

PDE-based Group Equivariant Convolutional Neural Networks

Bart M.N. Smets* Jim Portegies† Erik J. Bekkers‡ Remco Duits*

January 24, 2020

Abstract

We present a PDE-based framework that generalizes Group equivariant Convolutional Neural Networks (G-CNNs). In this framework, a network layer is seen as a set of PDE-solvers where the equation’s geometrically meaningful coefficients become the layer’s trainable weights. Formulating our PDEs on homogeneous spaces allows these networks to be designed with built-in symmetries such as rotation equivariance instead of being restricted to just translation equivariance as in traditional CNNs. Having all the desired symmetries included in the design obviates the need to include them by means of costly techniques such as data augmentation. Roto-translation equivariance for image analysis applications is the example we will be using throughout the paper.

Our default PDE is solved by a combination of linear group convolutions and non-linear morphological group convolutions. Just like for linear convolution a morphological convolution is specified by a kernel and this kernel is what is being optimized during the training process. We demonstrate how the common CNN operations of max/min-pooling and ReLUs arise naturally from solving a PDE and how they are subsumed by morphological convolutions.

We present a proof-of-concept experiment to demonstrate the potential of this framework in increasing the performance of deep learning based imaging applications.

1 Introduction

1.1 PDE-guided CNN Design on Homogeneous Spaces

In this article we introduce *PDE-guided CNN design on homogeneous spaces*: we interpret layers of convolutional neural networks as solvers of Partial Differential Equations, and show how to use this interpretation in the design of such layers and full neural networks.

More specifically, we will explain how a convolutional neural network layer approximately solves a set of evolutionary PDEs driven by **convection**, **(fractional) diffusion** and **dilation and erosion** as is illustrated in Fig. 1. They correspond to the usual sublayers in a CNN: the convolution layer, a sub or super sampling layer and a max-pooling layer.

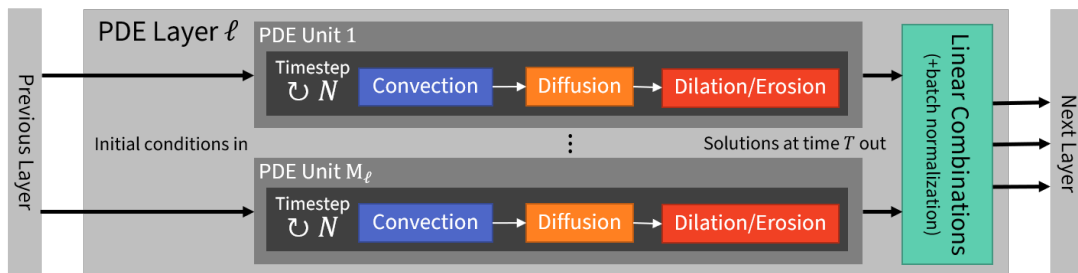


Figure 1: Illustrating the architecture of a PDE Layer. The output from previous layer serves as initial conditions to a set of M_ℓ evolution equations at layer ℓ , the solutions of which at a fixed time T will be combined into new initial conditions for the next layer.

*ADG, Department of Mathematics and Computer Science, Eindhoven University of Technology, email: b.m.n.smets@tue.nl

†CASA, Department of Mathematics and Computer Science, Eindhoven University of Technology

‡Amsterdam Machine Learning Lab, University of Amsterdam

1.2 Drawing Inspiration from PDE-based Image Analysis

Since the Partial Differential Equations that will arise from CNN layers are well-known in the context of geometric image analysis [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11], the layers also get an interpretation in terms of classical image-processing operators. This allows intuition and techniques from geometric PDE-based image analysis to be carried over to neural networks.

In geometric PDE-based image processing it can be beneficial to include mean curvature or other geometric flows [12, 13, 14, 15] as regularization and our framework provides a natural way for such flows to be included into neural networks. In the PDE-layer from Fig. 1 we only mention diffusion as a means of regularization, but mean curvature flow could easily be integrated by replacing the diffusion sub-layer with a mean curvature flow sub-layer. This would require replacing the linear convolution for diffusion by a median filtering approximation of mean curvature flow.

1.3 The Need for Lifting Images

In geometric image analysis it is often useful to *lift* images from a 2D picture to a 3D orientation score as in Fig. 2 and do further processing on the orientation scores [16, 17]. A typical image processing task in which such a lift is beneficial is that of the segmentation of blood vessels in a medical image. Algorithms based on processing the 2D picture directly, usually fail around points where two blood vessels cross, but algorithms that lift the image to an orientation score manage to decouple the blood vessels with different orientations as is illustrated in the bottom row of Fig. 2.

To be able to endow image-processing neural networks with the added capabilities (such as decoupling orientations and guaranteeing equivariance) that result from lifting data to an extended domain, we develop our theory for the more general CNNs defined on *homogeneous spaces*, rather than just the prevalent CNNs defined on Euclidean space. One can then choose which homogeneous space to use based on the needs of one's application (such as needing to decouple orientations). A homogeneous space is, given subgroup H of a group G , the manifold of left cosets, denoted by G/H . In the above image-analysis example, the group G would be the special Euclidean group $G = SE(d)$, the subgroup H would be the stabilizer subgroup of a fixed reference axis, and the corresponding homogeneous space G/H would be the space of positions and orientations \mathbb{M}_d , which is the lowest dimensional homogeneous space able to decouple orientations. By considering convolutional neural networks on homogeneous spaces such as \mathbb{M}_d these networks have access to the same benefits of decoupling structures with different orientations as was highly beneficial for geometric image processing [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32].

1.4 The Need for Equivariance

We require the layers to be *equivariant*: a transformation of the input should lead to a corresponding transformation of the output, in such a way that first transforming the input and then applying the network or first applying the network and then transforming the output yield the same result. A particular example, in which the output transformation is trivial (i.e. the identity transformation), is that of *invariance*: in many classification tasks, such as the recognition of objects in pictures, an apple should still be recognized as an apple even if it is shifted or otherwise transformed in the picture as illustrated in Fig. 3. By guaranteeing equivariance of the network, the amount of data necessary or the need for (geometric) data augmentation are reduced as the required symmetries are intrinsic to the network and need not be trained.

1.5 Overall Architecture

We call the neural networks that follow from our design *PDE-G-CNNs* as they put equivariant G-CNNs [33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45] in a broader, PDE-based context.

A key aspect in our design is that we include equivariant morphological convolutions [46] on the homogeneous space G/H in our PDE-G-CNN. Normally, morphological convolutions are considered on \mathbb{R}^d [47, 48], but when extended to Lie groups such as $SE(d)$ they have many benefits in applications (e.g. crossing-preserving flow [49] or tracking [50, 51]), this requires an overall network architecture as shown in Fig. 4.

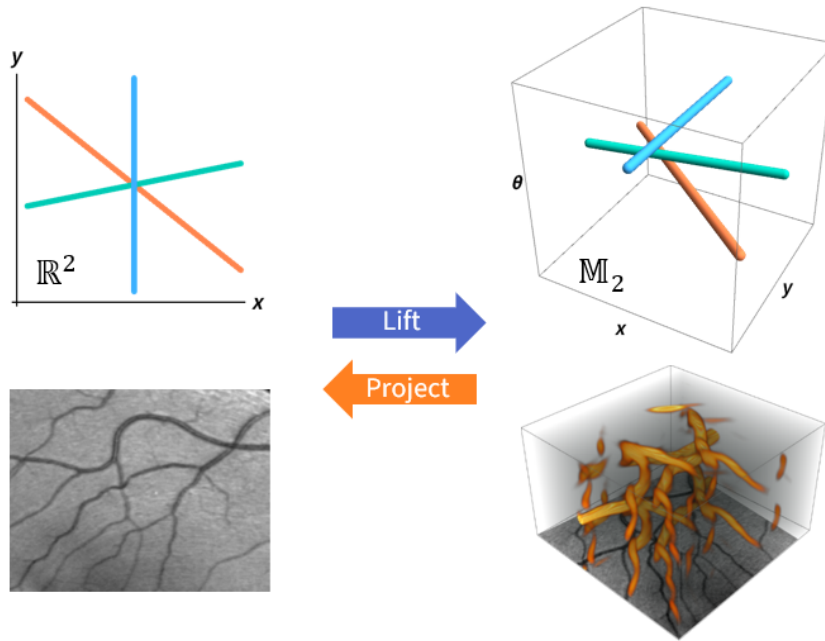


Figure 2: Illustrating the process of lifting and projecting, in this case the advantage of lifting an image from \mathbb{R}^2 to the 2D space of positions and orientations \mathbb{M}_2 derives from the disentanglement of the lines at the crossing.

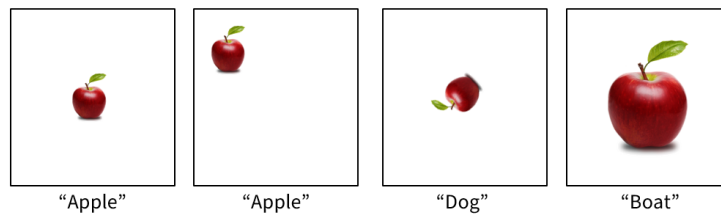


Figure 3: Spatial CNNs, as used for image classification for example, are translation equivariant but not necessarily equivariant with respect to rotation, scaling and other transformations as the tags of the differently transformed apples images suggest. Building a G-CNN with the appropriately chosen group confers the network with all the equivariances appropriate for the chosen application.

By treating the layers of CNNs as solvers of PDEs, we gain *geometric interpretation* of the layers and the involved training parameters. Moreover, by making choices regarding which parameter to train and which not, one can interpolate between a strongly guided design with many geometric priors, towards a much more expressive network with many more PDE parameters being determined by a training algorithm.

Remark 1.1 (Generality of the architecture). Although not considered here, for other Lie groups applications (e.g. frequency scores[52], velocity scores, scale-orientation scores[53]) the same structure applies, therefore we keep our theory in the general setting of homogeneous spaces G/H .

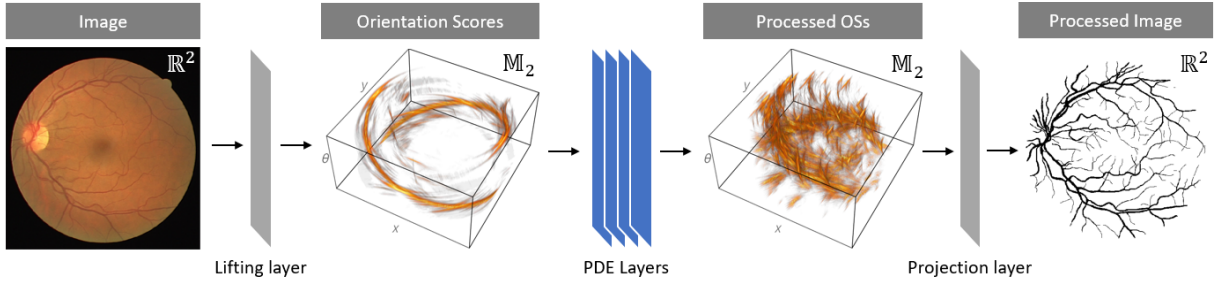


Figure 4: Illustrating the overall architecture of a PDE-G-CNN (example: retinal vessel segmentation). An input image is lifted to a homogeneous space from which point on it can be fed through subsequent PDE layers that replace the convolution layers in conventional CNNs. Finally the result is projected back to the desired output space.

1.6 Related Work

G-CNNs After the introduction of G-CNNs by Cohen & Welling [33] in the field of machine and deep learning, G-CNNs became popular. This resulted in many articles on showing the benefits of G-CNNs over classical spatial CNNs. These works can be roughly categorised as

- discrete G-CNNs [33, 34, 35, 36, 37],
- regular continuous G-CNNs [50, 39, 40, 41, 54],
- and steerable continuous G-CNNs [42, 43, 44, 45, 55] that rely on Fourier transforms on homogeneous spaces [56, 31].

Both regular and steerable G-CNNs naturally arise from linear mappings between functions on homogeneous spaces that are placed under equivariance constraints [42, 44, 55, 54]. Regular G-CNNs explicitly extend the domain and lift feature maps to a larger homogeneous space of a group, whereas steerable CNNs extend the co-domain by generating fiber bundles in which a steerable feature vector is assigned to each position in the base domain. In this work we adopt the domain extension viewpoint. Although steerable operators have clear benefits in terms of computational efficiency and accuracy [57, 58], working with steerable representations puts constraints on non-linear activations within the networks which limits representation power of G-CNNs [55]. Like regular G-CNNs, the proposed PDE-G-CNNs do not suffer from this. In our proposed PDE-G-CNN framework it is moreover essential that we adapt the domain-extension viewpoint, as this allows to naturally and transparently construct PDEs via left-invariant vector fields.

Neural Networks and Differential Equations The connection between neural networks and differential equations became widely known in 2017, when Weinan E [59] made explicit the connection between neural networks and dynamical systems especially in the context of the ultradeep ResNet [60]. This point of view was further expanded by Lu et al. [61], showing how many ultradeep neural networks can be viewed as discretizations of ordinary differential equations. The somewhat opposite point of view was taken by Chen et al. [62], who introduced a new type of neural network which no longer has discrete layers, them being replaced by a field parameterized by a continuous time variable.

