without loss of generality $\rho_{ij} < 0$ and $\rho_{ik} > 0$, then any pair $(x_j, x_k) \in \mathbb{R}^2$ such that $x_j > 0$, $x_k < 0$ satisfies $\rho_{ij}x_j + \rho_{ik}x_k < 0$. In both cases, using the fact that $\tilde{\theta}$ is linear on $\mathbb{R}_+$ and on $\mathbb{R}_-$, (16) gives

$$\begin{aligned} & \alpha_-(\rho_{ij}x_j + \rho_{ik}x_k) = \alpha_+\rho_{ij}x_j + \alpha_-\rho_{ik}x_k \,, \\ \iff & \alpha_-\rho_{ij}x_j = \alpha_+\rho_{ij}x_j \\ \iff & \alpha_- = \alpha_+ \,. \end{aligned}$$

□

We are now ready to prove Theorem 3.

Proof of Theorem 3. To characterize the functions θ and representations ρ such that $\bar{\theta}_F$ is equivariant, we proceed by a disjunction of cases on θ, depending on whether it is affine.

If θ is affine, say $\theta(x) = \alpha x + \beta$, then $\bar{\theta}_F$ is equivariant if and only if, for any $x \in \mathbb{R}^D$, $\bar{\theta}_{\mathbb{R}^D}(\rho x) = \rho\bar{\theta}_{\mathbb{R}^D}(x)$. This is equivalent to

$$\begin{aligned} & \forall (i, x) \in [1, d] \times \mathbb{R}^D \,, \quad \sum_{j=1}^{D} \rho_{i,j}\theta(x_j) = \theta\left(\sum_{j=1}^{D} \rho_{i,j}x_j\right) \\ \iff & \forall (i, x) \in [1, d] \times \mathbb{R}^D \,, \quad \sum_{j=1}^{D} \rho_{i,j}(\alpha x_j + \beta) = \alpha\left(\sum_{j=1}^{D} \rho_{i,j}x_j\right) + \beta \\ \iff & \forall i \in [1, d] \,, \quad \beta\left(\sum_{j=1}^{D} \rho_{i,j} - 1\right) = 0 \,. \end{aligned}$$

This shows that if θ is affine, then $\bar{\theta}_F$ is equivariant if and only $\beta = 0$, i.e., θ is linear (case 1 of Theorem 3), or $\rho\mathbf{1} = \mathbf{1}$ (case 2 of Theorem 3).

If θ is not affine and $\bar{\theta}_F$ is equivariant, then by Lemma 6 we know that ρ can have at most one nonzero entry per row. Since ρ is invertible, it must have at least one nonzero entry per row, so we conclude that if contains exactly one nonzero entry per row, hence a total of D nonzero entries. Being invertible, it must also contain at least one nonzero entry per column, so we conclude that it contains also exactly one nonzero entry per column. Using the fact that $\rho^2 = I$, we can further clarify how nonzero entries must be organized:

- For a nonzero entry $\rho_{ii} \neq 0$ on the diagonal, we must have $\rho_{ii}^2 = 1$, i.e., $\rho_{ii} \in \{-1, +1\}$.
- For an off-diagonal nonzero entry $\rho_{ij} \neq 0$ with $i \neq j$, we must have $\rho_{ij}\rho_{ji} = 1$, i.e., $\rho_{ji} = \rho_{ij}^{-1}$.

Splitting the nonzero entries by sign, this implies that there exists a permutation matrix Π such that

$$\hat{\rho} := \Pi^{-1}\rho(-1)\Pi = \bigoplus_{i=1}^{a} \begin{pmatrix} 0 & \lambda_i \\ \lambda_i^{-1} & 0 \end{pmatrix} \oplus \bigoplus_{i=1}^{b} \begin{pmatrix} 0 & -\mu_j \\ -\mu_j^{-1} & 0 \end{pmatrix} \oplus (1)^{\oplus c} \oplus (-1)^{\oplus d} \,, \tag{17}$$

