

Figure 2: We consider a Lipschitz function $f(x) = \sum_{j=1}^{2^d} z_j \phi(x - x_j)$ where $z_j = \pm 1$, $x_j \in \mathbb{R}^d$ is placed in each quadrant, and ϕ a locally supported Lipschitz ‘bump’. Unless we observe the function in most of the 2^d quadrants, we will incur in a constant error in predicting it. This simple geometric argument can be formalised through the notion of *Maximum Discrepancy* (von Luxburg and Bousquet, 2004), defined for the Lipschitz class as $\kappa(d) = \mathbb{E}_{x, x'} \sup_{f \in \text{Lip}(1)} \left| \frac{1}{N} \sum_l f(x_l) - \frac{1}{N} \sum_l f(x'_l) \right| \simeq N^{-1/d}$, which measures the largest expected discrepancy between two independent N -sample expectations. Ensuring that $\kappa(d) \simeq \epsilon$ requires $N = \Theta(\epsilon^{-d})$; the corresponding sample $\{x_l\}_l$ defines an ϵ -net of the domain. For a d -dimensional Euclidean domain of diameter 1, its size grows exponentially as ϵ^{-d} .

3 Geometric Priors

Modern data analysis is synonymous with high-dimensional learning. While the simple arguments of Section 2.1 reveal the impossibility of learning from generic high-dimensional data as a result of the curse of dimensionality, there is hope for physically-structured data, where we can employ two fundamental principles: *symmetry* and *scale separation*. In the settings considered in this text, this additional structure will usually come from the structure of the domain underlying the input signals: we will assume that our machine learning system operates on *signals* (functions) on some domain Ω . While in many cases linear combinations of points on Ω is not well-defined, we can linearly combine signals on it, i.e., the space of signals forms a vector space. Moreover, since we can define an inner product between signals, this space is a *Hilbert space*.

Ω must be a vector space in order for an expression $\alpha u + \beta v$ to make sense.

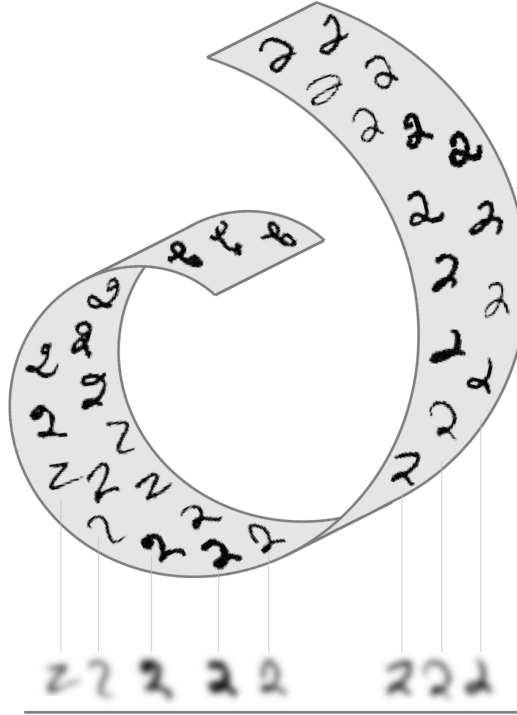


Figure 3: If the unknown function f is presumed to be well approximated as $f(\mathbf{x}) \approx g(\mathbf{A}\mathbf{x})$ for some unknown $\mathbf{A} \in \mathbb{R}^{k \times d}$ with $k \ll d$, then shallow neural networks can capture this inductive bias, see e.g. [Bach \(2017\)](#). In typical applications, such dependency on low-dimensional projections is unrealistic, as illustrated in this example: a low-pass filter projects the input images to a low-dimensional subspace; while it conveys most of the semantics, substantial information is lost.

The space of \mathcal{C} -valued signals on Ω (for Ω a set, possibly with additional structure, and \mathcal{C} a vector space, whose dimensions are called *channels*)

$$\mathcal{X}(\Omega, \mathcal{C}) = \{x : \Omega \rightarrow \mathcal{C}\} \quad (1)$$

is a function space that has a vector space structure. Addition and scalar multiplication of signals is defined as:

$$(\alpha x + \beta y)(u) = \alpha x(u) + \beta y(u) \quad \text{for all } u \in \Omega,$$

with real scalars α, β . Given an inner product $\langle v, w \rangle_{\mathcal{C}}$ on \mathcal{C} and a measure μ on Ω (with respect to which we can define an integral), we can define an inner product on $\mathcal{X}(\Omega, \mathcal{C})$ as

$$\langle x, y \rangle = \int_{\Omega} \langle x(u), y(u) \rangle_{\mathcal{C}} d\mu(u). \quad (2)$$

When Ω has some additional structure, we may further restrict the kinds of signals in $\mathcal{X}(\Omega, \mathcal{C})$. For example, when Ω is a smooth manifold, we may require the signals to be smooth. Whenever possible, we will omit the range \mathcal{C} for brevity.

When the domain Ω is discrete, μ can be chosen as the *counting measure*, in which case the integral becomes a sum. In the following, we will omit the measure and use du for brevity.

As a typical illustration, take $\Omega = \mathbb{Z}_n \times \mathbb{Z}_n$ to be a two-dimensional $n \times n$ grid, x an RGB image (i.e. a signal $x : \Omega \rightarrow \mathbb{R}^3$), and f a function (such as a single-layer Perceptron) operating on $3n^2$ -dimensional inputs. As we will see in the following with greater detail, the domain Ω is usually endowed with certain geometric structure and symmetries. Scale separation results from our ability to preserve important characteristics of the signal when transferring it onto a coarser version of the domain (in our example, subsampling the image by coarsening the underlying grid).

We will show that both principles, to which we will generically refer as *geometric priors*, are prominent in most modern deep learning architectures. In the case of images considered above, geometric priors are built into Convolutional Neural Networks (CNNs) in the form of convolutional filters with shared weights (exploiting translational symmetry) and pooling (exploiting scale separation). Extending these ideas to other domains such as graphs and manifolds and showing how geometric priors emerge from fundamental principles is the main goal of Geometric Deep Learning and the *leitmotif* of our text.

3.1 Symmetries, Representations, and Invariance

Informally, a *symmetry* of an object or system is a transformation that leaves a certain property of said object or system unchanged or *invariant*. Such transformations may be either smooth, continuous, or discrete. Symmetries are ubiquitous in many machine learning tasks. For example, in computer vision the object category is unchanged by shifts, so shifts are symmetries in the problem of visual object classification. In computational chemistry, the task of predicting properties of molecules independently of their orientation in space requires *rotational invariance*. Discrete symmetries emerge naturally when describing particle systems where particles do not have canonical ordering and thus can be arbitrarily permuted, as well as in many dynamical systems, via the time-reversal symmetry (such as systems in detailed balance or the Newton's second law of motion). As we will see in Section 4.1, permutation symmetries are also central to the analysis of graph-structured data.