Weinan E also indicated a relationship between CNNs and PDEs, or rather with evolution equations involving a nonlocal operator.

Implicitly, the connection between neural networks and differential equations was known before, such as in the work by Chen et al. [63] who learn parameters in a reaction-diffusion equation. This connection between neural networks and PDEs was then made explicit and more extensive by Long et al. who made it possible to learn a much wider class of PDEs [64] with their PDE-Net. This contrasts with our equivariant PDE approach on homogeneous spaces which are solvable by linear and (as we will see) morphological convolutions.

A particular useful aspect of the connection between neural networks and differential equations is the possible insight that the stability of the differential equation can give geometric insight into the generalization ability of the neural networks [65].

1.7 Structure of the Article

The structure of the article is as follows. We first introduce the needed theoretical preliminaries from Lie group theory in Section 2 where we also define the space of positions and orientations \mathbb{M}_d that will allow us to construct roto-translation equivariant networks. We assume that the reader has basic background in differential geometry.

In Section 3 we give the overall architecture of a PDE-G-CNN and the ancillary tools that are needed to support the PDE layers that form the core of a PDE-G-CNN. We detail the design of these PDE layers in Section 4 and propose an equivariant PDE that models commonly used operations in CNNs.

In Section 5 we detail how our PDE of interest can be solved using a process called operator splitting and how the splitted sub-operators relate to operations in conventional CNNs. Additionally, we give tangible approximations to the fundamental solutions of the PDEs that are both easy to compute and sufficiently accurate for practical applications. The derivation of the approximate fundamental solution to the morphological part of our PDE is considered in Section 6 as it is a more technical process.

We end our paper with a proof-of-concept experiment in Section 7 and concluding remarks in Section 8.

The framework we propose covers transformations and CNNs on homogeneous spaces in general and as such we develop the theory in an abstract fashion. To maintain a bridge with practical applications we give details throughout the article on what form the abstractions take in the case of roto-translation equivariant networks, specifically in 2D.

2 Equivariance: Groups & Homogeneous Spaces

We want to design the PDE-G-CNN, and its layers, in such a way that they are *equivariant*. Equivariance essentially means that one can either transform the input and then feed it through the network, or first feed it through the network and then transform the output, and both give the same result. We will give a precise definition after introducing general notation.

2.1 The General Case

A layer in a neural network (or indeed the whole network) can be viewed as an operator from a space of functions defined on a space X to a space of functions defined on a space Y . It may be helpful to think of these function spaces as spaces of images.

We assume that the possible transformations form a *Lie group* G . Think for instance of a group of translations which shift the domain into different directions. Mathematically, we assume that the Lie group G acts smoothly on both X and Y , which means that there are smooth maps $\rho_X : G \times X \rightarrow X$ and $\rho_Y : G \times Y \rightarrow Y$ such that for all $g, h \in G$,

$$\rho_X(gh, x) = \rho_X(g, \rho_X(h, x)) \quad \text{and} \quad \rho_Y(gh, x) = \rho_Y(g, \rho_Y(h, x)).$$

Instead of $\rho_X(g, x)$ or $\rho_Y(h, y)$ we will just write $g.x$ and $h.y$ respectively and refer to these maps as the group action of G on X respectively Y . In general it is clear from the element the group element is acting on which group action is used and so there is no confusion in using the same symbol for all group actions.

We will assume that the group G acts *transitively* on the spaces, meaning that for any two elements of the space there exists a transformation in G that maps one to the other. This has as the consequence that X and Y can be seen as *homogeneous spaces* [66]. In particular, this means that after selecting a reference element $x_0 \in X$ we can make the following isomorphism:

$$(X, G, x_0) \cong G/\text{Stab}_G(x_0) \tag{1}$$

using the mapping

$$x \mapsto \{g \in G \mid g.x_0 = x\}, \tag{2}$$

which is a bijection due to transitivity and the fact that $\text{Stab}_G(x_0)$ is a subgroup of G . Because of this we will represent a homogeneous space as the quotient G/H for some choice of subgroup H since all homogeneous spaces are isomorphic to such a quotient by the above construction.

The elements of the quotient G/H consist of subsets of G which we denote by the letter p , these subsets are known as left cosets of H since every one of them consists of the set $p = gH$ for some $g \in G$, the left cosets are a partition of G under the equivalence relation

$$g_1 \equiv g_2 \iff g_1^{-1}g_2 \in H.$$

We still use the group action notation to make a clear distinction between the group on the one hand and the quotient that represents the homogeneous space on the other hand by defining

$$g \cdot p := gp, \tag{3}$$

which is again a left coset and so an element of G/H . This is a natural extension of the group action since by the isomorphism (2) we now have

$$x \stackrel{(2)}{\mapsto} p \iff g \cdot x \stackrel{(2)}{\mapsto} g \cdot p \tag{4}$$

as expected.

While by design the p 's are subsets of G we generally use them as atomic entities ("points") that represent some $x \in X$ by $p \cdot x_0 = x$. In case we do want to use them explicitly as subsets of G to access the group elements within we use the more explicit notation

$$G_p := p \subset G. \tag{5}$$

Under this notation the group G consists of the disjoint union

$$G = \coprod_{p \in G/H} G_p. \tag{6}$$

The left coset that is associated with the reference element $x_0 \in X$ is H and for that reason we also alias it by $p_0 := H$ so that the isomorphism (2) maps $x_0 \mapsto p_0$.

Remark 2.1 (Principal homogeneous space). Observe that by choosing $H = \{e\}$ we get $G/H \equiv G$, i.e. the Lie group is a homogeneous space of itself. This is called the principal homogeneous space. In that case the group action is equivalent to the group composition.

The action on a homogeneous space G/H induces an action on spaces of functions on G/H . A neural network layer is itself an operator (from functions on G/H_X to functions on G/H_Y), and we require the function to be equivariant with respect to the actions on these function spaces.

Definition 2.2 (Equivariance). Let G be a Lie group with homogeneous spaces X and Y . Let Φ be an operator from functions (of some function class) on X to functions on Y , then we say that Φ is equivariant with respect to G if for all functions f (of that class) we have that:

$$\forall g \in G, y \in Y : (\Phi f)(g \cdot y) = \left(\Phi [x \mapsto f(g \cdot x)] \right)(y). \tag{7}$$

Indeed, this definition expresses that one can either first apply the transformation g and then apply Φ , or first apply Φ and then apply the transformation g , and the result is the same.

We will denote the group action/left-multiplication by an element $g \in G$ by the operator $L_g : G/H \rightarrow G/H$ given by

$$L_g p := g \cdot p \quad \text{for all } p \in G/H. \tag{8}$$

In addition, we denote the left-regular representation of G on functions f defined on G/H by \mathcal{L}_g defined by

$$(\mathcal{L}_g f)(p) := f(g^{-1} \cdot p). \tag{9}$$

With this notation, condition (7) on our neural network operator Φ can be rewritten as: for all $g \in G$,

$$\boxed{\mathcal{L}_g \circ \Phi = \Phi \circ \mathcal{L}_g.} \tag{10}$$

2.2 Vector and Tensor Fields

The particular operators that we will base our framework on are vector and tensor fields and we explain what left invariance means for these object next.

For $g \in G$ and $p \in G/H$, the pushforward

$$(L_g)_* : T_p(G/H) \rightarrow T_{g.p}(G/H)$$

of the group action L_g is defined by the condition that for all smooth functions f on G/H and all $\mathbf{v} \in T_p(G/H)$ we have that

$$((L_g)_* \mathbf{v}) f := \mathbf{v}(f \circ L_g). \quad (11)$$

Remark 2.3 (Tangent vectors as differential operators). Other then the usual geometric interpretation of tangent vectors as being the velocity vectors $\dot{\gamma}(t)$ tangent to some differentiable curve $\gamma : \mathbb{R} \rightarrow G/H$ we will simultaneously use them as differential operators acting on functions as we did in (11). This algebraic viewpoint defines the action of the tangent vector $\dot{\gamma}(t)$ on a differentiable function f as

$$\dot{\gamma}(t)f := \left. \frac{\partial}{\partial s} f(\gamma(s)) \right|_t.$$

In the flat setting of $G = (\mathbb{R}^d, +)$, where the tangent spaces are isomorphic to the base manifold \mathbb{R}^d , when we have a tangent vector $\mathbf{c} \in \mathbb{R}^d$ its application to a function is the familiar directional derivative:

$$\mathbf{c}f = \mathbf{c} \cdot \nabla f = df(\mathbf{c}).$$

See [67, §2.1.1] for details on this double interpretation.

Vector fields that have the special property that the push forward $(L_g)_*$ maps them to themselves in the sense that

$$\forall g \in G, \forall p \in G/H : \mathbf{v}(p)f = \mathbf{v}(g.p) [L_g f], \quad (12)$$

for all differentiable function f and where $\mathbf{v} : p \mapsto T_p(G/H)$ is a vector field, are referred to as left invariant.

Definition 2.4 (Left-invariant vector field on a homogeneous space). A vector field \mathbf{v} on G/H is left invariant with respect to G if it satisfies

$$\forall g \in G, \forall p \in G/H : \mathbf{v}(g.p) = (L_g)_* \mathbf{v}(p). \quad (13)$$

It is straightforward to check that (12) and (13) are equivalent and that these imply the following.

Corollary 2.5 (Properties of left-invariant vector fields). *On a homogeneous space G/H with reference element p_0 the left-invariant vector fields have the following properties:*

1. they are fully determined by their value $\mathbf{v}(p_0)$ in p_0 ,
2. $\forall h \in H, \forall \mathbf{v} \in T_{p_0}(G/H) : (L_h)_* \mathbf{v} = \mathbf{v}$.

We also introduce left-invariant metric tensor fields.

Definition 2.6 (Left-invariant metric tensor field). Let G be a Lie group and G/H a homogeneous space then the metric tensor field \mathcal{G} on G/H is left-invariant with respect to G if and only if

$$\forall g \in G, \forall p \in G/H, \forall \mathbf{v}, \mathbf{w} \in T_p(G/H) : \mathcal{G}|_p(\mathbf{v}, \mathbf{w}) = \mathcal{G}|_{g.p} \left((L_g)_* \mathbf{v}, (L_g)_* \mathbf{w} \right). \quad (14)$$

Recall that $L_g p := g.p$ and so the push-forward $(L_g)_*$ maps tangent vector from T_p to $T_{g.p}$. Again it follows immediately from this definition that a left-invariant metric has similar properties as a left-invariant vector field.

Corollary 2.7 (Properties of left-invariant metric tensor fields). *On a homogeneous space G/H with reference element p_0 a left-invariant metric tensor field \mathcal{G} has the following properties:*

1. it is fully determined by its metric tensor $\mathcal{G}|_{p_0}$ at p_0 ,
2. $\forall h \in H, \forall \mathbf{v}, \mathbf{w} \in T_{p_0}(G/H) : \mathcal{G}|_{p_0}(\mathbf{v}, \mathbf{w}) = \mathcal{G}|_{p_0}((L_h)_* \mathbf{v}, (L_h)_* \mathbf{w})$.

Or in words, the metric has to be symmetric with respect to the subgroup H .

We end our theoretical preliminaries by introducing the space of positions and orientations \mathbb{M}_d .

2.3 Example: The Group $SE(d)$ and the Homogeneous Space \mathbb{M}_d

Our main example and Lie group of interest is the *Special Euclidean* group $SE(d)$ of the rotations and translations of \mathbb{R}^d , in particular for $d \in \{2, 3\}$. When we take $H = \{0\} \times SO(d-1)$ we obtain the space of positions and orientations $\mathbb{M}_d = SE(d)/(\{0\} \times SO(d-1))$. This homogeneous space and group will enable the construction of roto-translation equivariant networks.

As a set we identify \mathbb{M}_d with $\mathbb{R}^d \times S^{d-1}$ and choose some reference direction $\mathbf{a} \in S^{d-1} \subset \mathbb{R}^d$ as the forward direction so that we can set the reference point of the space as $p_0 = (\mathbf{0}, \mathbf{a})$. We can then see that elements of H are those rotations that map \mathbf{a} to itself, i.e. rotations with the forward direction as their axis.

If we denote elements of $SE(d)$ as translation/rotation pairs $(\mathbf{y}, R) \in \mathbb{R}^d \times SO(d)$ then group multiplication is given by

$$g_1 = (\mathbf{y}_1, R_1), g_2 = (\mathbf{y}_2, R_2) \in G : g_1 g_2 = (\mathbf{y}_1, R_1) (\mathbf{y}_2, R_2) = (\mathbf{y}_1 + R_1 \mathbf{y}_2, R_1 R_2), \quad (15)$$

and the group action on elements $p = (\mathbf{x}, \mathbf{n}) \in \mathbb{R}^d \times S^{d-1} \equiv \mathbb{M}_d$ is given as

$$g.p = (\mathbf{y}, R).(\mathbf{x}, \mathbf{n}) = (\mathbf{y} + R\mathbf{x}, R\mathbf{n}). \quad (16)$$

Proposition 2.8 (Left-invariant Riemannian metric tensors fields on \mathbb{M}_d). *The only Riemannian metric tensor fields on \mathbb{M}_d that are left invariant with respect to $SE(d)$ are of the form:*

$$\mathcal{G}|_{(\mathbf{x}, \mathbf{n})} \left((\dot{\mathbf{x}}, \dot{\mathbf{n}}), (\dot{\mathbf{x}}, \dot{\mathbf{n}}) \right) = D_M |\dot{\mathbf{x}} \cdot \mathbf{n}|^2 + D_L \|\dot{\mathbf{x}} \wedge \mathbf{n}\|^2 + D_A \|\dot{\mathbf{n}}\|^2, \quad (17)$$

with $D_M, D_L, D_A > 0$ weighing the main, lateral and angular motion respectively and where the inner product, outer product and norm are the standard Euclidean constructs.