for some $(a, b, c, d) \in \mathbb{N}^4$ such that $a + b + c + d = D$ and $(\lambda, \mu) \in \mathbb{R}_+^a \times \mathbb{R}_+^b$. For any $i \in [1, D]$, let us now denote by $\tau(i)$ the column corresponding to the nonzero entry of the i-th row of $\hat{\rho}$, i.e., the only index such that $\hat{\rho}_{i\tau(i)} \neq 0$. Then the action of $\hat{\rho}$ on a vector $v \in \mathbb{R}^D$ has the simple form $[\hat{\rho}v]_i = \hat{\rho}_{i\tau(i)} v_{\tau(i)}$. By writing the equivariance property $\rho \circ \bar{\theta}_F = \bar{\theta}_F \circ \rho$ coordinate by coordinate, we can therefore say that $\bar{\theta}_F$ is equivariant if and only if:

$$\forall (i,x) \in [1,D] \times \mathbb{R}\,, \quad \theta(\hat{\rho}_{i\tau(i)} x) = \hat{\rho}_{i\tau(i)} \theta(x)\,. \tag{18}$$

Let us now consider two possible cases:

- If there exists $i \in [1,D]$ such that $|\hat{\rho}_{i\tau(i)}| \neq 1$, then by Lemma 5 $\tilde{\theta}$ is a leaky ReLu function, i.e., there exist $(\alpha_+, \alpha_-, \beta) \in \mathbb{R}^3$ such that $\forall x \in \mathbb{R}\,, \theta(x) = \alpha_{sign(x)} x + \beta$. In that case, by (18), $\bar{\theta}_F$ is equivariant if and only if:

$$\begin{aligned} &\forall (i,x) \in [1,D] \times \mathbb{R}\,, \quad \alpha_{sign(\hat{\rho}_{i\tau(i)} x)} \hat{\rho}_{i\tau(i)} x + \beta = \hat{\rho}_{i\tau(i)} \left(\alpha_{sign(x)} x + \beta \right)\,, \\ \iff & \forall i \in [1,D]\,, \quad \begin{cases} \alpha_{sign(\hat{\rho}_{i\tau(i)})} &= \alpha_+\,, \\ \alpha_{sign(-\hat{\rho}_{i\tau(i)})} &= \alpha_-\,, \\ \beta = \hat{\rho}_{i\tau(i)} \beta\,, \end{cases} \\ \iff & \begin{cases} \forall i \in [1,D]\,, \quad \alpha_{sign(\hat{\rho}_{i\tau(i)})} &= \alpha_+\,, \\ \beta = 0\,, \end{cases} \end{aligned} \tag{19}$$

  where the first equivalence comes from identifying the coefficients of the linear equation in x on $\mathbb{R}_-$ and $\mathbb{R}_+$, and the second equivalence comes from the observation that the two conditions in α in the first equivalence are themselves equivalent to each other, so we can keep only one of them, and that the condition on β is equivalent to $\beta = 0$ since we assume the existence of an $i \in [1,D]$ such that $\hat{\rho}_{i\tau(i)} \neq 1$. Since we assume that θ is not affine, we can not have $\alpha_- = \alpha_+$, which by (19) rules out the possibility of having negative entries in $\hat{\rho}$, i.e., necessarily $b = d = 0$ in (17). If that is not the case, then the condition on α in (19) is automatically met for all $i \in [1,D]$, so we have that $\bar{\theta}_F$ is equivariant if and only if $\beta = 0$, i.e., if and only if θ is a leaky ReLu function. This is the second statement in Case 3 of Theorem 3, when we further notice that when $b = 0$ the only entry in $\hat{\rho}$ that can have been different from -1 and 1 is a λ_i in (17).

- If for all $i \in [1,D]$, $|\hat{\rho}_{i\tau(i)}| = 1$, then (17) simplifies as

$$\hat{\rho} = \bigoplus_{i=1}^{a} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \oplus \bigoplus_{i=1}^{b} \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \oplus (1)^{\oplus c} \oplus (-1)^{\oplus d}\,.$$

  In that case, the equivariance condition (18) is particularly simple, and true for any θ for positive values. For each i such that $\hat{\rho}_{i\tau(i)} = -1$ it reads $\forall x \in \mathbb{R}, -\theta(x) = \theta(-x)$, and is therefore true if and only if θ is odd. Noticing that the latter constraint occurs if and only if $b + d > 0$ finally leads to the first and third statements in Case 3 of Theorem 3.