Symmetry groups The set of symmetries of an object satisfies a number of properties. First, symmetries may be combined to obtain new symmetries: if g and h are two symmetries, then their compositions $g \circ h$ and $h \circ g$ are also symmetries. The reason is that if both transformations leave the object invariant, then so does the composition of transformations, and hence the composition is also a symmetry. Furthermore, symmetries are always invertible, and the inverse is also a symmetry. This shows that the collection of all symmetries form an algebraic object known as a *group*. Since these objects will be a centerpiece of the mathematical model of Geometric Deep Learning, they deserve a formal definition and detailed discussion:

We will follow the juxtaposition notation convention used in group theory, $g \circ h = gh$, which should be read right-to-left: we first apply h and then g . The order is important, as in many cases symmetries are non-commutative. Readers familiar with Lie groups might be disturbed by our choice to use the Fraktur font to denote group elements, as it is a common notation of Lie algebras.

A *group* is a set \mathcal{G} along with a binary operation $\circ : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$ called *composition* (for brevity, denoted by juxtaposition $g \circ h = gh$) satisfying the following axioms:

Associativity: $(gh)\mathfrak{k} = g(h\mathfrak{k})$ for all $g, h, \mathfrak{k} \in \mathcal{G}$.

Identity: there exists a unique $\mathfrak{e} \in \mathcal{G}$ satisfying $\mathfrak{e}g = g\mathfrak{e} = g$ for all $g \in \mathcal{G}$.

Inverse: For each $g \in \mathcal{G}$ there is a unique inverse $g^{-1} \in \mathcal{G}$ such that $gg^{-1} = g^{-1}g = \mathfrak{e}$.

Closure: The group is closed under composition, i.e., for every $g, h \in \mathcal{G}$, we have $gh \in \mathcal{G}$.

Note that *commutativity* is not part of this definition, i.e. we may have $gh \neq hg$. Groups for which $gh = hg$ for all $g, h \in \mathcal{G}$ are called commutative or *Abelian*.

After the Norwegian mathematician Niels Henrik Abel (1802–1829).

Though some groups can be very large and even infinite, they often arise from compositions of just a few elements, called *group generators*. Formally, \mathcal{G} is said to be *generated* by a subset $S \subseteq \mathcal{G}$ (called the *group generator*) if every element $g \in \mathcal{G}$ can be written as a finite composition of the elements of S and their inverses. For instance, the symmetry group of an equilateral triangle (dihedral group D_3) is generated by a 60° rotation and a reflection (Figure 4). The 1D *translation group*, which we will discuss in detail in the following, is generated by infinitesimal displacements; this is an example of a *Lie group* of differentiable symmetries.

Note that here we have defined a group as an abstract object, without saying what the group elements *are* (e.g. transformations of some domain), only how they *compose*. Hence, very different kinds of objects may have the same symmetry group. For instance, the aforementioned group of rotational and reflection symmetries of a triangle is the same as the group of permutations of a sequence of three elements (we can permute the corners in the triangle in any way using a rotation and reflection – see Figure 4).

Lie groups have a differentiable manifold structure. One such example that we will study in Section 4.3 is the special orthogonal group $SO(3)$, which is a 3-dimensional manifold.

The diagram shown in Figure 4 (where each node is associated with a group element, and each arrow with a generator), is known as the *Cayley diagram*.

Group Actions and Group Representations Rather than considering groups as abstract entities, we are mostly interested in how groups *act* on data. Since we assumed that there is some domain Ω underlying our data, we will study how the group acts on Ω (e.g. translation of points of the plane), and from there obtain actions of the same group on the space of signals $\mathcal{X}(\Omega)$ (e.g. translations of planar images and feature maps).

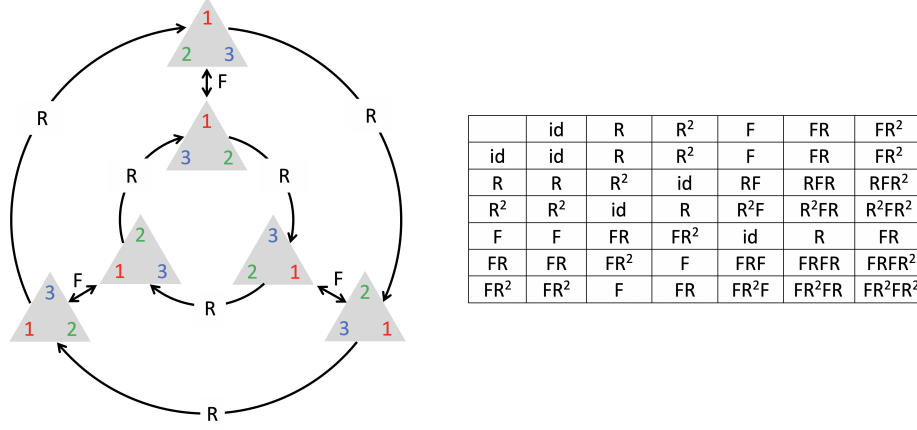


Figure 4: Left: an equilateral triangle with corners labelled by 1, 2, 3, and all possible rotations and reflections of the triangle. The group D_3 of rotation/reflection symmetries of the triangle is generated by only two elements (rotation by 60° R and reflection F) and is the same as the group Σ_3 of permutations of three elements. Right: the multiplication table of the group D_3 . The element in the row g and column h corresponds to the element gh .

Technically, what we define here is a *left* group action.

Distance-preserving transformations are called *isometries*. According to Klein's Erlangen Programme, the classical Euclidean geometry arises from this group.

A *group action* of \mathcal{G} on a set Ω is defined as a mapping $(g, u) \mapsto g.u$ associating a group element $g \in \mathcal{G}$ and a point $u \in \Omega$ with some other point on Ω in a way that is compatible with the group operations, i.e., $g.(h.u) = (gh).u$ for all $g, h \in \mathcal{G}$ and $u \in \Omega$. We shall see numerous instances of group actions in the following sections. For example, in the plane the *Euclidean group* $E(2)$ is the group of transformations of \mathbb{R}^2 that preserves Euclidean distances, and consists of translations, rotations, and reflections. The same group, however, can also act on the space of *images* on the plane (by translating, rotating and flipping the grid of pixels), as well as on the representation spaces learned by a neural network. More precisely, if we have a group \mathcal{G} acting on Ω , we automatically obtain an action of \mathcal{G} on the space $\mathcal{X}(\Omega)$:

$$(g.x)(u) = x(g^{-1}u). \quad (3)$$

Due to the inverse on g , this is indeed a valid group action, in that we have $(g.(h.x))(u) = ((gh).x)(u)$.