Proof. It follows that to satisfy (14) at the tangent space $T_{(\mathbf{x}, \mathbf{n})}$ of a particular (\mathbf{x}, \mathbf{n}) the metric tensor needs to be symmetric with respect to rotations about \mathbf{n} both spatially and angularly which leads to the three degrees of freedom contained in (17) irrespective of d . \square

For $d = 2$ we represent elements of \mathbb{M}_2 with $(x, y, \theta) \in \mathbb{R}^3$ where x, y are the usual Cartesian coordinates and θ the angle with respect to the x -axis, the reference element is then simply denoted by $(0, 0, 0)$. The left-invariant metric tensors are then given by

$$\begin{aligned} \mathcal{G}|_{(x, y, \theta)} \left((\dot{x}, \dot{y}, \dot{\theta}), (\dot{x}, \dot{y}, \dot{\theta}) \right) &= D_M (|\dot{x} \cos \theta|^2 + |\dot{y} \sin \theta|^2) \\ &+ D_L (|\dot{x} \sin \theta|^2 + |\dot{y} \cos \theta|^2) + D_A |\dot{\theta}|^2. \end{aligned} \quad (18)$$

While left-invariant metric tensor fields on \mathbb{M}_d are of essentially the same form and with the same three degrees of freedom for all $d \geq 2$ there is a fundamental difference between left-invariant vector fields for $d = 2$ and $d \geq 3$.

Proposition 2.9 (Left-invariant vector fields on \mathbb{M}_d). *On \mathbb{M}_2 the left-invariant vector fields are spanned by the following basis:*

$$\begin{cases} \mathcal{A}_1|_{(x, y, \theta)} &= \cos \theta \partial_x|_{(x, y, \theta)} + \sin \theta \partial_y|_{(x, y, \theta)}, \\ \mathcal{A}_2|_{(x, y, \theta)} &= -\sin \theta \partial_x|_{(x, y, \theta)} + \cos \theta \partial_y|_{(x, y, \theta)}, \\ \mathcal{A}_3|_{(x, y, \theta)} &= \partial_\theta|_{(x, y, \theta)}. \end{cases} \quad (19)$$

For $d \geq 3$ the case is somewhat simplified, let $\partial_{\mathbf{a}} \in T_{p_0}(\mathbb{M}_d)$ be the tangent vector in the reference point in the main direction \mathbf{a} , specifically:

$$\partial_{\mathbf{a}} f := \lim_{t \rightarrow 0} \frac{f((t\mathbf{a}, \mathbf{a})) - f((\mathbf{0}, \mathbf{a}))}{t},$$

then all left-invariant vector fields are spanned by the vector field:

$$p \mapsto \mathcal{A}_1 \Big|_p := \left(L_{g_p} \right)_* \partial_{\mathbf{a}}, \quad (20)$$

where $g_p \in G$ is chosen such that $g_p \cdot p_0 = p$ or equivalently $g_p \in G_p$.

Proof. For $d = 2$ we have $\mathbb{M}_2 \equiv SE(2)$ and the left-invariant vector fields on \mathbb{M}_2 are exactly the left-invariant vector fields on $SE(2)$ given by (19).

For $d \geq 3$ we can see that (20) are the only left-invariant vector fields since for all $h \in H$ we have $(g_p h) \cdot p_0 = p$ and so in order to be well-defined we must require $(L_h)_* \mathbf{v} = \mathbf{v}$ on $T_{p_0}(\mathbb{M}_d)$, and this is true for $\partial_{\mathbf{a}}$ (and its scalar multiples) but not true for any other tangent vectors at $T_{p_0}(\mathbb{M}_d)$. \square

3 Overall Architecture

A key ingredient in of what we call a PDE-G-CNN is the PDE layer that we detail in the next section, however to make a complete network we need more. Specifically we need a layer that transforms the network's input into a format that is suitable for the PDE layers and a layer that takes the output of the PDE layers and transforms it to the desired output format. We call this input and output transformation *lifting* respectively *projection*, this yields the overall architecture of a PDE-G-CNN as illustrated in Fig. 4.

As our theoretical preliminaries suggest we aim to do processing on homogeneous spaces but the input and output of the network do not necessarily live on that homogeneous space. Indeed in the case of images the data lives on \mathbb{R}^2 and not on \mathbb{M}_2 where we propose to do processing.

This necessitates the addition of *lifting* and *projection* layers to first transform the input to the desired homogeneous space and end with transforming it back to the required output space. Of course for the entire network to be equivariant we require these transformation layers to be equivariant as well. In this paper we focus on the design of the PDE layers, details on appropriate equivariant lifting and projection layers in the case of $SE(2)$ can be found in [40, 15].

Remark 3.1 (General equivariant linear transformations between homogeneous spaces). A general way to lift and project from one homogeneous space to another in a trainable fashion is the following. Consider two homogeneous spaces X and Y of a Lie group G with x_0 being the reference element of X , let $f \in L^2(X)$ and $k \in L^1(Y)$ with the following property:

$$\forall h \in \text{Stab}_G(x_0), y \in Y : k(h \cdot y) = k(y)$$

then the operator \mathcal{T} defined by

$$\forall y \in Y : (\mathcal{T}f)(y) := \int_G k(g^{-1} \cdot y) f(g \cdot x_0) d\mu_G(g) \quad (21)$$

transforms f from a function on X to a function on Y in an equivariant manner. Here the kernel k is the trainable part and μ_G is the Haar measure on the group.

Moreover it can be shown via the Dunford-Pettis[68] theorem that (under mild restrictions) all linear transforms between homogeneous spaces are of this form.

Remark 3.2 (Lifting and projecting on \mathbb{M}_2). Lifting an image (function) on \mathbb{R}^2 to \mathbb{M}_2 can either be performed by a non-trainable *Orientation Score Transform* [16] or a trainable lift [40] in the style of Remark 3.1.

Projecting from \mathbb{M}_2 back down to \mathbb{R}^2 can be performed by a simple maximum projection: let $f : \mathbb{M}_2 \rightarrow \mathbb{R}$ then

$$(x, y) \mapsto \max_{\theta \in [0, 2\pi)} f(x, y, \theta) \quad (22)$$

is a roto-translation equivariant projection as used in [40]. A variation on the above projection is detailed in [15, Ch. 3.3.3].

4 Design of a PDE Layer

A PDE layer operates by taking its inputs as the initial conditions for a set of evolution equations, hence there will be a PDE associated with each input feature. The idea is that we let each of these evolution equations work on the inputs up to a fixed time T . Afterwards, we take these solutions at time T and take (batch normalized) linear combinations of them to produce the outputs of the layer and as such the initial conditions for the next set of PDEs.

If we index network layers (i.e. the depth of the network) with ℓ and denote the width (i.e. the number of features or channels) at layer ℓ with M_ℓ then we have M_ℓ PDEs and take $M_{\ell+1}$ linear combinations of their solutions. We divide a PDE layer into PDE units that each solve a PDE and the linear combination unit. This design is illustrated in Fig. 1.

4.1 The PDE Unit

The PDE unit is the part of the network that has a one-to-one correspondence to a particular PDE, it interprets its input as an initial condition (at $t = 0$) for an evolution PDE, and produces as output the (approximate) solution to the PDE at a time $t = T$. Many choices of PDEs are possible, and the choice can vary within the network.

It is essential that we require network layers, and thereby all PDE units, to be *equivariant*. This has consequences for the class of PDEs that is allowed.

The default PDE unit that we will consider in this article computes the approximate solution to the PDE

$$\begin{cases} \frac{\partial W}{\partial t}(p, t) = \underbrace{-cW(p, t)}_{\text{Convection}} - \underbrace{\left(-\Delta_{\mathcal{G}_1}\right)^\alpha W(p, t)}_{\text{Fractional diffusion}} \pm \underbrace{\left\|\nabla_{\mathcal{G}_2} W(p, t)\right\|_{\mathcal{G}_2}^{2\alpha}}_{\text{Dilation/Erosion}} & \text{for } p \in G/H, t \geq 0, \\ W(p, 0) = U(p) & \text{for } p \in G/H. \end{cases} \quad (23)$$

Here, c is a left-invariant vector field on G/H (recall (20) and our use of tangent vectors as differential operators per Remark 2.3), $\alpha \in [1/2, 1]$, \mathcal{G}_1 and \mathcal{G}_2 are left-invariant metric tensor fields on G/H for $i \in \{1, 2\}$, U is the initial condition and $\Delta_{\mathcal{G}}$ and $\|\cdot\|_{\mathcal{G}}$ denote the Laplace-Beltrami operator and norm induced by the metric tensor field \mathcal{G} . As the labels indicate, the three terms have distinct effects:

- convection: moving data around,
- (fractional) diffusion: regularizing data (which relates to sub sampling by destroying data),
- dilation (+ sign) and erosion (− sign): pooling/sharpening of data.

The geometric interpretation of each of the terms in (23) is illustrated in Fig. 5 for the examples $G/H = \mathbb{R}^2$ and $G/H = \mathbb{M}_2$.

Since the convection vector field c and the metric tensor fields \mathcal{G}_1 and \mathcal{G}_2 are left-invariant, the PDE unit, and so the network layer, is automatically equivariant.

Remark 4.1 (The PDE in a conventional setting). In the usual spatial setting of \mathbb{R}^d with $\alpha = 1$ the PDE (23) can be written as:

$$\begin{cases} \frac{\partial W}{\partial t} = -c \cdot \nabla W - \Delta W \pm \|\nabla W\|^2 & \text{for } p \in G/H, t \geq 0, \\ W(p, 0) = U(p) & \text{for } p \in G/H, \end{cases}$$

which is a more familiar convection/diffusion/dilation/erosion equation.

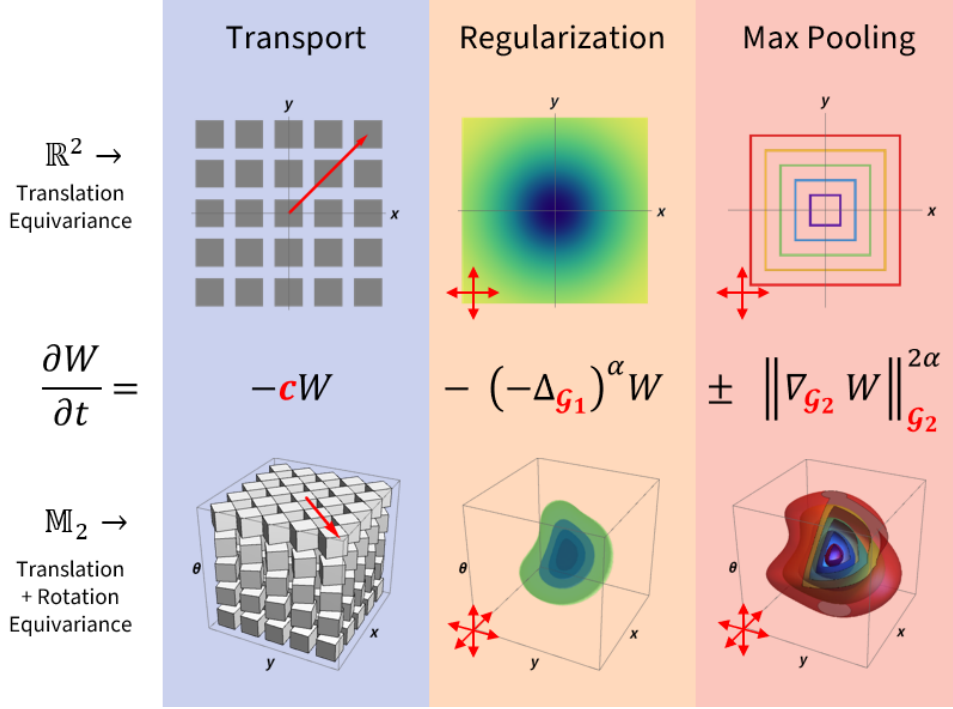


Figure 5: Geometric interpretation of the PDE (23). In the top row we illustrate the effects of this PDEs separate terms in a conventional 2D spatial setting (yielding translation equivariance), in the bottom row we illustrate its effects in \mathbb{M}_2 (yielding rotation and translation equivariance). Recall (18) for the definition of the \mathcal{G}_i which along with the convection vector field c are what we will be training, which in this case controls how much the regularization and max pooling is extended along each of the three dimensions.

4.2 Training

Training the PDE unit comes down to adapting the parameters in the PDE, such as those of the convection vector field c and the metric tensor fields \mathcal{G}_i , in order to minimize a given loss function (the choice of which depends on the application and we will not consider in this article). In this sense, the vector field and the metric tensors are analogous to the weights of this layer.

