



PDE-Based Group Equivariant Convolutional Neural Networks

Bart M. N. Smets¹ · Jim Portegies² · Erik J. Bekkers³ · Remco Duits¹

Received: 9 July 2021 / Accepted: 3 July 2022
© The Author(s) 2022

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 geometrically meaningful PDE-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 in addition to the standard translation equivariance of 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. We will discuss our PDE-based G-CNNs (PDE-G-CNNs) in a general homogeneous space setting while also going into the specifics of our primary case of interest: roto-translation equivariance. We solve the PDE of interest by a combination of linear group convolutions and nonlinear morphological group convolutions with analytic kernel approximations that we underpin with formal theorems. Our kernel approximations allow for fast GPU-implementation of the PDE-solvers; we release our implementation with this article in the form of the LieTorch extension to PyTorch, available at <https://gitlab.com/bsmetsjr/lietorch>. Just like for linear convolution, a morphological convolution is specified by a kernel that we train in our PDE-G-CNNs. In PDE-G-CNNs, we do not use non-linearities such as max/min-pooling and ReLUs as they are already subsumed by morphological convolutions. We present a set of experiments to demonstrate the strength of the proposed PDE-G-CNNs in increasing the performance of deep learning-based imaging applications with far fewer parameters than traditional CNNs.

Keywords PDE · Group equivariance · Deep learning · Morphological scale-space

1 Introduction

In this work, we introduce *PDE-based Group CNNs*. The key idea is to replace the typical trifecta of convolution, pooling and ReLUs found in CNNs with a Hamilton–Jacobi type evolution PDE, or more accurately a solver for a Hamilton–Jacobi type PDE. This substitution is illustrated in Fig. 1 where we retain (channel-wise) affine combinations as the means of composing feature maps.

The PDE we propose to use in this setting comes from the geometric image analysis world [1–12]. It was chosen based on the fact that it exhibits similar behavior on images as traditional CNNs do through convolution, pooling and ReLUs. Additionally it can be formulated on Lie groups to yield equivariant processing, which makes our PDE approach compatible with Group CNNs [13–25]. Finally, an approximate solver for our PDE can be efficiently implemented on modern highly parallel hardware, making the choice a practical one as well.

Our solver uses the operator splitting method to solve the PDE in a sequence of steps, each step corresponding to a term of the PDE. The sequence of steps for our PDE is illustrated in Fig. 2. The morphological convolutions that are used to solve for the nonlinear terms of the PDE are a key aspect of our design. Normally, morphological convolutions are considered on \mathbb{R}^d [26,27], but when extended to Lie groups such as $SE(d)$ they have many benefits in applications (e.g., crossing-preserving flow [28] or tracking [29,30]). Using morphological convolutions allows our net-

✉ Bart M. N. Smets
b.m.n.smets@tue.nl

¹ Department of Mathematics and Computer Science, Cluster: CASA (Center for Analysis, Scientific Computing and Applications), Research Groups: Geometric Learning and Differential Geometry, Eindhoven University of Technology, Eindhoven, The Netherlands

² Applied Analysis, Eindhoven University of Technology, Eindhoven, The Netherlands

³ Machine Learning Lab, Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands

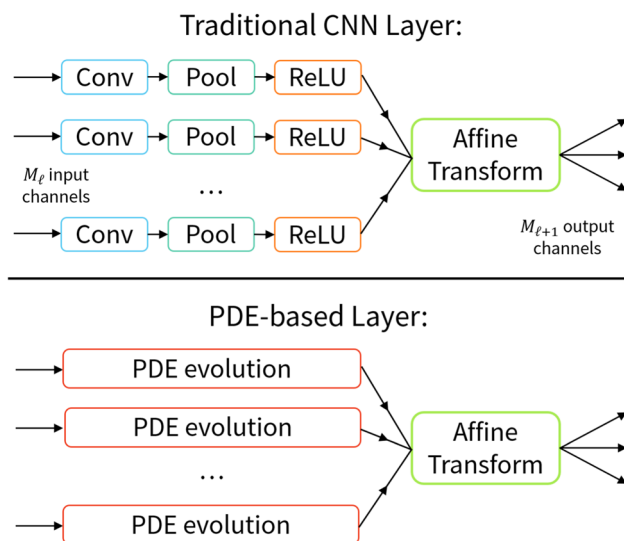


Fig. 1 In a PDE-based CNN, we replace the traditional convolution, pooling and ReLU operations by a PDE solver. The inputs of a given layer serve as initial conditions for a set of evolution PDEs; the outputs consist of affine combinations of the solutions of those PDEs at a fixed point in time. The parameters of the PDE become the trainable weights (alongside the affine parameters) over which we optimize

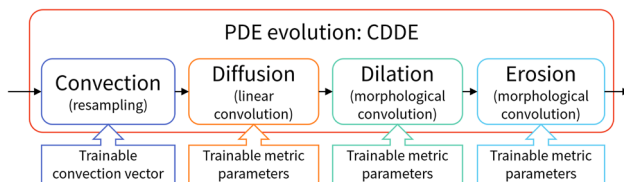


Fig. 2 Our Hamilton–Jacobi type PDE of choice contains a convection, diffusion, dilation and erosion term (CDDE for short). Through operator splitting, we solve for these terms separately by using resampling (for convection), linear convolution (for diffusion) and morphological convolution (for dilation and erosion)

work to have trainable non-linearities instead of the fixed non-linearities in (G-)CNNs.