The most important kind of group actions, which we will encounter repeatedly throughout this text, are *linear* group actions, also known as *group representations*. The action on signals in equation (3) is indeed linear, in the

sense that

$$\mathbf{g} \cdot (\alpha x + \beta x') = \alpha(\mathbf{g} \cdot x) + \beta(\mathbf{g} \cdot x')$$

for any scalars α, β and signals $x, x' \in \mathcal{X}(\Omega)$. We can describe linear actions either as maps $(\mathbf{g}, x) \mapsto \mathbf{g} \cdot x$ that are linear in x , or equivalently, by currying, as a map $\rho : \mathfrak{G} \rightarrow \mathbb{R}^{n \times n}$ that assigns to each group element \mathbf{g} an (invertible) matrix $\rho(\mathbf{g})$. The dimension n of the matrix is in general arbitrary and not necessarily related to the dimensionality of the group or the dimensionality of Ω , but in applications to deep learning n will usually be the dimensionality of the feature space on which the group acts. For instance, we may have the group of 2D translations acting on a space of images with n pixels.

As with a general group action, the assignment of matrices to group elements should be compatible with the group action. More specifically, the matrix representing a composite group element \mathbf{gh} should equal the matrix product of the representation of \mathbf{g} and \mathbf{h} :

A n -dimensional real *representation* of a group \mathfrak{G} is a map $\rho : \mathfrak{G} \rightarrow \mathbb{R}^{n \times n}$, assigning to each $\mathbf{g} \in \mathfrak{G}$ an *invertible* matrix $\rho(\mathbf{g})$, and satisfying the condition $\rho(\mathbf{gh}) = \rho(\mathbf{g})\rho(\mathbf{h})$ for all $\mathbf{g}, \mathbf{h} \in \mathfrak{G}$. A representation is called *unitary* or *orthogonal* if the matrix $\rho(\mathbf{g})$ is unitary or orthogonal for all $\mathbf{g} \in \mathfrak{G}$.

When Ω is infinite, the space of signals $\mathcal{X}(\Omega)$ is infinite dimensional, in which case $\rho(\mathbf{g})$ is a linear operator on this space, rather than a finite dimensional matrix. In practice, one must always discretise to a finite grid, though.

Similarly, a complex representation is a map $\rho : \mathfrak{G} \rightarrow \mathbb{C}^{n \times n}$ satisfying the same equation.

Written in the language of group representations, the action of \mathfrak{G} on signals $x \in \mathcal{X}(\Omega)$ is defined as $\rho(\mathbf{g})x(u) = x(\mathbf{g}^{-1}u)$. We again verify that

$$(\rho(\mathbf{g})(\rho(\mathbf{h})x))(u) = (\rho(\mathbf{gh})x)(u).$$

Invariant and Equivariant functions The symmetry of the domain Ω underlying the signals $\mathcal{X}(\Omega)$ imposes structure on the function f defined on such signals. It turns out to be a powerful inductive bias, improving learning efficiency by reducing the space of possible interpolants, $\mathcal{F}(\mathcal{X}(\Omega))$, to those which satisfy the symmetry priors. Two important cases we will be exploring in this text are *invariant* and *equivariant* functions.

In general, f depends both on the signal and the domain, i.e., $\mathcal{F}(\mathcal{X}(\Omega), \Omega)$. We will often omit the latter dependency for brevity.

A function $f : \mathcal{X}(\Omega) \rightarrow \mathcal{Y}$ is \mathfrak{G} -*invariant* if $f(\rho(\mathbf{g})x) = f(x)$ for all $\mathbf{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega)$, i.e., its output is unaffected by the group action on the input.

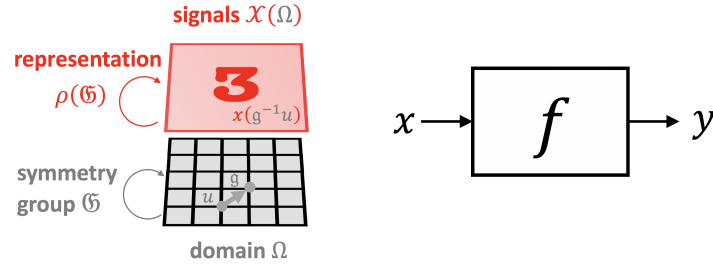


Figure 5: Three spaces of interest in Geometric Deep Learning: the (physical) *domain* Ω , the space of *signals* $\mathcal{X}(\Omega)$, and the *hypothesis* class $\mathcal{F}(\mathcal{X}(\Omega))$. Symmetries of the domain Ω (captured by the group \mathfrak{G}) act on signals $x \in \mathcal{X}(\Omega)$ through group representations $\rho(g)$, imposing structure on the functions $f \in \mathcal{F}(\mathcal{X}(\Omega))$ acting on such signals.

Note that signal processing books routinely use the term ‘shift-invariance’ referring to shift-equivariance, e.g. Linear Shift-invariant Systems.

A classical example of invariance is *shift-invariance*, arising in computer vision and pattern recognition applications such as image classification. The function f in this case (typically implemented as a Convolutional Neural Network) inputs an image and outputs the probability of the image to contain an object from a certain class (e.g. cat or dog). It is often reasonably assumed that the classification result should not be affected by the position of the object in the image, i.e., the function f must be shift-invariant. Multi-layer Perceptrons, which can approximate any smooth function, do not have this property – one of the reasons why early attempts to apply these architectures to problems of pattern recognition in the 1970s failed. The development of neural network architectures with local weight sharing, as epitomised by Convolutional Neural Networks, was, among other reasons, motivated by the need for shift-invariant object classification.

If we however take a closer look at the convolutional layers of CNNs, we will find that they are not shift-invariant but *shift-equivariant*: in other words, a shift of the input to a convolutional layer produces a shift in the output feature maps by the same amount.

More generally, we might have $f : \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$ with input and output spaces having different domains Ω, Ω' and representations ρ, ρ' of the same group \mathfrak{G} . In this case, equivariance is defined as

$$f(\rho(g)x) = \rho'(g)f(x).$$

A function $f : \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega)$ is \mathfrak{G} -equivariant if $f(\rho(g)x) = \rho(g)f(x)$ for all $g \in \mathfrak{G}$, i.e., group action on the input affects the output in the same way.

Resorting again to computer vision, a prototypical application requiring

shift-equivariance is image segmentation, where the output of f is a pixel-wise image mask. Obviously, the segmentation mask must follow shifts in the input image. In this example, the domains of the input and output are the same, but since the input has three color channels while the output has one channel per class, the representations $(\rho, \mathcal{X}(\Omega, \mathcal{C}))$ and $(\rho', \mathcal{X}(\Omega, \mathcal{C}'))$ are somewhat different.

However, even the previous use case of image classification is usually implemented as a sequence of convolutional (shift-equivariant) layers, followed by global pooling (which is shift-invariant). As we will see in Section 3.5, this is a general blueprint of a majority of deep learning architectures, including CNNs and Graph Neural Networks (GNNs).