Since we required the convection vector field and the metric tensor fields to be left-invariant, the parameter space is finite-dimensional as a consequence of Cor. 2.5 and 2.7.

Remark 4.2 (Left-invariant vector fields with respect to $SE(d)$). In the case of the group $SE(d)$ we recap what left-invariant vector fields exist on its homogeneous spaces in the following table.

Homogeneous space	Left-invariant vector fields of the homogeneous space
\mathbb{M}_2	all left-invariant vector fields of the group $SE(d)$
\mathbb{M}_d for $d \geq 3$	vector fields of constant magnitude aligned to the local primary direction (which implies all these vector fields are scalar multiples of each other)
$\mathbb{R}^d, \mathcal{S}^{d-1}$	only the trivial zero vector field

For our main application on \mathbb{M}_2 each PDE unit would have the following 9 trainable parameters:

- 3 parameters to specify the convection vector field as a linear combination of (19),
- 3 parameters to specify the fractional diffusion metric tensor field \mathcal{G}_1 ,
- and 3 parameters to specify the dilation/erosion metric tensor field \mathcal{G}_2 ,

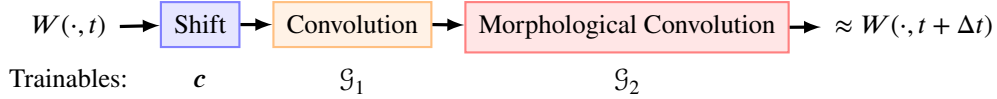
where both metric tensor fields are of the form (18).

Surprisingly for higher dimensions \mathbb{M}_d has less trainable parameters than for $d = 2$. This is caused by the left-invariant vector fields on \mathbb{M}_d for $d \geq 3$ being spanned by a single basis element (20) instead of the three (19) basis elements available for $d = 2$. Since the left-invariant metric tensor fields are determined by only 3 parameters irrespective of dimensions we count a total of 7 parameters for each PDE unit for applications on \mathbb{M}_d for $d \geq 3$.

5 Implementation of a PDE Unit

A PDE unit will be an N -fold repetition of a timestep-unit which is a composition of convection, diffusion, and dilation/erosion substeps, where N is some natural number. These units all take their input as an initial condition of a PDE, and produce as output the solution of a PDE at time $t = T$. The output of a previous timestep-unit is taken as the input for the next timestep-unit, as illustrated in Fig. 1.

The convection, diffusion and dilation/erosion steps are implemented with respectively a shift, convolution, and morphological convolution, as illustrated in the next diagram.



The composition of the substeps does not solve (23) exactly, but for small Δt , it approximates the solution by a principle called *operator splitting*.

We will now discuss each of these substeps separately.

5.1 Convection

The convection step has as input a function $U^1 : G/H \rightarrow \mathbb{R}$ and takes it as initial condition of the PDE

$$\begin{cases} \frac{\partial W^1}{\partial t}(p, t) = -c(p)W^1(\cdot, t) & \text{for } p \in G/H, t \geq 0, \\ W^1(p, 0) = U^1(p) & \text{for } p \in G/H. \end{cases}$$

The output of the layer is the solution of the PDE evaluated at time $t = T$, i.e. the output is the function $p \mapsto W^1(p, T)$.

The solution of the PDE is found by the method of characteristics, and is given by

$$W^{(1)}(p, t) = \left(\mathcal{L}_{\gamma_c(t)} U^1 \right) (p) = U^1 \left(\gamma_c(t)^{-1} \cdot p \right), \quad (24)$$

where $\gamma_c : \mathbb{R} \rightarrow G$ is the curve that satisfies

$$\frac{\partial}{\partial t} (\gamma_c \cdot p) (t) = c (\gamma_c(t) \cdot p), \quad (25)$$

i.e. γ_c is the curve in the group G that induces the integral curves of the left-invariant vector field c on G/H when acting on elements of the homogeneous space.

5.2 Fractional Diffusion

The (fractional) diffusion step solves the PDE

$$\begin{cases} \frac{\partial W^2}{\partial t} = - \left(-\Delta_{\mathcal{G}_2} \right)^\alpha W^2(p, t) & \text{for } p \in G/H, t \geq 0, \\ W^2(p, 0) = U^2(p) & \text{for } p \in G/H. \end{cases} \quad (26)$$

As with (fractional) diffusion on \mathbb{R}^n , there exists a smooth function

$$K_t^\alpha : (0, \infty) \times (G/H) \rightarrow [0, \infty),$$

called the fundamental solution of the α -diffusion equation, such that for every initial condition U^2 , the solution to the PDE (23) is given by the convolution of the function U^2 with the fundamental solution K_t^α

$$W^2(p, t) = (K_t^\alpha *_G U^2(\cdot, t))(p).$$

The convolution $*_G$ is specified by the following definition.

Definition 5.1 (Linear group convolution). Let $H = \text{Stab}_G(p_0)$ be compact with reference element $p_0 \in G/H$, let $f \in L^2(G/H)$ and $k \in L^1(G/H)$ such that:

$$\forall h \in H, p \in G/H : k(h.p) = k(p) \quad (\text{kernel compatibility})$$

then we define:

$$(k *_G f)(p) := \int_G k(g^{-1}.p) f(g.p_0) d\mu_G(g), \quad (27)$$

where μ_G is the left-invariant Haar measure on the group.

In general an analytic expression for the fundamental solution K_t^α requires complicated steerable filter operators [31, Thm. 1 & 2] and for that reason we contend ourselves with more easily computable approximations. For now, let us not elaborate on the quality of those approximations (nor on the involved asymptotics) Our approximations will use make use of the following norm on the Lie algebra of the group G that is induces by a metric tensor field on a homogeneous space.

Definition 5.2 (Homogeneous Lie algebra norm). Let \mathcal{G} be a left-invariant metric tensor field on the homogeneous space G/H with reference element p_0 then

$$\forall v \in T_e G : \|v\|_{\mathcal{G}} := \left\| \frac{\partial}{\partial t} \exp_G(tv) . p_0 \Big|_{t=0} \right\|_{\mathcal{G}|_{p_0}} \quad (28)$$

is its induced norm on the Lie algebra $T_e(G)$ where $\exp_G : T_e(G) \rightarrow G$ is the exponential map of G .

Note that due to the left invariance of \mathcal{G} that we can pick any point $p \in G/H$ instead of p_0 in the definition above and get the same result, so there is no arbitrary choice being made.

For small enough distances from p_0 this norm then approximates the metric $d_{\mathcal{G}}$ of the homogeneous space as (recall (5) for the definition of G_p)

$$d_{\mathcal{G}}(p_0, p) \approx \inf_{g \in G_p} \|\log_G g\|_{\mathcal{G}}, \quad (29)$$

where \log_G is the logarithmic map of G . We label this estimate as follows.

Definition 5.3 (Logarithmic metric estimate). Let \mathcal{G} be a left-invariant metric tensor field on the homogeneous space G/H then we define

$$\rho_{\mathcal{G}}(p) := \inf_{g \in G_p} \|\log_G g\|_{\mathcal{G}}. \quad (30)$$

We can interpret this metric estimate as finding all exponential curves in G whose actions on the homogeneous space connect p_0 (at $t = 0$) to p (at $t = 1$) and then from that set we choose the exponential curve that has the lowest velocity according to the norm in Def. 5.2 and use its velocity as the distance estimate.

Remark 5.4 (Logarithmic metric estimate in principal homogeneous spaces). When we take a principal homogeneous space such as $\mathbb{M}_2 \equiv SE(2)$ with a left-invariant metric tensor field the metric estimate simplifies to

$$\rho_G(g) = \|\log_G g\|_{\mathfrak{g}|_e},$$

hence we see that this construction generalizes the logarithmic estimate, as used in [69, 70], to homogeneous spaces other than the principal.

Remark 5.5 (Logarithmic metric estimate for \mathbb{M}_2). Using the (x, y, θ) coordinates for \mathbb{M}_2 and a left-invariant metric tensor field of the form (18) we formulate the metric estimate in terms of the following auxiliary functions called the exponential coordinates of the first kind:

$$\begin{aligned} c^1(x, y, \theta) &:= \begin{cases} \frac{\theta}{2} \left(y + x \cot \frac{\theta}{2} \right) & \text{if } \theta \neq 0, \\ x & \text{if } \theta = 0, \end{cases} \\ c^2(x, y, \theta) &:= \begin{cases} \frac{\theta}{2} \left(-x + y \cot \frac{\theta}{2} \right) & \text{if } \theta \neq 0, \\ y & \text{if } \theta = 0, \end{cases} \\ c^3(x, y, \theta) &:= \theta. \end{aligned}$$

The logarithmic metric estimate for $SE(2)$ is then given by

$$\rho_G(x, y, \theta) = \sqrt{D_M c^1(x, y, \theta)^2 + D_L c^2(x, y, \theta)^2 + D_A c^3(x, y, \theta)^2},$$

this estimate is illustrated in figure 6 where it is contrasted against the exact metric.

Remark 5.6 (Logarithmic metric estimate for \mathbb{M}_3). On \mathbb{M}_3 using the coordinates $(\mathbf{x}, \mathbf{n}) \in \mathbb{R}^3 \times S^2$ the transformation $g \in SE(3) \equiv \mathbb{R}^3 \times SO(3)$ that minimizes (5.4) is given by $g = (\mathbf{x}, R_{\mathbf{n}})$ where $R_{\mathbf{n}}$ is the in-plane rotation that rotates the reference direction $\mathbf{a} \in S^2$ to the desired direction \mathbf{n} as shown in [69, Thm. 1].

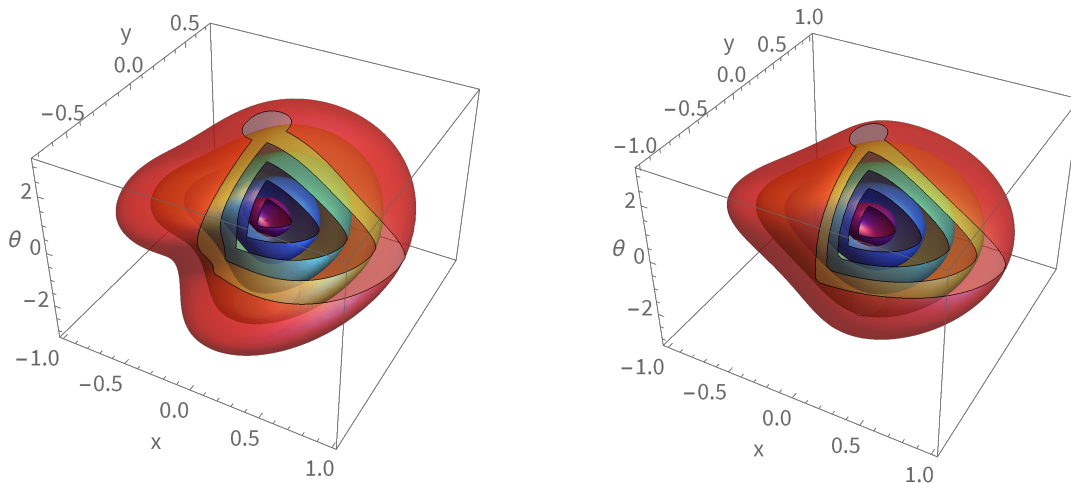


Figure 6: Comparing the ‘exact’ Riemannian distance (left) obtained through numerically solving the Eikonal equation [50] versus the logarithmic metric estimate (right) on $SE(2)$ endowed with a left-invariant Riemannian metric tensor field (18) with $D_M = 1$, $D_L = 2$, $D_A = 1/\pi$.

We can see that the metric estimate ρ_G (and consequently any function of ρ_G) has the necessary compatibility property to be a kernel used in convolutions per Def. 5.1.

Lemma 5.7 (Kernel compatibility of $\rho_{\mathcal{G}}$). *In the same setting as Def. 5.2 and 5.3 we have*

$$\forall h \in H : \rho_{\mathcal{G}}(h.p) = \rho_{\mathcal{G}}(p). \quad (31)$$

Proof. We apply Def. 5.2 and find

$$\rho_{\mathcal{G}}(p) = \inf_{g \in G_p} \left\| \frac{\partial}{\partial t} \exp_G (t \log_G g) \cdot p_0 \Big|_{t=0} \right\|_{\mathcal{G}|_{p_0}}.$$

Due to the left invariance of \mathcal{G} and the fact that $h.p_0 = p_0$ the following equality holds for all $h \in H$:

$$= \inf_{g \in G_p} \left\| (L_h)_* \frac{\partial}{\partial t} \exp_G (t \log_G g) \cdot p_0 \Big|_{t=0} \right\|_{\mathcal{G}|_{p_0}}.$$

This can be rewritten as:

$$= \inf_{g \in G_p} \left\| \frac{\partial}{\partial t} h \exp_G (t \log_G g) \cdot p_0 \Big|_{t=0} \right\|_{\mathcal{G}|_{p_0}}.$$

Now we see that we are optimizing over a set of left-invariant curves whose actions connect p_0 (at $t = 0$) to $h.p$ (at $t = 1$) i.e. we have:

$$\begin{aligned} &= \inf_{g \in G_{h.p}} \left\| \frac{\partial}{\partial t} \exp_G (t \log_G g) \cdot p_0 \Big|_{t=0} \right\|_{\mathcal{G}|_{p_0}} \\ &= \rho_{\mathcal{G}}(h.p). \end{aligned}$$

□

While a general formula for the fundamental solution to the fractional diffusion problem can be obtained via the Fourier transform on the homogeneous space G/H [31], this usually results in difficult expressions. In most cases we can make sufficiently good approximations using the tools that we just developed.