□

A.6 Additional result

A.6.1 Effect of data augmentation and size for non-equivariant models

Given a non-equivariant model, a simple way to let it "learn" to be equivariant is to train it with data augmentation, where for each sequence in the training set we add its reverse complement to the training set. This doubles the size of the training set, which increases the training time. If we compare such a non-equivariant model with an equivariant model with the same number of channels in each layers, then it has about twice the same number of free parameters to train, and we therefore call it "big"; as an alternative, one may want to restrict the number of channels in each layer to enforce the same number of parameters as the equivariant model. To assess the benefits of data augmentation and number of channels, we plot in Figure 6 the performance of a standard, non-equivariant model with or without data augmentation, and with the same number of channels or half of it, on the binary classification tasks. We see that the number of channels has no significant impact on the performance, but that data augmentation has a significant positive impact. In the main text, we therefore restrict ourselves to the standard model with data augmentation as non-equivariant baseline model.

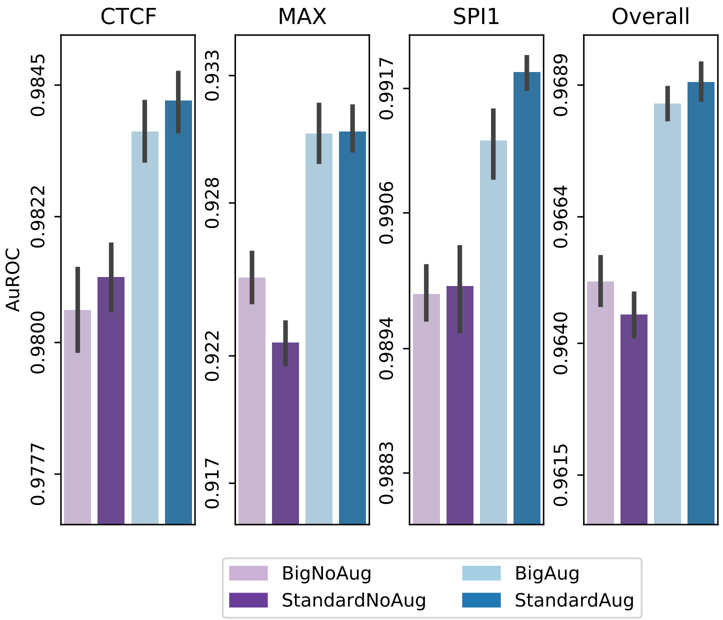

Figure 6: Binary task performance of a standard, non-equivariant model trained with ("Aug") or without ("NoAug") data augmentation, and with more ("Big") or less ("Standard") channels.

A.6.2 Comparison of learning curves

Because equivariant model are supposed to converge faster, we looked into the learning curves of our models, i.e., how the test performance increases as a function of the number of epochs during training. However, we do not see a major difference in the learning dynamics between the equivariant and non equivariant models.

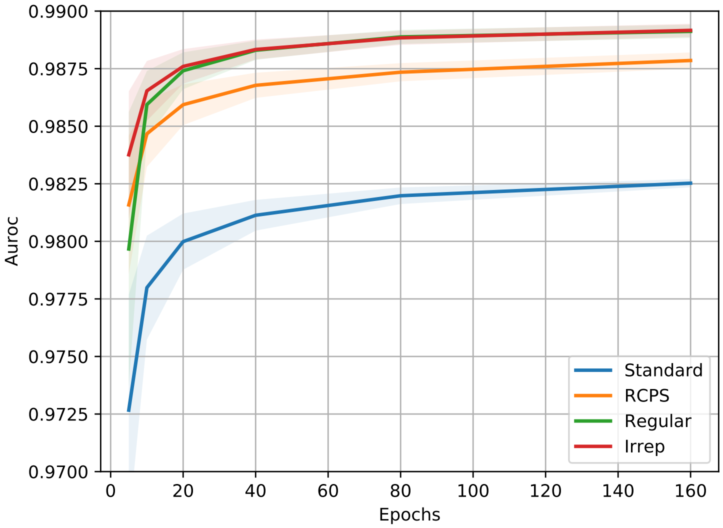

Figure 7: AuROC performance of the four different models on the three binary classification problems CTCF, MAX and SPI1, as well as their average over the course of learning.