3.2 Isomorphisms and Automorphisms

Subgroups and Levels of structure As mentioned before, a symmetry is a transformation that preserves some property or structure, and the set of all such transformations for a given structure forms a symmetry group. It happens often that there is not one but multiple structures of interest, and so we can consider several *levels of structure* on our domain Ω . Hence, what counts as a symmetry depends on the structure under consideration, but in all cases a symmetry is an invertible map that respects this structure.

Invertible and structure-preserving maps between different objects often go under the generic name of *isomorphisms* (Greek for ‘equal shape’). An isomorphism from an object to itself is called an *automorphism*, or symmetry.

On the most basic level, the domain Ω is a *set*, which has a minimal amount of structure: all we can say is that the set has some *cardinality*. Self-maps that preserve this structure are *bijections* (invertible maps), which we may consider as set-level symmetries. One can easily verify that this is a group by checking the axioms: a composition of two bijections is also a bijection (closure), the associativity stems from the associativity of the function composition, the map $\tau(u) = u$ is the identity element, and for every τ the inverse exists by definition, satisfying $(\tau \circ \tau^{-1})(u) = (\tau^{-1} \circ \tau)(u) = u$.

For a finite set, the cardinality is the number of elements (‘size’) of the set, and for infinite sets the cardinality indicates different kinds of infinities, such as the countable infinity of the natural numbers, or the uncountable infinity of the continuum \mathbb{R} .

Depending on the application, there may be further levels of structure. For instance, if Ω is a topological space, we can consider maps that preserve *continuity*: such maps are called *homeomorphisms* and in addition to simple bijections between sets, are also continuous and have continuous inverse. Intuitively, continuous functions are well-behaved and map points in a neighbourhood (open set) around a point u to a neighbourhood around $\tau(u)$.

Every differentiable function is continuous. If the map is continuously differentiable ‘sufficiently many times’, it is said to be *smooth*.

One can further demand that the map and its inverse are (continuously) *differentiable*, i.e., the map and its inverse have a derivative at every point (and the derivative is also continuous). This requires further differentiable structure that comes with differentiable manifolds, where such maps are called *diffeomorphisms* and denoted by $\text{Diff}(\Omega)$. Additional examples of structures we will encounter include *distances* or *metrics* (maps preserving them are called *isometries*) or *orientation* (to the best of our knowledge, orientation-preserving maps do not have a common Greek name).

A *metric* or *distance* is a function $d : \Omega \times \Omega \rightarrow [0, \infty)$ satisfying for all $u, v, w \in \Omega$:

Identity of indiscernibles: $d(u, v) = 0$ iff $u = v$.

Symmetry: $d(u, v) = d(v, u)$.

Triangle inequality: $d(u, v) \leq d(u, w) + d(w, v)$.

A space equipped with a metric (Ω, d) is called a *metric space*.

The right level of structure to consider depends on the problem. For example, when segmenting histopathology slide images, we may wish to consider flipped versions of an image as equivalent (as the sample can be flipped when put under the microscope), but if we are trying to classify road signs, we would only want to consider orientation-preserving transformations as symmetries (since reflections could change the meaning of the sign).

As we add levels of structure to be preserved, the symmetry group will get smaller. Indeed, adding structure is equivalent to selecting a *subgroup*, which is a subset of the larger group that satisfies the axioms of a group by itself:

Let (\mathcal{G}, \circ) be a group and $\mathfrak{H} \subseteq \mathcal{G}$ a subset. \mathfrak{H} is said to be a *subgroup* of \mathcal{G} if (\mathfrak{H}, \circ) constitutes a group with the same operation.

For instance, the group of Euclidean isometries $E(2)$ is a subgroup of the group of planar diffeomorphisms $\text{Diff}(2)$, and in turn the group of orientation-preserving isometries $SE(2)$ is a subgroup of $E(2)$. This hierarchy of structure follows the Erlangen Programme philosophy outlined in the Preface: in Klein’s construction, the Projective, Affine, and Euclidean geometries

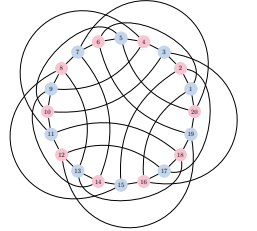
have increasingly more invariants and correspond to progressively smaller groups.

Isomorphisms and Automorphisms We have described symmetries as structure preserving and invertible maps *from an object to itself*. Such maps are also known as *automorphisms*, and describe a way in which an object is equivalent to itself. However, an equally important class of maps are the so-called *isomorphisms*, which exhibit an equivalence between two non-identical objects. These concepts are often conflated, but distinguishing them is necessary to create clarity for our following discussion.

To understand the difference, consider a set $\Omega = \{0, 1, 2\}$. An automorphism of the set Ω is a bijection $\tau : \Omega \rightarrow \Omega$ such as a cyclic shift $\tau(u) = u + 1 \pmod 3$. Such a map preserves the cardinality property, and maps Ω onto itself. If we have another set $\Omega' = \{a, b, c\}$ with the same number of elements, then a bijection $\eta : \Omega \rightarrow \Omega'$ such as $\eta(0) = a, \eta(1) = b, \eta(2) = c$ is a *set isomorphism*.

As we will see in Section 4.1 for graphs, the notion of structure includes not just the number of nodes, but also the connectivity. An isomorphism $\eta : \mathcal{V} \rightarrow \mathcal{V}'$ between two graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is thus a bijection between the nodes that maps pairs of connected nodes to pairs of connected nodes, and likewise for pairs of non-connected nodes. Two isomorphic graphs are thus structurally identical, and differ only in the way their nodes are ordered. On the other hand, a graph automorphism or symmetry is a map $\tau : \mathcal{V} \rightarrow \mathcal{V}$ maps the nodes of the graph back to itself, while preserving the connectivity. A graph with a non-trivial automorphism (i.e., $\tau \neq \text{id}$) presents symmetries.

I.e., $(\eta(u), \eta(v)) \in \mathcal{V}'$ iff $(u, v) \in \mathcal{V}$.

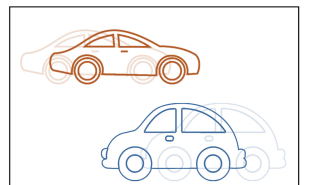


3.3 Deformation Stability

The symmetry formalism introduced in Sections 3.1–3.2 captures an idealised world where we know exactly which transformations are to be considered as symmetries, and we want to respect these symmetries *exactly*. For instance in computer vision, we might assume that planar translations are exact symmetries. However, the real world is noisy and this model falls short in two ways.

Firstly, while these simple groups provide a way to understand *global* sym-

The Folkman graph (Folkman, 1967) is a beautiful example of a graph with 3840 automorphisms, exemplified by the many symmetric ways to draw it.