Proposition 5.8 (Approximation of the fractional diffusion kernel). *For small enough t and $\alpha = 1$ or $\alpha = 1/2$ the following estimates can be made*

$$K_t^1(p) \approx K_t^{1,approx}(p) = C_t \exp\left(-\frac{\rho_{\mathcal{G}_2}(p)^2}{4t}\right), \quad (32)$$

$$K_t^{1/2}(p) \approx K_t^{1/2,approx}(p) = C_t \frac{t}{\left(t^2 + \rho_{\mathcal{G}_2}(p)^2\right)^{\frac{\dim(G/H)+1}{2}}} \quad (33)$$

where C_t is the appropriate $L^1(G/H)$ -normalization constant for a given t .

Sketch of proof. The top expression (32) for $\alpha = 1$ is a consequence of the parametrix expansion, cf. [71]. For $\alpha \in (1/2, 1)$ we do not have an analytic approximation but can obtain a general relation involving (strongly continuous) semigroups that are generated by a fractional power of the given generator [72, Section IX.11]. In our case the fractional power of the generator equals $-(\Delta_{\mathcal{G}_2})^\alpha$ and the general relation is given by [31, Eq. (77) and (78)].

Only for $\alpha = 1/2$ does this relation evaluate to an analytic kernel, and this indeed yields the Poisson kernel (33). □

In Fig. 7 we illustrate the shape of the level sets of these kernels for different parameter settings of the involved metric tensor fields.

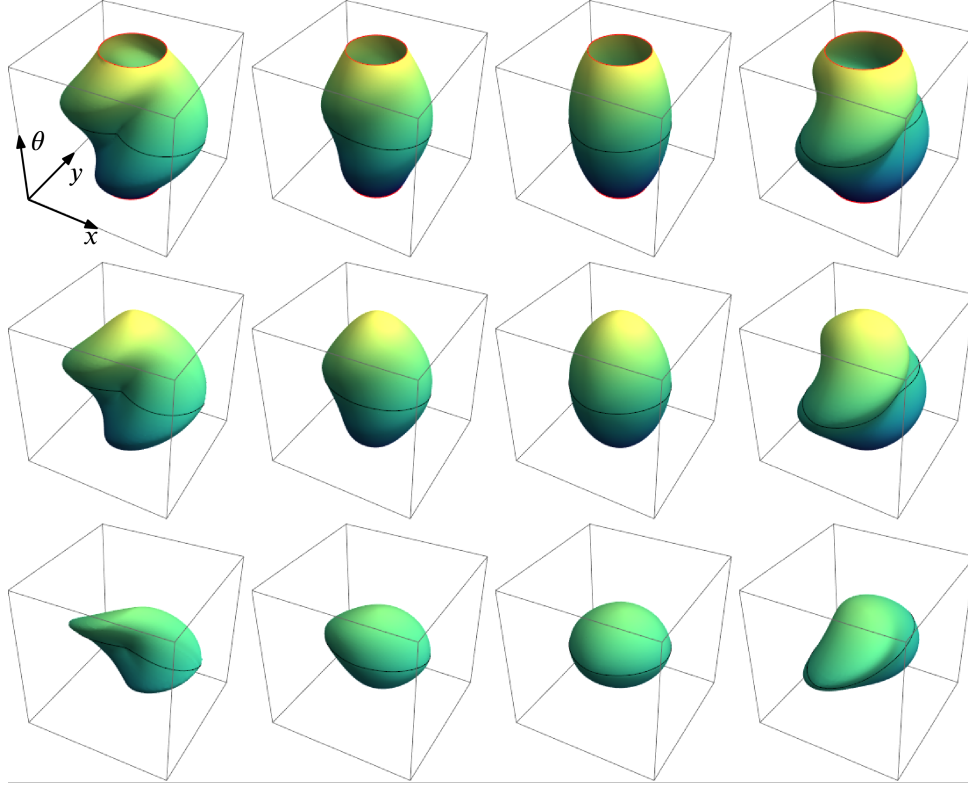


Figure 7: Shapes of the level sets of the kernels on \mathbb{M}_2 for solving fractional diffusion (K_t^α) and dilation/erosion (k_t^α) for various values of the trainable metric tensor field parameters D_M, D_L and D_A . This shape is essentially what is being optimized during the training process of a metric tensor field on \mathbb{M}_2 and does not depend on the choice of α .

5.3 Dilation and Erosion

The dilation/erosion step solves the PDE

$$\begin{cases} \frac{\partial W^3}{\partial t} = \pm \|\nabla_{\mathcal{G}_3} W^3(p, t)\|_{\mathcal{G}_3}^{2\alpha} & \text{for } p \in G/H, t \geq 0, \\ W^3(p, 0) = U^3(p) & \text{for } p \in G/H. \end{cases} \quad (34)$$

By a generalization of the Hopf-Lax formula [73, Ch.10.3], the solution is given by morphological convolution

$$W^3(p, t) = - (k_t^\alpha \square_G - U^3)(p) \quad (35)$$

for the + (dilation) variant and

$$W^3(p, t) = (k_t^\alpha \square_G U^3)(p) \quad (36)$$

for the - (erosion) variant, where the kernel k_t^α (also called the structuring element in the context of morphology) is a proper (i.e. not everywhere equal to ∞) lower semi-continuous function of the type

$$k_t^\alpha : (0, \infty) \times (G/H) \rightarrow \mathbb{R} \cup \{\infty\}. \quad (37)$$

The morphological convolution \square_G (alternatively: the infimal convolution) is specified as follows.

Definition 5.9 (Morphological group convolution). Let $f \in L^\infty(G/H)$, let $k : G/H \rightarrow \mathbb{R} \cup \{\infty\}$ be proper and let $p_0 \in G/H$ be the reference element of the homogeneous space, then we define:

$$(k \square_G f)(p) := \inf_{g \in G} k(g^{-1} \cdot p) + f(g \cdot p_0).$$

Remark 5.10 (Grayscale morphology). Morphological convolution is related to the grayscale morphology operations \oplus (dilation) and \ominus (erosion) on \mathbb{R}^d as follows:

$$\begin{aligned} f_1 \oplus f_2 &= -(-f_1 \square_{\mathbb{R}^d} - f_2), \\ f_1 \ominus f_2 &= f_1 \square_{\mathbb{R}^d} [\mathbf{x} \mapsto -f_2(-\mathbf{x})], \end{aligned}$$

where f_1 and f_2 are proper functions on \mathbb{R}^d . Hence our use of the terms dilation and erosion, but mathematically we will only use \square_G as the actual operation to be performed and avoid \oplus and \ominus .

As with fractional diffusion we do not have a general analytic expression for the fundamental solution to the dilation/erosion problem but we can make the following analytic estimates.

Proposition 5.11 (Approximation of the dilation/erosion kernel). *The morphological convolution kernel k_t^α is for small times t and $\alpha \in (1/2, 1]$ well-approximated by*

$$k_t^\alpha(p) \approx k_t^{\alpha, \text{approx}}(p) = \left(\frac{2\alpha - 1}{(2\alpha)^{2\alpha/(2\alpha-1)}} \right) t^{-\frac{1}{2\alpha-1}} \rho_{\mathcal{G}_3}(p)^{\frac{2\alpha}{2\alpha-1}}, \quad (38)$$

and for $\alpha = 1/2$ by

$$k_t^{1/2}(p) \approx k_t^{1/2, \text{approx}}(p) = \begin{cases} 0 & \text{if } \rho_{\mathcal{G}_3}(p) \leq t, \\ \infty & \text{if } \rho_{\mathcal{G}_3}(p) > t \end{cases} \quad (39)$$

where $\rho_{\mathcal{G}_3}$ is the estimate of the Riemannian distance between p and p_0 induced by the (trainable) left-invariant metric tensor field \mathcal{G}_3 given by Def. 5.3.

Section 6 is dedicated to obtaining the estimates in (38) and (39) as it is a fairly involved process, in the remainder of this section we will demonstrate how morphological convolution is able to generalize some operations commonly used in CNNs.

To get an idea of how the kernel in (38) operates in conjunction with morphological convolution we take $G = G/H = \mathbb{R}$ and see how morphological convolution evolves simple data, the kernels and results at $t = 1$ are shown in Fig. 8. Observe that with α close to $1/2$ (kernel and result in red) that we obtain what amounts to max/min pooling.

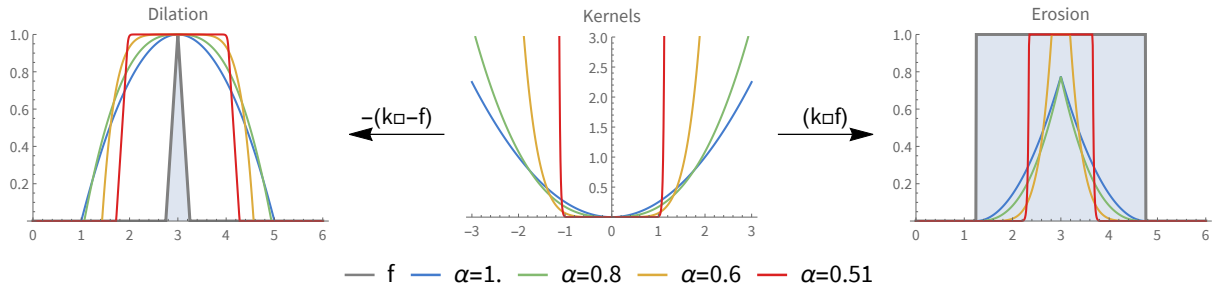


Figure 8: In the center we have kernels of the type (38) in \mathbb{R} for some $\alpha \in (1/2, 1]$ and $t = 1$, which solves dilation/erosion. For $\alpha \rightarrow 1/2$ this kernel converges to the type in Thm. 5.12, i.e. the solution is obtained by max/min pooling. On the left we morphologically convolve a spike (in gray) with a few of these kernels, we see that if $\alpha \rightarrow 1/2$ we get max pooling, conversely we can call the case $\alpha > 1/2$ soft max pooling. On the right we similarly erode a plateau, which for $\alpha \rightarrow 1/2$ yields min pooling.

The level sets of the kernels k_t^α for $\alpha > 1/2$ are of the same shape as for the approximate diffusion kernels, see Fig. 7, for $\alpha = 1/2$ these are the stencils over which we would perform min/max pooling.

5.3.1 Max Pooling as Morphological Convolution

The ordinary max pooling operation commonly found in neural networks can also be seen as a morphological convolution with a kernel for $\alpha = 1/2$.

Proposition 5.12 (Max pooling). *Let $f \in L^\infty(G/H)$, let $S \subset G/H$ be non empty and define $k_S : G/H \rightarrow \mathbb{R} \cup \{\infty\}$ as:*

$$k_S(p) := \begin{cases} 0 & \text{if } p \in S, \\ \infty & \text{else.} \end{cases} \quad (40)$$

Then:

$$-(k_S \square -f)(p) = \sup_{g \in G: g^{-1} \cdot p \in S} f(g \cdot p_0). \quad (41)$$

We can recognize the morphological convolution as a generalized form of max pooling of the function f with stencil S .

Proof. Filling in (40) into definition 5.9 yields:

$$\begin{aligned} -(k_S \square -f)(p) &= -\inf \left\{ \inf_{g \in G: g^{-1} \cdot p \in S} -f(g \cdot p_0), \inf_{g \in G: g^{-1} \cdot p \notin S} -f(g \cdot p_0) + \infty \right\} \\ &= -\inf_{g \in G: g^{-1} \cdot p \in S} -f(g \cdot p_0) \\ &= \sup_{g \in G: g^{-1} \cdot p \in S} f(g \cdot p_0) \end{aligned}$$

□

In particular cases we recover a more familiar form of max pooling as the following corollary shows.

Corollary 5.13 (Euclidean Max Pooling). *Let $G = G/H = \mathbb{R}^n$ and let $f \in C^0(\mathbb{R}^n)$ with $S \subset \mathbb{R}^n$ compact then:*

$$-(k_S \square_{\mathbb{R}^n} -f)(x) = \max_{y \in S} f(x - y).$$

The observation that max pooling is a particular limiting case of morphological convolution allows us to think of the case with $\alpha > 1/2$ as a *soft* variant of max pooling, one that is better behaved under small perturbations in a discretized context.

5.3.2 ReLUs as Morphological Convolution

Max pooling is not the only common CNN operation that can be generalized by morphological convolution as the following theorem shows.

Proposition 5.14 (ReLU). *Let f be a compactly supported continuous function on G/H . Then dilation with the kernel*

$$k_{ReLU, f}(p) := \begin{cases} 0 & \text{if } p = p_0, \\ \sup_{y \in G/H} f(y) & \text{else,} \end{cases}$$

equates to applying a Rectified Linear Unit to the function f :

$$-(k_{ReLU} \square -f)(p) = \max \{0, f(p)\}.$$

Proof. Filling in k into the definition of morphological convolution:

$$\begin{aligned} -(k_{ReLU} \square -f)(p) &= -\inf_{g \in G} k_{ReLU}(g^{-1} \cdot p) - f(g \cdot p_0) \\ &= -\inf_{g \in G} \left\{ \inf_{g^{-1} \cdot p = p_0} -f(g \cdot p_0), \inf_{g^{-1} \cdot p \neq p_0} -f(g \cdot p_0) + \sup_{y \in G/H} f(y) \right\} \\ &= \sup \left\{ f(p), \sup_{z \in G/H: z \neq p} f(z) - \sup_{y \in G/H} f(y) \right\}, \end{aligned}$$

due to the continuity and compact support of f its supremum exists and moreover we have $\sup_{z \in G/H: z \neq p_0} f(z) = \sup_{y \in G/H} f(y)$ and thereby we obtain the required result

$$= \max \left\{ f(p), 0 \right\}.$$

□

We conclude that morphological convolution allows us to:

- do max pooling in an equivariant manner with transformations other than translation,
- do *soft* max pooling that is continuous under domain transformations (illustrated in Fig. 8),
- learn the max pooling region by considering the kernel k (or rather the metric tensor field \mathcal{G}_3) as trainable,
- incorporate the action of a ReLU.

6 Approximations to the Dilation and Erosion Kernels

Recall from Fig. 5 that dilation and erosion PDEs are solved by morphological convolution with a kernel (35)(36). In this section we describe how we obtain the approximation of the morphological kernel from (38). We do this by using a transformation that is able to relate the dilation/erosion PDE with the fractional diffusion PDE as in [47]. We first introduce this so-called Cramér-Fourier transform [46] in \mathbb{R}^d but since we do not know a generalization to the group we introduce an approximate Cramér-Fourier transform via the Lie algebra. It is this approximate transform that we will apply to the approximate diffusion kernel (32) to get an approximate morphological kernel. This double approximation works well in practice for sufficiently small evolution times where the kernels are well localized as we will see in Section 7 in the experiment we performed.

6.1 Cramér-Fourier Transform on Euclidean Space

On \mathbb{R}^d the Cramér-Fourier transform [46] is given as follows.

Definition 6.1 (Cramér-Fourier transform). For functions on \mathbb{R}^d that have a real-valued non-negative Fourier transform we define the Cramér-Fourier transform as

$$\mathcal{C}_{\mathcal{F}} := \mathfrak{F} \circ -\log \circ \mathcal{F}, \quad (42)$$

where \mathcal{F} denotes the Fourier transform, \log denotes the point-wise logarithm and \mathfrak{F} denotes the Fenchel transform (i.e. convex conjugation).

The reason we introduce this transformation is because it relates linear convolution to morphological convolution in the following manner (see [46, Thm.1] for details and proof).

Lemma 6.2 (Cramér-Fourier convolution theorem). *On \mathbb{R}^d the Cramér-Fourier transform relates convolution and morphological convolution in the following manner: let f_1 and f_2 have real-valued non-negative Fourier transforms then*

$$\mathcal{C}_{\mathcal{F}} [f_1 *_{\mathbb{R}^d} f_2] = \mathcal{C}_{\mathcal{F}} [f_1] \square_{\mathbb{R}^d} \mathcal{C}_{\mathcal{F}} [f_2]. \quad (43)$$

We can see how this comes about as the Fourier transform turns a convolution into a multiplication, the logarithm turns the multiplication into an addition and finally the Fenchel transform turns the addition into a morphological convolution.

The second relevant property of the Cramér-Fourier transform is that it relates fractional diffusion with erosion as follows.

Lemma 6.3. *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be differentiable with compact support and a real-valued non-negative Fourier transform then for $\alpha \in [1/2, 1]$:*

$$\mathcal{C}_{\mathcal{F}} [-(-\Delta)^{\alpha} f] = - \left\| \nabla (\mathcal{C}_{\mathcal{F}} f) \right\|^{2\alpha}. \quad (44)$$

Again see [46, Thm. 2] for details and proof.

These equalities allow us to relate the fractional diffusion system in (26) to the dilation/erosion system in (34) and use the approximate solution we have for the first system to construct an approximate solution to the latter system.

Assuming that in fact we are working on \mathbb{R}^d and we have a solution $W^2 : \mathbb{R}^d \times \mathbb{R}_+ \rightarrow \mathbb{R}$ to the system (26) of the form $W^2(\cdot, t) = K_t^\alpha *_{\mathbb{R}^d} U^2$

$$\begin{aligned}
& \frac{\partial W^2}{\partial t} = -(-\nabla)^\alpha W^2 \\
\Rightarrow & \frac{\partial}{\partial t} (K_t^\alpha * U^2) = -(-\nabla)^\alpha (K_t^\alpha * U^2) \\
\Rightarrow & \mathcal{C}_{\mathcal{F}} \left[\frac{\partial}{\partial t} (K_t^\alpha * U^2) \right] = \mathcal{C}_{\mathcal{F}} [-(-\nabla)^\alpha (K_t^\alpha * U^2)] \\
\Rightarrow & \frac{\partial}{\partial t} \mathcal{C}_{\mathcal{F}} [K_t^\alpha * U^2] = - \left\| \nabla \mathcal{C}_{\mathcal{F}} [K_t^\alpha * U^2] \right\|^{2\alpha} \\
\Rightarrow & \frac{\partial}{\partial t} (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square \mathcal{C}_{\mathcal{F}} [U^2]) = - \left\| \nabla (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square \mathcal{C}_{\mathcal{F}} [U^2]) \right\|^{2\alpha}.
\end{aligned}$$

Now choose $U^2 = \mathcal{C}_{\mathcal{F}}^{-1} [U^3]$ and let $W^3 = \mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square U^3$ then we see that W^3 solves the erosion version of the system (34) by performing a morphological convolution with the initial condition U^3 using the kernel $\mathcal{C}_{\mathcal{F}} [K_t^\alpha]$. The solution to the dilation version of (34) now follows from the last expression with a few extra steps:

$$\begin{aligned}
\Rightarrow & \frac{\partial}{\partial t} (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square U^3) = - \left\| \nabla (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square U^3) \right\|^{2\alpha} \\
\Rightarrow & \frac{\partial}{\partial t} (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square -U^3) = - \left\| \nabla (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square -U^3) \right\|^{2\alpha} \\
\Rightarrow & - \frac{\partial}{\partial t} (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square -U^3) = \left\| \nabla (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square -U^3) \right\|^{2\alpha} \\
\Rightarrow & - \frac{\partial}{\partial t} (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square -U^3) = \left\| -\nabla (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square -U^3) \right\|^{2\alpha} \\
\Rightarrow & \frac{\partial}{\partial t} (- (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square -U^3)) = \left\| \nabla (- (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square -U^3)) \right\|^{2\alpha}.
\end{aligned}$$

Where we see that letting $W^3 = - (\mathcal{C}_{\mathcal{F}} [K_t^\alpha] \square -U^3)$ solves the dilation PDE.

Next, we generalize this transform to homogeneous spaces in an approximate manner so that we can estimate the solution to the erosion/dilation PDE on homogeneous spaces the same way.

6.2 Local Approximation of the Cramér-Fourier Transform on Homogeneous Spaces

The Cramér-Fourier transform requires a function on \mathbb{R}^d , or at least a d -dimensional vector space that we can then naturally identify with \mathbb{R}^d . On a homogeneous space we can use the Lie algebra of its group as that vector space and use the exponential and logarithmic maps to translate between the homogeneous space and the group, this leads to an approximate Cramér-Fourier transform.

Definition 6.4 (Approximate Cramér-Fourier transform). For functions on G/H that have the property that the following Fourier transform

$$\mathcal{F} [\mathbf{v} \mapsto f(\exp_G \mathbf{v} \cdot p_0)] \quad (45)$$

is real-valued and non-negative for all $\mathbf{v} \in T_e G$, we define the approximate Cramér-Fourier transform as

$$\mathcal{C}_{\mathcal{F}}^{\text{approx}} f(p) := \inf_{g \in G_p} \mathcal{C}_{\mathcal{F}} [\mathbf{v} \mapsto f(\exp_G \mathbf{v} \cdot p_0)] (\log_G g), \quad (46)$$

where we recall G_p from (5).

Note that above we naturally identified the n dimensional vector space $T_e(G)$ (i.e. the Lie algebra of G) with \mathbb{R}^n . Nevertheless, the reader should keep in mind that $\mathbf{v} \in T_e(G)$ (and not in \mathbb{R}^n).

Remark 6.5 (Approximate Cramér-Fourier transform on \mathbb{M}_2). A function f on \mathbb{M}_2 expressed in terms of coordinates (x, y, θ) is first transformed to a function f_1 on the Lie algebra expressed in coordinates relative to the basis (19) (i.e. the exponential coordinates of the first kind) as:

$$f_1(c^1, c^2, c^3) := \begin{cases} f\left(\frac{c^1 \sin c^3 - c^2(1 - \cos c^3)}{c^3}, \frac{c^1(1 - \cos c^3) + c^2 \sin c^3}{c^3}, c^3\right) & \text{if } c^3 \neq 0, \\ f(c^1, c^2, 0) & \text{if } c^3 = 0. \end{cases} \quad (47)$$

This function now lives on \mathbb{R}^3 and we can apply the Cramér-Fourier transform on it to obtain the function f_2 :

$$f_2 := \mathcal{C}_{\mathcal{F}} f_1, \quad (48)$$

the function f_2 again lives on \mathbb{R}^3 and via the mapping $(c^1, c^2, c^3) \mapsto c^1 \mathcal{A}_1|_e + c^2 \mathcal{A}_2|_e + c^3 \mathcal{A}_3|_e$ on the Lie algebra. We can use the exponential mapping to bring this function back to \mathbb{M}_2 again to complete the approximate transform:

$$(\mathcal{C}_{\mathcal{F}}^{\text{approx}} f)(x, y, \theta) = \begin{cases} f_2\left(\frac{\theta}{2}\left(y + x \cot \frac{\theta}{2}\right), \frac{\theta}{2}\left(-x + y \cot \frac{\theta}{2}\right), \theta\right) & \text{if } \theta \neq 0, \\ f_2(x, y, 0) & \text{if } \theta = 0. \end{cases} \quad (49)$$

6.3 Dilation/Erosion Kernel Approximation

Now that we have developed the approximate Cramér-Fourier transform on homogeneous spaces we can obtain an approximation of the morphological kernel k_t^α .

Proposition 6.6 (Dilation/erosion kernel estimate). *Let $\alpha \in (1/2, 1]$ and let \mathcal{G} be a left-invariant metric tensor field on G/H , then:*

$$k_t^\alpha(p) \approx k_t^{\alpha, \text{approx}}(p) = (\mathcal{C}_{\mathcal{F}}^{\text{approx}} K_t^{\alpha, \text{approx}})(p) = \frac{2\alpha - 1}{(2\alpha)^{2\alpha/(2\alpha-1)}} \frac{\rho_{\mathcal{G}}(p)^{2\alpha/(2\alpha-1)}}{t^{1/(2\alpha-1)}}. \quad (50)$$

For $\alpha \rightarrow 1/2$ this converges to:

$$k_t^{1/2, \text{approx}}(p) = \begin{cases} 0 & \text{if } \rho_{\mathcal{G}}(p) \leq t, \\ \infty & \text{elsewhere.} \end{cases} \quad (51)$$

These estimates hold for all $p \in G/H$ for sufficiently small $t > 0$.

Proof of the Formula. We leave analyzing how good the approximation is for later work (following ideas in [70]) and for now only derive the analytic formula for what we take to be a good approximation to use practice. From the construction of $K_t^{\alpha, \text{approx}}$ we have that

$$\mathcal{F}[v \mapsto K_t^{\alpha, \text{approx}}(\exp_G v.p_0)] = e^{-t \|v\|_{\mathcal{G}}^{2\alpha}},$$

which is real valued and positive, taking $-\log$ yields

$$v \mapsto t \|v\|_{\mathcal{G}}^{2\alpha}.$$

Applying the Fenchel transform \mathfrak{F} to this (where we use the Riesz representations w of the duals $\omega \in T_e^*G$ instead of the duals themselves)

$$\mathfrak{F}[t \|\cdot\|_{\mathcal{G}}^{2\alpha}](w) = \sup_{v \in T_e G} \left\{ (w, v)_{\mathcal{G}} - t \|v\|_{\mathcal{G}}^{2\alpha} \right\}.$$

Obviously to maximize this v must be chosen aligned with \mathbf{w} (assume $\neq \mathbf{0}$) and pointing in the same direction, i.e. $\mathbf{v} = \lambda \mathbf{w} / \|\mathbf{w}\|_{\mathcal{G}}^2$ for some $\lambda > 0$:

$$\begin{aligned} &= \sup_{\lambda > 0} \left\{ \left(\mathbf{w}, \lambda \mathbf{w} / \|\mathbf{w}\|_{\mathcal{G}}^2 \right)_{\mathcal{G}} - t \left\| \lambda \mathbf{w} / \|\mathbf{w}\|_{\mathcal{G}}^2 \right\|_{\mathcal{G}}^{2\alpha} \right\} \\ &= \sup_{\lambda > 0} \left\{ \lambda - t \frac{\lambda^{2\alpha}}{\|\mathbf{w}\|_{\mathcal{G}}^{2\alpha}} \right\}. \end{aligned}$$

Now we consider two cases.