Two objects moving at different velocities in a video define a transformation group outside the translation group.

metries of the domain Ω (and by extension, of signals on it, $\mathcal{X}(\Omega)$), they do not capture *local* symmetries well. For instance, consider a video scene with several objects, each moving along its own different direction. At subsequent frames, the resulting scene will contain approximately the same semantic information, yet no global translation explains the transformation from one frame to another. In other cases, such as a deformable 3D object viewed by a camera, it is simply very hard to describe the group of transformations that preserve the object identity. These examples illustrate that in reality we are more interested in a far larger set of transformations where global, exact invariance is replaced by a local, inexact one. In our discussion, we will distinguish between two scenarios: the setting where the domain Ω is fixed, and signals $x \in \mathcal{X}(\Omega)$ are undergoing deformations, and the setting where the domain Ω itself may be deformed.

Stability to signal deformations In many applications, we know a priori that a small deformation of the signal x should not change the output of $f(x)$, so it is tempting to consider such deformations as symmetries. For instance, we could view small diffeomorphisms $\tau \in \text{Diff}(\Omega)$, or even small bijections, as symmetries. However, small deformations can be composed to form large deformations, so “small deformations” do not form a group, and we cannot ask for invariance or equivariance to small deformations only. Since large deformations can actually materially change the semantic content of the input, it is not a good idea to use the full group $\text{Diff}(\Omega)$ as symmetry group either.

E.g., the composition of two ϵ -isometries is a 2ϵ -isometry, violating the closure property.

A better approach is to quantify how “far” a given $\tau \in \text{Diff}(\Omega)$ is from a given symmetry subgroup $\mathfrak{G} \subset \text{Diff}(\Omega)$ (e.g. translations) with a complexity measure $c(\tau)$, so that $c(\tau) = 0$ whenever $\tau \in \mathfrak{G}$. We can now replace our previous definition of exact invariance and equivariance under group actions with a ‘softer’ notion of *deformation stability* (or *approximate invariance*):

$$\|f(\rho(\tau)x) - f(x)\| \leq Cc(\tau)\|x\|, \quad , \quad \forall x \in \mathcal{X}(\Omega) \quad (4)$$

where $\rho(\tau)x(u) = x(\tau^{-1}u)$ as before, and where C is some constant independent of the signal x . A function $f \in \mathcal{F}(\mathcal{X}(\Omega))$ satisfying the above equation is said to be *geometrically stable*. We will see examples of such functions in the next Section 3.4.

Since $c(\tau) = 0$ for $\tau \in \mathfrak{G}$, this definition generalises the \mathfrak{G} -invariance property defined above. Its utility in applications depends on introducing an

appropriate deformation cost. In the case of images defined over a continuous Euclidean plane, a popular choice is $c^2(\tau) := \int_{\Omega} \|\nabla \tau(u)\|^2 du$, which measures the ‘elasticity’ of τ , i.e., how different it is from the displacement by a constant vector field. This deformation cost is in fact a norm often called the *Dirichlet energy*, and can be used to quantify how far τ is from the translation group.

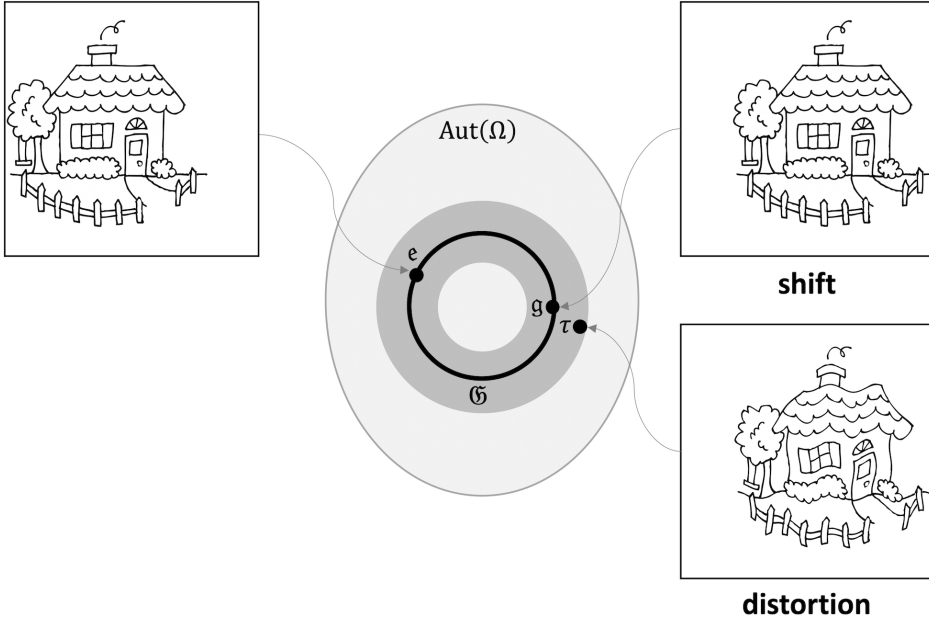


Figure 6: The set of all bijective mappings from Ω into itself forms the *set automorphism group* $\text{Aut}(\Omega)$, of which a symmetry group \mathcal{G} (shown as a circle) is a subgroup. Geometric Stability extends the notion of \mathcal{G} -invariance and equivariance to ‘transformations around \mathcal{G} ’ (shown as gray ring), quantified in the sense of some metric between transformations. In this example, a smooth distortion of the image is close to a shift.

Stability to domain deformations In many applications, the object being deformed is not the signal, but the geometric domain Ω itself. Canonical instances of this are applications dealing with graphs and manifolds: a graph can model a social network at different instance of time containing slightly different social relations (follow graph), or a manifold can model a 3D object undergoing non-rigid deformations. This deformation can be quantified

as follows. If \mathcal{D} denotes the space of all possible variable domains (such as the space of all graphs, or the space of Riemannian manifolds), one can define for $\Omega, \tilde{\Omega} \in \mathcal{D}$ an appropriate metric ('distance') $d(\Omega, \tilde{\Omega})$ satisfying $d(\Omega, \tilde{\Omega}) = 0$ if Ω and $\tilde{\Omega}$ are equivalent in some sense: for example, the graph edit distance vanishes when the graphs are isomorphic, and the Gromov-Hausdorff distance between Riemannian manifolds equipped with geodesic distances vanishes when two manifolds are isometric.

The graph edit distance measures the minimal cost of making two graphs isomorphic by a sequences of graph edit operations. The Gromov-Hausdorff distance measures the smallest possible metric distortion of a correspondence between two metric spaces, see [Gromov \(1981\)](#).

A common construction of such distances between domains relies on some family of invertible mapping $\eta : \Omega \rightarrow \tilde{\Omega}$ that try to 'align' the domains in a way that the corresponding structures are best preserved. For example, in the case of graphs or Riemannian manifolds (regarded as metric spaces with the geodesic distance), this alignment can compare pair-wise adjacency or distance structures (d and \tilde{d} , respectively),

$$d_{\mathcal{D}}(\Omega, \tilde{\Omega}) = \inf_{\eta \in \mathfrak{G}} \|d - \tilde{d} \circ (\eta \times \eta)\|$$

where \mathfrak{G} is the group of isomorphisms such as bijections or isometries, and the norm is defined over the product space $\Omega \times \Omega$. In other words, a distance between elements of $\Omega, \tilde{\Omega}$ is 'lifted' to a distance between the domains themselves, by accounting for all the possible alignments that preserve the internal structure. Given a signal $x \in \mathcal{X}(\Omega)$ and a deformed domain $\tilde{\Omega}$, one can then consider the deformed signal $\tilde{x} = x \circ \eta^{-1} \in \mathcal{X}(\tilde{\Omega})$.

Two graphs can be aligned by the Quadratic Assignment Problem (QAP), which considers in its simplest form two graphs G, \tilde{G} of the same size n , and solves $\min_{\mathbf{P} \in \Sigma_n} \text{trace}(\mathbf{A}\mathbf{P}\tilde{\mathbf{A}}\mathbf{P}^T)$, where $\mathbf{A}, \tilde{\mathbf{A}}$ are the respective adjacency matrices and Σ_n is the group of $n \times n$ permutation matrices. The graph edit distance can be associated with such QAP ([Bougleux et al., 2015](#)).

By slightly abusing the notation, we define $\mathcal{X}(\mathcal{D}) = \{(\mathcal{X}(\Omega), \Omega) : \Omega \in \mathcal{D}\}$ as the ensemble of possible input signals defined over a varying domain. A function $f : \mathcal{X}(\mathcal{D}) \rightarrow \mathcal{Y}$ is stable to domain deformations if

$$\|f(x, \Omega) - f(\tilde{x}, \tilde{\Omega})\| \leq C\|x\|d_{\mathcal{D}}(\Omega, \tilde{\Omega}) \quad (5)$$

for all $\Omega, \tilde{\Omega} \in \mathcal{D}$, and $x \in \mathcal{X}(\Omega)$. We will discuss this notion of stability in the context of manifolds in Sections 4.4–4.6, where isometric deformations play a crucial role. Furthermore, it can be shown that the stability to domain deformations is a natural generalisation of the stability to signal deformations, by viewing the latter in terms of deformations of the volume form [Gama et al. \(2019\)](#).

3.4 Scale Separation

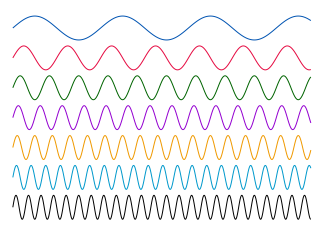
While deformation stability substantially strengthens the global symmetry priors, it is not sufficient in itself to overcome the curse of dimensionality, in

the sense that, informally speaking, there are still “too many” functions that respect (4) as the size of the domain grows. A key insight to overcome this curse is to exploit the multiscale structure of physical tasks. Before describing multiscale representations, we need to introduce the main elements of Fourier transforms, which rely on frequency rather than scale.

Fourier Transform and Global invariants Arguably the most famous signal decomposition is the *Fourier transform*, the cornerstone of harmonic analysis. The classical one-dimensional Fourier transform

$$\hat{x}(\xi) = \int_{-\infty}^{+\infty} x(u) e^{-i\xi u} du$$

expresses the function $x(u) \in L^2(\Omega)$ on the domain $\Omega = \mathbb{R}$ as a linear combination of orthogonal oscillating *basis functions* $\varphi_\xi(u) = e^{i\xi u}$, indexed by their rate of oscillation (or *frequency*) ξ . Such an organisation into frequencies reveals important information about the signal, e.g. its smoothness and localisation. The Fourier basis itself has a deep geometric foundation and can be interpreted as the natural vibrations of the domain, related to its geometric structure (see e.g. [Berger \(2012\)](#)).



Fourier basis functions have global support. As a result, local signals produce energy across all frequencies.

The Fourier transform plays a crucial role in signal processing as it offers a dual formulation of *convolution*,

$$(x \star \theta)(u) = \int_{-\infty}^{+\infty} x(v) \theta(u - v) dv$$

a standard model of linear signal filtering (here and in the following, x denotes the signal and θ the filter). As we will show in the following, the convolution operator is diagonalised in the Fourier basis, making it possible to express convolution as the product of the respective Fourier transforms,

$$\widehat{(x \star \theta)}(\xi) = \hat{x}(\xi) \cdot \hat{\theta}(\xi),$$

a fact known in signal processing as the Convolution Theorem.

As it turns out, many fundamental differential operators such as the Laplacian are described as convolutions on Euclidean domains. Since such differential operators can be defined intrinsically over very general geometries, this provides a formal procedure to extend Fourier transforms beyond Euclidean domains, including graphs, groups and manifolds. We will discuss this in detail in Section 4.4.

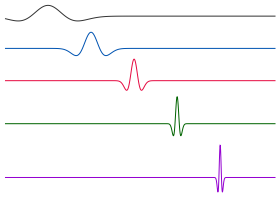
In the following, we will use convolution and (cross-)correlation

$$(x \star \theta)(u) = \int_{-\infty}^{+\infty} x(v) \theta(u+v) dv$$

interchangeably, as it is common in machine learning: the difference between the two is whether the filter is reflected, and since the filter is typically learnable, the distinction is purely notational.

An essential aspect of Fourier transforms is that they reveal *global* properties of the signal and the domain, such as smoothness or conductance. Such global behavior is convenient in presence of global symmetries of the domain such as translation, but not to study more general diffeomorphisms. This requires a representation that trades off spatial and frequential localisation, as we see next.

See [Mallat \(1999\)](#) for a comprehensive introduction.



Contrary to Fourier, wavelet atoms are localised and multi-scale, allowing to capture fine details of the signal with atoms having small spatial support and coarse details with atoms having large spatial support.

The term *atom* here is synonymous with ‘basis element’ in Fourier analysis, with the caveat that wavelets are redundant (over-complete).

Multiscale representations The notion of local invariance can be articulated by switching from a Fourier frequency-based representation to a *scale-based* representation, the cornerstone of multi-scale decomposition methods such as *wavelets*. The essential insight of multi-scale methods is to decompose functions defined over the domain Ω into elementary functions that are localised *both in space and frequency*. In the case of wavelets, this is achieved by correlating a translated and dilated filter (*mother wavelet*) ψ , producing a combined spatio-frequency representation called a *continuous wavelet transform*

$$(W_{\psi}x)(u, \xi) = \xi^{-1/2} \int_{-\infty}^{+\infty} \psi\left(\frac{v-u}{\xi}\right) x(v) dv.$$

The translated and dilated filters are called *wavelet atoms*; their spatial position and dilation correspond to the coordinates u and ξ of the wavelet transform. These coordinates are usually sampled dyadically ($\xi = 2^{-j}$ and $u = 2^{-j}k$), with j referred to as *scale*. Multi-scale signal representations bring important benefits in terms of capturing regularity properties beyond global smoothness, such as piece-wise smoothness, which made them a popular tool in signal and image processing and numerical analysis in the 90s.