Case 1 : $\alpha = 1/2$

$$\begin{aligned} \mathfrak{F} \left[t \|\cdot\|_{\mathcal{G}}^{2\alpha} \right] (\mathbf{w}) &= \sup_{\lambda > 0} \left\{ \lambda - t \frac{\lambda}{\|\mathbf{w}\|_{\mathcal{G}}} \right\} \\ &= \sup_{\lambda > 0} \left\{ \lambda \left(1 - \frac{t}{\|\mathbf{w}\|_{\mathcal{G}}} \right) \right\} \\ &= \begin{cases} 0 & \text{if } \|\mathbf{w}\|_{\mathcal{G}} \leq t, \\ \infty & \text{if } \|\mathbf{w}\|_{\mathcal{G}} > t. \end{cases} \end{aligned}$$

Next taking the mapping to the homogeneous space we get

$$\begin{aligned} (\mathcal{C}_{\mathcal{F}}^{\text{approx}} K_t^{\alpha, \text{approx}}) (p) &= \inf_{g \in G_p} \mathfrak{F} \left[t \|\cdot\|_{\mathcal{G}}^{2\alpha} \right] (\log_G g) \\ &= \inf_{g \in G_p} \begin{cases} 0 & \text{if } \|\log_G g\|_{\mathcal{G}} \leq t, \\ \infty & \text{if } \|\log_G g\|_{\mathcal{G}} > t, \end{cases} \\ &= \begin{cases} 0 & \text{if } \inf_{g \in G_p} \|\log_G g\|_{\mathcal{G}} \leq t, \\ \infty & \text{if } \inf_{g \in G_p} \|\log_G g\|_{\mathcal{G}} > t, \end{cases} \\ &= \begin{cases} 0 & \text{if } \rho_{\mathcal{G}}(p) \leq t, \\ \infty & \text{if } \rho_{\mathcal{G}}(p) > t, \end{cases} \end{aligned}$$

which proves (51).

Case 2 : $\alpha \in (1/2, 1]$

Seeing that the objective function is concave we apply the first order test to find the supremum is reached for

$$\lambda = \frac{\|\mathbf{w}\|_{\mathcal{G}}^{\frac{2\alpha}{2\alpha-1}}}{(2\alpha t)^{\frac{1}{2\alpha-1}}},$$

after substituting and simplifying this yields

$$\mathfrak{F} \left[t \|\cdot\|_{\mathcal{G}}^{2\alpha} \right] (\mathbf{w}) = \frac{2\alpha - 1}{(2\alpha)^{2\alpha/(2\alpha-1)}} \frac{\|\mathbf{w}\|_{\mathcal{G}}^{2\alpha/(2\alpha-1)}}{t^{1/(2\alpha-1)}},$$

finally taking the mapping to the homogeneous space just like in the first case we confirm (50). \square

7 Proof of Concept

To verify that our PDE-based framework is a sensible one to adopt for CNN applications we perform an experiment to see if the incorporation of morphological convolution in G-CNNs can improve performance. For this experiment

we will take a G-CNN as used in retinal vessel segmentation [40] and augment it with a dilation sub-layer. We compare the performance of this augmented network against the non-augmented and a conventional spatial CNN of comparable size making the experiment consisting of the following three networks.

- Spatial CNN, consisting of
 - 6 convolution layers,
 - 34580 parameters.
- $SE(2)$ CNN, consisting of
 - lifting layer, 4 group convolution layers, projection layer,
 - 33916 parameters.
- Augmented $SE(2)$ CNN, consisting of
 - lifting layer, 4 group convolution layers, dilation layer, projection layer,
 - 33916 parameters.

These networks are trained repeatedly on the same dataset for the same amount of epochs after which we evaluate their performance. For details on the networks and the training setup see [15, §3.4]. We compare the performance of these networks via the cross-entropy loss function and the area under ROC curve, the results are summarized in Fig. 9 and show a marked improvement from the included PDE-based morphological convolution layer.

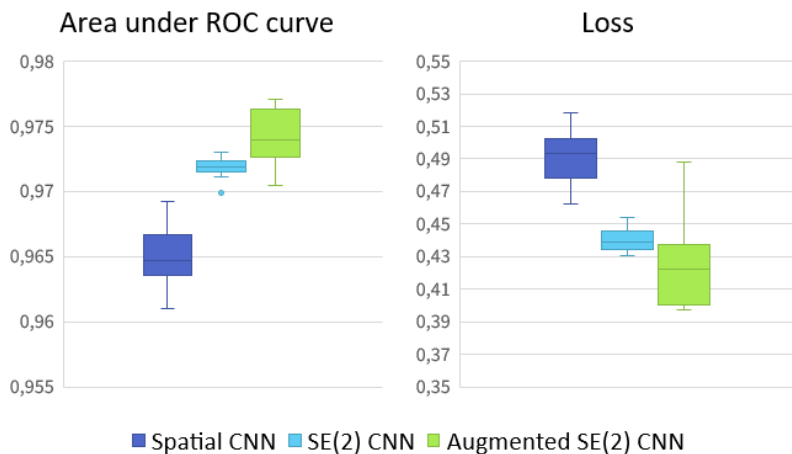


Figure 9: Comparing the performance of our proof-of-concept network against a conventional CNN and a G-CNN. We measure the area under the ROC curve between the networks output versus the ground truth (left) and the cross entropy loss function (right).

To see how well our dilation kernel approximation (Prop. 5.11) holds up in a practical application we compare its discretization as used in the augmented $SE(2)$ CNN (with parameters $D_M = 1$, $D_L = 2$, $D_A = \pi^{-1}$) against one based on the Eikonal solution. The comparison can be seen in Fig. 10 from which we may conclude that our approximation is sufficiently accurate for use in this type of application.

8 Concluding Remarks

In this paper we have developed a general mathematical framework of equivariant CNNs based on geometric PDEs. PDEs are well studied mathematical objects and we have shown how common CNN operations like max pooling and ReLUs naturally arise from them. These insights not only allow for the geometric interpretation of CNNs and related architectures but open up new avenues for the study and development of these types of networks.

In more detail the main conclusions and results of this manuscript in chronological order are as follows.

We have given the tools in the setting of homogeneous spaces, namely left invariant vector and tensor fields as the fundamental building blocks for equivariant PDEs (see for example Prop. 2.9).

We characterized how input data can be transformed from its given domain to a homogeneous space in an equivariant

manner (by (21) for linear transforms) and how it can be projected back to its domain.

We have explained how the PDE unit solves the PDE (23) by means of linear and morphological convolutions with geometrically meaningful kernels, recall Fig. 5. We greatly reduced the amount of trainable parameters by respecting the equivariance constraint.

We have given tangible analytical approximations of the fundamental solutions for fractional diffusion and fractional dilation/erosion in Prop. 5.8 respectively Prop. 5.11. Furthermore we have shown the strength of morphological convolutions in G-CNNs as they allow for soft max-pooling, recall Fig. 8 over geodesic balls in the homogeneous space. Even the ReLU operator can be expressed as a morphological convolution (and can potentially be replaced by it).

We also provided an intrinsic relation between the (approximate) linear and morphological fundamental solutions by way of an approximate Cramér-Fourier transform, recall Prop. 6.6.

With a modest proof-of-concept experiment we have verified that PDE-G-CNNs can considerably improve performance over G-CNNs in the context of automatic vessel segmentation. As shown in [40] G-CNNs have significant advantages over conventional CNNs and we expect that our PDE framework can improve on these results in other applications as well.

Author Contributions

All authors have had significant contributions to this article. Roughly the main contributions are as follows.

- Bart Smets: writer, generalization from groups to homogeneous spaces, proof-of-concept implementation, ReLU proposition.
- Jim Portegies: generalization from groups to homogeneous spaces, embedding, overall article structure, introduction, kernel approximations.
- Erik Bekkers: the proof-of-concept experiment implementation and network design[40].
- Remco Duits: idea of the PDE layer/unit in CNNs, framework formulation on groups, maxpool proposition, kernel approximations, approximate Cramér-Fourier transform.

References

- [1] M. Welk and J. Weickert. “PDE Evolutions for M-Smoother: From Common Myths to Robust Numerics”. In: *Scale Space and Variational Methods in Computer Vision*. Ed. by J. Lellmann, M. Burger, and J. Moder-sitzki. Cham: Springer International Publishing, 2019, pp. 236–248.
- [2] J. Fadili, G. Kutyniok, G. Peyré, G. Plonka-Hoch, and G. Steidl. “Guest Editorial: Mathematics and Image Analysis”. In: *Journal of Mathematical Imaging and Vision* 52.3 (July 2015), pp. 315–316.
- [3] G. Peyré, M. Péchaud, R. Keriven, and L. D. Cohen. “Geodesic Methods in Computer Vision and Graphics”. In: *Found. Trends. Comput. Graph. Vis.* 5.3 (2010), pp. 197–397.
- [4] A. Dubrovina-Karni, G. Rosman, and R. Kimmel. “Multi-Region Active Contours with a Single Level Set Function”. In: *IEEE PAMI* 37.8 (2015), pp. 1585–1601.
- [5] M. Burger, A. Sawatzky, and G. Steidl. *First Order Algorithms in Variational Image Processing*. Ed. by Roland Glowinski, Stanley J. Osher, and Wotao Yin. Cham: Springer International Publishing, 2016, pp. 345–407.
- [6] C. Sbert J. Duran M. Moeller and D. Cremers. “Collaborative Total Variation: A General Framework for Vectorial TV Models”. In: *SIAM SIIMS* 9.1 (2016), pp. 116–151.
- [7] J. Weickert, S. Grewenig, C. Schroers, and A. Bruhn. “Cyclic Schemes for PDE-Based Image Analysis”. In: *International Journal of Computer Vision* 118.3 (July 2016), pp. 275–299.
- [8] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001. DOI: [10.1017/CB09780511626319](https://doi.org/10.1017/CB09780511626319).
- [9] J.A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

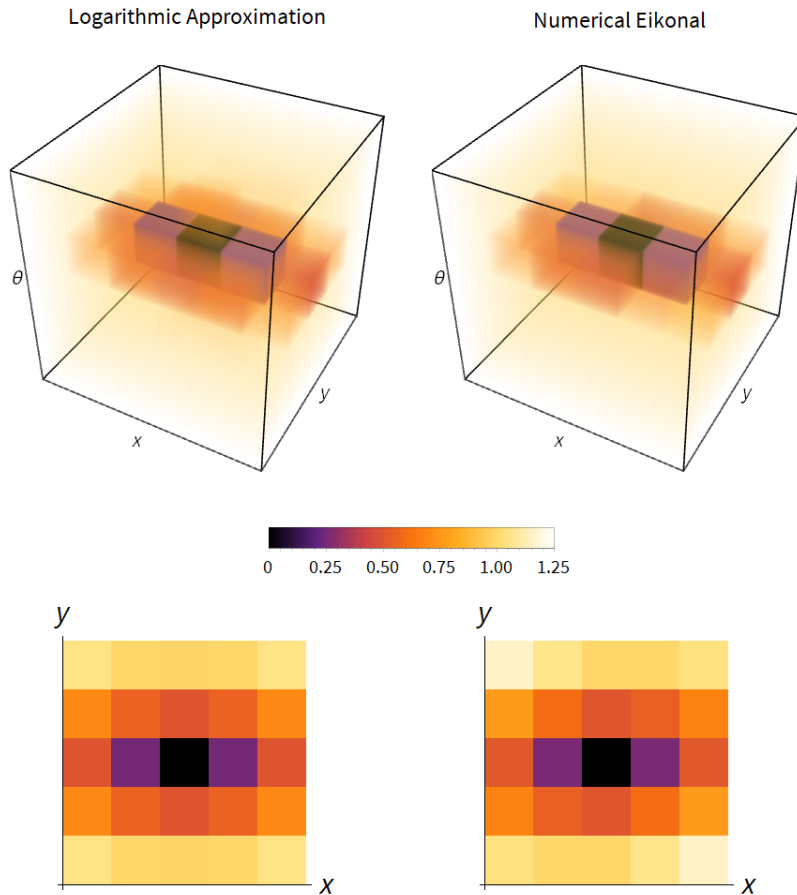


Figure 10: Comparing a discretized ($5 \times 5 \times 5$) approximate dilation kernel as used in the augmented $SE(2)$ CNN (left) versus the numerically exact kernel (right) as obtained by solving the Eikonal equation on the same color scale. Top row: the kernels in \mathbb{M}_2 , bottom row: minimum projection along the orientation dimension. For all practical purposes they are interchangeable.