Deformation stability of Multiscale representations: The benefit of multiscale localised wavelet decompositions over Fourier decompositions is revealed when considering the effect of small deformations ‘nearby’ the underlying symmetry group. Let us illustrate this important concept in the Euclidean domain and the translation group. Since the Fourier representation diagonalises the shift operator (which can be thought of as convolution, as we will see in more detail in Section 4.2), it is an efficient representation for translation transformations. However, Fourier decompositions are unstable under high-frequency deformations. In contrast, wavelet decompositions offer a stable representation in such cases.

Indeed, let us consider $\tau \in \text{Aut}(\Omega)$ and its associated linear representation $\rho(\tau)$. When $\tau(u) = u - v$ is a shift, as we will verify in Section 4.2, the operator $\rho(\tau) = S_v$ is a *shift operator* that commutes with convolution. Since convolution operators are diagonalised by the Fourier transform, the action of shift in the frequency domain amounts to shifting the complex phase of the Fourier transform,

$$(\widehat{S_v x})(\xi) = e^{-i\xi v} \hat{x}(\xi).$$

Thus, the *Fourier modulus* $f(x) = |\hat{x}|$ removing the complex phase is a simple shift-invariant function, $f(S_v x) = f(x)$. However, if we have only approximate translation, $\tau(u) = u - \tilde{\tau}(u)$ with $\|\nabla \tau\|_\infty = \sup_{u \in \Omega} \|\nabla \tilde{\tau}(u)\| \leq \epsilon$, the situation is entirely different: it is possible to show that

$$\frac{\|f(\rho(\tau)x) - f(x)\|}{\|x\|} = \mathcal{O}(1)$$

irrespective of how small ϵ is (i.e., how close is τ to being a shift). Consequently, such Fourier representation is *unstable under deformations*, however small. This instability is manifested in general domains and non-rigid transformations; we will see another instance of this instability in the analysis of 3d shapes using the natural extension of Fourier transforms described in Section 4.4.

Wavelets offer a remedy to this problem that also reveals the power of multi-scale representations. In the above example, we can show (Mallat, 2012) that the wavelet decomposition $W_\psi x$ is *approximately equivariant* to deformations,

$$\frac{\|\rho(\tau)(W_\psi x) - W_\psi(\rho(\tau)x)\|}{\|x\|} = \mathcal{O}(\epsilon).$$

This notation implies that $\rho(\tau)$ acts on the spatial coordinate of $(W_\psi x)(u, \xi)$.

In other words, decomposing the signal information into scales using localised filters rather than frequencies turns a global unstable representation into a family of locally stable features. Importantly, such measurements at different scales are not yet invariant, and need to be progressively processed towards the low frequencies, hinting at the deep compositional nature of modern neural networks, and captured in our Blueprint for Geometric Deep Learning, presented next.

Scale Separation Prior: We can build from this insight by considering a multiscale coarsening of the data domain Ω into a hierarchy $\Omega_1, \dots, \Omega_J$. As it turns out, such coarsening can be defined on very general domains,

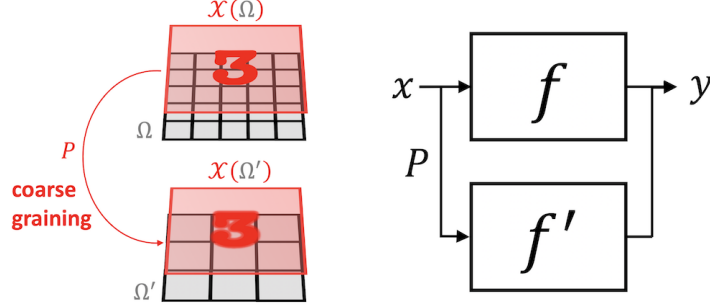


Figure 7: Illustration of Scale Separation for image classification tasks. The classifier f' defined on signals on the coarse grid $\mathcal{X}(\Omega')$ should satisfy $f \approx f' \circ P$, where $P : \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$.

including grids, graphs, and manifolds. Informally, a coarsening assimilates nearby points $u, u' \in \Omega$ together, and thus only requires an appropriate notion of *metric* in the domain. If $\mathcal{X}_j(\Omega_j, \mathcal{C}_j) := \{x_j : \Omega_j \rightarrow \mathcal{C}_j\}$ denotes signals defined over the coarsened domain Ω_j , we informally say that a function $f : \mathcal{X}(\Omega) \rightarrow \mathcal{Y}$ is *locally stable* at scale j if it admits a factorisation of the form $f \approx f_j \circ P_j$, where $P_j : \mathcal{X}(\Omega) \rightarrow \mathcal{X}_j(\Omega_j)$ is a non-linear *coarse graining* and $f_j : \mathcal{X}_j(\Omega_j) \rightarrow \mathcal{Y}$. In other words, while the target function f might depend on complex long-range interactions between features over the whole domain, in locally-stable functions it is possible to *separate* the interactions across scales, by first focusing on localised interactions that are then propagated towards the coarse scales.

Fast Multipole Method (FMM) is a numerical technique originally developed to speed up the calculation of long-ranged forces in n -body problems. FMM groups sources that lie close together and treats them as a single source.

Such principles are of fundamental importance in many areas of physics and mathematics, as manifested for instance in statistical physics in the so-called renormalisation group, or leveraged in important numerical algorithms such as the Fast Multipole Method. In machine learning, multiscale representations and local invariance are the fundamental mathematical principles underpinning the efficiency of Convolutional Neural Networks and Graph Neural Networks and are typically implemented in the form of *local pooling*. In future work, we will further develop tools from computational harmonic analysis that unify these principles across our geometric domains and will shed light onto the statistical learning benefits of scale separation.

3.5 The Blueprint of Geometric Deep Learning

The geometric principles of Symmetry, Geometric Stability, and Scale Separation discussed in Sections 3.1–3.4 can be combined to provide a universal blueprint for learning stable representations of high-dimensional data. These representations will be produced by functions f operating on signals $\mathcal{X}(\Omega, \mathcal{C})$ defined on the domain Ω , which is endowed with a symmetry group \mathfrak{G} .

The geometric priors we have described so far do not prescribe a specific *architecture* for building such representation, but rather a series of necessary conditions. However, they hint at an axiomatic construction that provably satisfies these geometric priors, while ensuring a highly expressive representation that can approximate any target function satisfying such priors.