- [10] J. Weickert. “Theoretical foundations of anisotropic diffusion in image processing.” In: *Computing, Suppl.* 11 (1996), pp. 221–236.
- [11] J.M. Morel and S. Solimini. *Variational methods in image segmentation: with seven image processing experiments*. Progress in nonlinear differential equations and their applications. Birkhäuser, 1995. ISBN: 9780817637200. URL: <https://books.google.nl/books?id=rABEAQAIAAJ>.
- [12] Giovanna Citti, Benedetta Franceschiello, Gonzalo Sanguinetti, and Alessandro Sarti. “Sub-Riemannian mean curvature flow for image processing”. In: *SIAM Journal on Imaging Sciences* 9.1 (2016), pp. 212–237.
- [13] A. Chambolle and T. Pock. “A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging”. In: *Journal of Mathematical Imaging and Vision* 40.1 (2011), pp. 120–145.
- [14] Antonin Chambolle and Thomas Pock. “Total roto-translational variation”. In: *Numerische Mathematik* 142.3 (July 2019), pp. 611–666. ISSN: 0945-3245. DOI: [10.1007/s00211-019-01026-w](https://doi.org/10.1007/s00211-019-01026-w). URL: <https://doi.org/10.1007/s00211-019-01026-w>.
- [15] B.M.N. Smets. “Geometric Image Denoising and Machine Learning”. Master thesis. TU Eindhoven, 2019. URL: <https://bmnsnets.com/publication/snets2019msc/>.
- [16] Remco Duits. “Perceptual organization in image analysis: a mathematical approach based on scale, orientation and curvature”. PhD thesis. Eindhoven University of Technology, 2005.

- [17] Remco Duits, Michael Felsberg, Gösta Granlund, and Bart ter Haar Romeny. “Image analysis and reconstruction using a wavelet transform constructed from a reducible representation of the Euclidean motion group”. In: *International Journal of Computer Vision* 72.1 (2007), pp. 79–102.
- [18] M. H. J. Janssen, A. J. E. M. Janssen, E. J. Bekkers, J. Oliván Bescós, and R. Duits. “Design and Processing of Invertible Orientation Scores of 3D Images”. In: *Journal of Mathematical Imaging and Vision* 60.9 (Nov. 1, 2018), pp. 1427–1458. ISSN: 1573-7683. DOI: [10.1007/s10851-018-0806-0](https://doi.org/10.1007/s10851-018-0806-0). URL: <https://doi.org/10.1007/s10851-018-0806-0>.
- [19] B. Franceschiello, A. Mashtakov, G. Citti, and A. Sarti. “Geometrical optical illusion via sub-Riemannian geodesics in the roto-translation group”. In: *Differential Geometry and its Applications* 65 (2019), pp. 55–77.
- [20] G. Citti and A. Sarti. “A cortical based model of perceptual completion in the roto-translation space”. In: *Journal of Mathematical Imaging and Vision* 24.3 (2006), pp. 307–326.
- [21] R. Duits and E. M. Franken. “Left Invariant Parabolic Evolution Equations on $SE(2)$ and Contour Enhancement via Invertible Orientation Scores, Part I: Linear Left-Invariant Diffusion Equations on $SE(2)$ ”. In: *Quarterly of Applied mathematics, AMS* 68 (June 2010), pp. 255–292.
- [22] R. Duits and E. M. Franken. “Left Invariant Parabolic Evolution Equations on $SE(2)$ and Contour Enhancement via Invertible Orientation Scores, Part II: Nonlinear Left-Invariant Diffusion Equations on Invertible Orientation Scores”. In: *Quarterly of Applied mathematics, AMS* 68 (June 2010), pp. 293–331.
- [23] J. Zhang, R. Duits, B.M. ter Haar Romeny, and G.R. Sanguinetti. “Numerical Approaches for Linear Left-invariant Diffusions on $SE(2)$, their Comparisons to Exact Solutions, and their Applications in Retinal Imaging”. In: *Numerical Mathematics: Theory Methods and Applications* 9.1 (Jan. 2016), pp. 1–50.
- [24] U. Boscaïn, R. A. Chertovskih, J. P. Gauthier, and A. O. Remizov. “Hypoelliptic Diffusion and Human Vision: A Semidiscrete New Twist”. In: *SIAM Journal on Imaging Sciences* 7.2 (2014), pp. 669–695.
- [25] M. Bertalmío, L. Calatroni, V. Franceschi, B. Franceschiello, and D. Prandi. “A Cortical-Inspired Model for Orientation-Dependent Contrast Perception: A Link with Wilson-Cowan Equations”. In: *Scale Space and Variational Methods in Computer Vision*. Ed. by Jan Lellmann, Martin Burger, and Jan Modersitzki. Cham: Springer International Publishing, 2019, pp. 472–484.
- [26] R. Duits, H. Fuehr, B.J. Janssen, L.M.J. Florack, and H.A.C. van Assen. “Evolution equations on Gabor transforms and their applications”. In: *ACHA* 35.3 (2013), pp. 483–526.
- [27] D. Barbieri, G. Citti, G. Cocci, and A. Sarti. “A Cortical-Inspired Geometry for Contour Perception and Motion Integration”. In: *Journal of Mathematical Imaging and Vision* 49.3 (2014), pp. 511–529.
- [28] J. Petitot. “The neurogeometry of pinwheels as a sub-Riemannian contact structure”. In: *Journal of Physiology - Paris* 97 (2003), pp. 265–309.
- [29] M. Felsberg, P-E. Forssen, and H. Scharr. “Channel Smoothing: Efficient robust smoothing of low-level signal features”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2006), pp. 209–222.
- [30] P. Savadjiev, G.J. Strijkers, A.J. Bakermans, E. Piuze, S.W. Zucker, and K. Siddiqi. “Heart wall myofibers are arranged in minimal surfaces to optimize organ function”. In: *PNAS* 109.24 (2012), pp. 9248–9253.
- [31] Remco Duits, Erik Bekkers, and Alexey Mashtakov. “Fourier transform on the homogeneous space of 3D positions and orientations for exact solutions to linear PDEs”. In: *Entropy* 21.1 (2019), p. 38.
- [32] P. Momayyez-Siahkal and K. Siddiqi. “3D Stochastic Completion Fields for Fiber Tractography”. In: *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit.* June 2009, pp. 178–185. DOI: [10.1109/CVPRW.2009.5204044](https://doi.org/10.1109/CVPRW.2009.5204044).
- [33] Taco S Cohen and Max Welling. “Group equivariant convolutional networks”. In: *Int. Conf. on Machine Learning*. 2016, pp. 2990–2999.
- [34] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. “Exploiting cyclic symmetry in convolutional neural networks”. In: *arXiv preprint arXiv:1602.02660* (2016).
- [35] Sander Dieleman, Kyle W Willett, and Joni Dambre. “Rotation-invariant convolutional neural networks for galaxy morphology prediction”. In: *Monthly notices of the royal astronomical society* 450.2 (2015), pp. 1441–1459.

- [36] Marysia Winkels and Taco S Cohen. “3d g-cnns for pulmonary nodule detection”. In: *arXiv preprint arXiv:1804.04656* (2018).
- [37] Daniel Worrall and Gabriel Brostow. “Cubenet: Equivariance to 3d rotation and translation”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 567–584.
- [38] Erik Johannes Bekkers, Marco Loog, Bart M ter Haar Romeny, and Remco Duits. “Template matching via densities on the roto-translation group”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.2 (2017), pp. 452–466.
- [39] Edouard Oyallon and Stéphane Mallat. “Deep roto-translation scattering for object classification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2865–2873.
- [40] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. “Roto-translation covariant convolutional networks for medical image analysis”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 440–448. URL: <https://arxiv.org/abs/1804.03393>.
- [41] Maurice Weiler, Fred A Hamprecht, and Martin Storath. “Learning steerable filters for rotation equivariant CNNs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 849–858.
- [42] Taco Cohen, Mario Geiger, and Maurice Weiler. “A General Theory of Equivariant CNNs on Homogeneous Spaces”. In: *arXiv preprint arXiv:1811.02017* (2018).
- [43] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. “Harmonic networks: Deep translation and rotation equivariance”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5028–5037.
- [44] Risi Kondor and Shubhendu Trivedi. “On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, July 2018, pp. 2747–2755. URL: <http://proceedings.mlr.press/v80/kondor18a.html>.
- [45] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. “Learning so(3) equivariant representations with spherical cnns”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 52–68.
- [46] Martin Schmidt and Joachim Weickert. “Morphological counterparts of linear shift-invariant scale-spaces”. In: *Journal of Mathematical Imaging and Vision* 56.2 (2016), pp. 352–366.
- [47] Marianne Akian, Jean-Pierre Quadrat, and Michel Viot. “Bellman processes”. In: *11th International Conference on Analysis and Optimization of Systems Discrete Event Systems*. Springer. 1994, pp. 302–311.
- [48] Bernhard Burgeth, Martin Welk, Christian Feddern, and Joachim Weickert. “Morphological operations on matrix-valued images”. In: *European Conference on Computer Vision*. Springer. 2004, pp. 155–167.
- [49] Remco Duits, Tom Dela Haije, Eric Creusen, and Arpan Ghosh. “Morphological and linear scale spaces for fiber enhancement in DW-MRI”. In: *Journal of Mathematical Imaging and Vision* 46.3 (2013), pp. 326–368.
- [50] Erik J Bekkers, Remco Duits, Alexey Mashtakov, and Gonzalo R Sanguinetti. “A PDE approach to data-driven sub-Riemannian geodesics in SE(2)”. In: *SIAM Journal on Imaging Sciences* 8.4 (2015), pp. 2740–2770.
- [51] Thomas CJ Dela Haije, Remco Duits, and Chantal MW Tax. “Sharpening fibers in diffusion weighted MRI via erosion”. In: *Visualization and processing of tensors and higher order descriptors for multi-valued data*. Springer, 2014, pp. 97–126.
- [52] Upanshu Sharma and Remco Duits. “Left-invariant evolutions of wavelet transforms on the similitude group”. In: *Applied and Computational Harmonic Analysis* 39.1 (2015), pp. 110–137.
- [53] Julius Hannink, Remco Duits, and Erik Bekkers. “Crossing-preserving multi-scale vesselness”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2014, pp. 603–610.
- [54] Erik J Bekkers. *B-Spline CNNs on Lie Groups*. 2019. arXiv: [1909.12057](https://arxiv.org/abs/1909.12057) [cs.LG].

- [55] Maurice Weiler and Gabriele Cesa. “General E (2)-Equivariant Steerable CNNs”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 14334–14345.
- [56] Gregory S Chirikjian and Alexander B Kyatkin. “An operational calculus for the Euclidean motion group with applications in robotics and polymer science”. In: *Journal of Fourier Analysis and Applications* 6.6 (2000), pp. 583–606.
- [57] Erik Franken, Markus van Almsick, Peter Rongen, Luc Florack, and Bart ter Haar Romeny. “An efficient method for tensor voting using steerable filters”. In: *European Conference on Computer Vision*. Springer. 2006, pp. 228–240.
- [58] Marco Reisert. “Group integration techniques in pattern analysis—A kernel view”. In: (2008).
- [59] E Weinan. “A proposal on machine learning via dynamical systems”. In: *Communications in Mathematics and Statistics* 5.1 (2017), pp. 1–11.
- [60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [61] Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. “Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations”. In: *arXiv preprint arXiv:1710.10121* (2017).
- [62] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. “Neural ordinary differential equations”. In: *Advances in neural information processing systems*. 2018, pp. 6571–6583.
- [63] Yunjin Chen, Wei Yu, and Thomas Pock. “On learning optimized reaction diffusion processes for effective image restoration”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 5261–5269.
- [64] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. “PDE-net: Learning PDEs from data”. In: *arXiv preprint arXiv:1710.09668* (2017).
- [65] Eldad Haber and Lars Ruthotto. “Stable architectures for deep neural networks”. In: *Inverse Problems* 34.1 (2017), p. 014004.
- [66] Takashi Koda. “An introduction to the geometry of homogeneous spaces”. In: *Proceedings of The Thirteenth International Workshop on Diff. Geom.* Vol. 13. 2009, pp. 121–144.
- [67] Jeffrey M Lee, Bennett Chow, Sun-Chin Chu, David Glickenstein, Christine Guenther, James Isenberg, Tom Ivey, Dan Knopf, Peng Lu, Feng Luo, et al. “Manifolds and differential geometry”. In: *Topology* 643 (2009), p. 658.
- [68] Wolfgang Arendt and Alexander V Bukhvalov. “Integral representations of resolvents and semigroups”. In: *Forum Mathematicum*. Vol. 6. 6. Walter de Gruyter, Berlin/New York. 1994, pp. 111–136.
- [69] Jorg Portegies, Gonzalo Sanguinetti, Stephan Meesters, and Remco Duits. “New approximation of a scale space kernel on SE (3) and applications in neuroimaging”. In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2015, pp. 40–52.
- [70] AFM Ter Elst and Derek W Robinson. “Weighted subcoercive operators on Lie groups”. In: *Journal of Functional Analysis* 157.1 (1998), pp. 88–163.
- [71] Marcel Berger, Paul Gauduchon, and Edmond Mazet. “Le spectre d’une variété riemannienne”. In: *Le Spectre d’une Variété Riemannienne*. Springer, 1971, pp. 141–241.
- [72] Kosaku Yosida. *Functional analysis*. springer, 1968.
- [73] Lawrence C Evans. “Partial differential equations”. In: *Providence, RI* (1998).