A simple initial observation is that, in order to obtain a highly expressive representation, we are required to introduce a non-linear element, since if f is linear and \mathfrak{G} -invariant, then for all $x \in \mathcal{X}(\Omega)$,

$$f(x) = \frac{1}{\mu(\mathfrak{G})} \int_{\mathfrak{G}} f(\mathfrak{g}.x) d\mu(\mathfrak{g}) = f\left(\frac{1}{\mu(\mathfrak{G})} \int_{\mathfrak{G}} (\mathfrak{g}.x) d\mu(\mathfrak{g})\right),$$

Here, $\mu(\mathfrak{g})$ is known as the *Haar measure* of the group \mathfrak{G} , and the integral is performed over the entire group.

which indicates that F only depends on x through the \mathfrak{G} -average $Ax = \frac{1}{\mu(\mathfrak{G})} \int_{\mathfrak{G}} (\mathfrak{g}.x) d\mu(\mathfrak{g})$. In the case of images and translation, this would entail using only the average RGB color of the input!

While this reasoning shows that the family of *linear invariants* is not a very rich object, the family of *linear equivariants* provides a much more powerful tool, since it enables the construction of rich and stable features by composition with appropriate non-linear maps, as we will now explain. Indeed, if $B : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega, \mathcal{C}')$ is \mathfrak{G} -equivariant satisfying $B(\mathfrak{g}.x) = \mathfrak{g}.B(x)$ for all $x \in \mathcal{X}$ and $\mathfrak{g} \in \mathfrak{G}$, and $\sigma : \mathcal{C}' \rightarrow \mathcal{C}''$ is an arbitrary (non-linear) map, then we easily verify that the composition $U := (\sigma \circ B) : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega, \mathcal{C}'')$ is also \mathfrak{G} -equivariant, where $\sigma : \mathcal{X}(\Omega, \mathcal{C}') \rightarrow \mathcal{X}(\Omega, \mathcal{C}'')$ is the element-wise instantiation of σ given as $(\sigma(x))(u) := \sigma(x(u))$.

This simple property allows us to define a very general family of \mathfrak{G} -invariants, by composing U with the group averages $A \circ U : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{C}''$. A natural question is thus whether any \mathfrak{G} -invariant function can be approximated at arbitrary precision by such a model, for appropriate choices of B and σ . It is not hard to adapt the standard Universal Approximation Theorems from unstructured vector inputs to show that shallow ‘geometric’ networks are

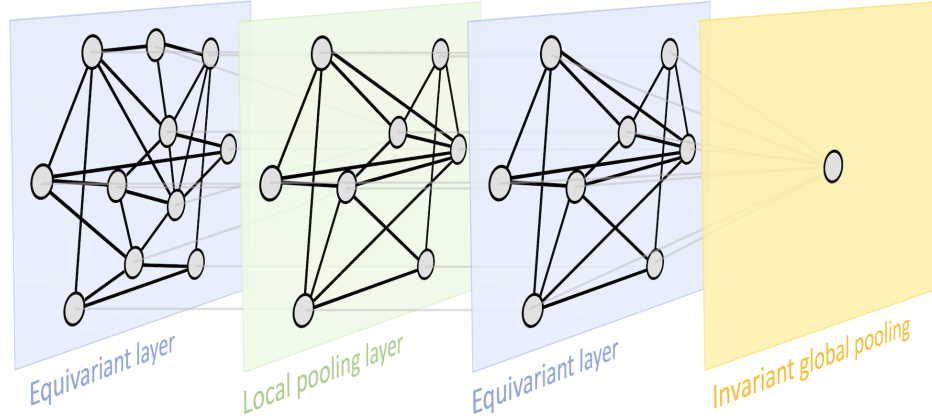


Figure 8: Geometric Deep Learning blueprint, exemplified on a graph. A typical Graph Neural Network architecture may contain permutation equivariant layers (computing node-wise features), local pooling (graph coarsening), and a permutation-invariant global pooling layer (readout layer).

Such proofs have been demonstrated, for example, for the Deep Sets model by [Zaheer et al. \(2017\)](#).

Meaningful metrics can be defined on grids, graphs, manifolds, and groups. A notable exception are sets, where there is no predefined notion of metric.

The term ‘receptive field’ originated in the neuroscience literature, referring to the spatial domain that affects the output of a given neuron.

also universal approximators, by properly generalising the group average to a general non-linear invariant. However, as already described in the case of Fourier versus Wavelet invariants, there is a fundamental tension between shallow global invariance and deformation stability. This motivates an alternative representation, which considers instead *localised* equivariant maps. Assuming that Ω is further equipped with a distance metric d , we call an equivariant map U localised if $(Ux)(u)$ depends only on the values of $x(v)$ for $\mathcal{N}_u = \{v : d(u, v) \leq r\}$, for some small radius r ; the latter set \mathcal{N}_u is called the *receptive field*.

A single layer of local equivariant map U cannot approximate functions with long-range interactions, but a composition of several local equivariant maps $U_J \circ U_{J-1} \cdots \circ U_1$ increases the receptive field while preserving the stability properties of local equivariants. The receptive field is further increased by interleaving downsampling operators that coarsen the domain (again assuming a metric structure), completing the parallel with Multiresolution Analysis (MRA, see e.g. [Mallat \(1999\)](#)).

In summary, the geometry of the input domain, with knowledge of an underlying symmetry group, provides three key building blocks: (i) a local equivariant map, (ii) a global invariant map, and (iii) a coarsening operator.

These building blocks provide a rich function approximation space with prescribed invariance and stability properties by combining them together in a scheme we refer to as the *Geometric Deep Learning Blueprint* (Figure 8).

Geometric Deep Learning Blueprint

Let Ω and Ω' be domains, \mathfrak{G} a symmetry group over Ω , and write $\Omega' \subseteq \Omega$ if Ω' can be considered a compact version of Ω .

We define the following building blocks:

Linear \mathfrak{G} -equivariant layer $B : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C}')$ satisfying $B(\mathfrak{g}.x) = \mathfrak{g}.B(x)$ for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

Nonlinearity $\sigma : \mathcal{C} \rightarrow \mathcal{C}'$ applied element-wise as $(\sigma(x))(u) = \sigma(x(u))$.

Local pooling (coarsening) $P : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C})$, such that $\Omega' \subseteq \Omega$.

\mathfrak{G} -invariant layer (global pooling) $A : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{Y}$ satisfying $A(\mathfrak{g}.x) = A(x)$ for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

Using these blocks allows constructing \mathfrak{G} -invariant functions $f : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{Y}$ of the form

$$f = A \circ \sigma_J \circ B_J \circ P_{J-1} \circ \dots \circ P_1 \circ \sigma_1 \circ B_1$$

where the blocks are selected such that the output space of each block matches the input space of the next one. Different blocks may exploit different choices of symmetry groups \mathfrak{G} .

Different settings of Geometric Deep Learning One can make an important distinction between the setting when the domain Ω is assumed to be *fixed* and one is only interested in varying input signals defined on that domain, or the domain is part of the input as *varies* together with signals defined on it. A classical instance of the former case is encountered in computer vision applications, where images are assumed to be defined on a fixed domain