The theoretical contribution of this paper consists of providing good analytical approximations to the kernels that go in the linear and morphological convolutions that solve our PDE. On \mathbb{R}^n , the formulation of these kernels is reasonably straightforward, but in order to achieve group equivariance we need to generalize them on homogeneous spaces.

Instead of training kernel weights, our goal is training the coefficients of the PDE. The coefficients of our PDE have the benefit of yielding geometrically meaningful parameters from a image analysis point of view. Additionally, we will need (much) less PDE parameters than kernel weights to achieve a given level of performance in image segmentation and classification tasks, arguably the greatest benefit of our approach.

This paper is a substantially extended journal version of [31] presented at the SSVM 2021 conference.

1.1 Structure of the Article

The structure of the article is as follows. We first place our work in its mathematical and deep learning context in Sect. 2. Then, we introduce the needed theoretical preliminaries from Lie group theory in Sect. 3 where we also define the space of positions and orientations \mathbb{M}_d that will allow us to construct roto-translation equivariant networks.

In Sect. 4, we give the overall architecture of a PDE-G-CNN and the ancillary tools that are needed to support a PDE-G-CNN. We propose an equivariant PDE that models commonly used operations in CNNs.

In Sect. 5, we detail how our PDE of interest can be solved using a process called operator splitting. Additionally, we give tangible approximations to the fundamental solutions (kernels) of the PDEs that are both easy to compute and sufficiently accurate for practical applications. We use them extensively in the PDE-G-CNNs GPU-implementations in PyTorch that can be downloaded from the GIT-repository: <https://gitlab.com/bsmetsjr/lietorch>.

Section 6 is dedicated to showing how common CNN operations such as convolution, max-pooling, ReLUs and skip connections can be interpreted in terms of PDEs.

We end our paper with some experiments showing the strength of PDE-G-CNNs in Sect. 7 and concluding remarks in Sect. 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 explicitly in the case of roto-translation equivariant networks acting on \mathbb{M}_d , specifically in 2D (i.e., $d = 2$).

2 Context

As this article touches on disparate fields of study, we use this section to discuss context and highlight some closely related work.

2.1 Drawing Inspiration from PDE-Based Image Analysis

Since the Partial Differential Equations that we use are well known in the context of geometric image analysis [1–11], 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 [32–35] as regularization and our framework provides a nat-

ural way for such flows to be included into neural networks. In the PDE layer from Fig. 2, 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].

2.2 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. 3 and do further processing on the orientation scores [36]. 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. 3.

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 $\mathbb{M}_d \equiv \mathbb{R}^d \times S^{d-1}$, 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 [37–51].

Remark 1 (Generality of the architecture) Although not considered here, for other Lie groups applications (e.g., frequency scores [52,53], velocity scores, scale orientation scores [54]) the same structure applies; therefore, we keep our theory in the general setting of homogeneous spaces G/H . This generality was also important in non-PDE-based learning [22], but also for PDE-based learning it is again beneficial.

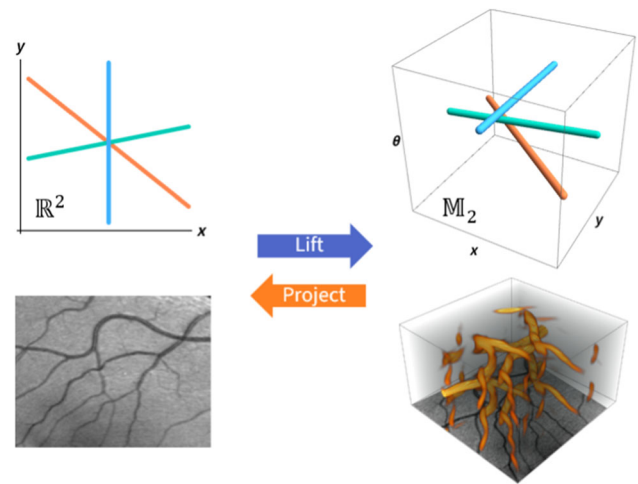


Fig. 3 Illustrating the process of lifting and projecting; in this case the advantage of lifting an image from \mathbb{R}^2 to the 2D space of positions and orientations \mathbb{M}_2 derives from the disentanglement of the lines at the crossings

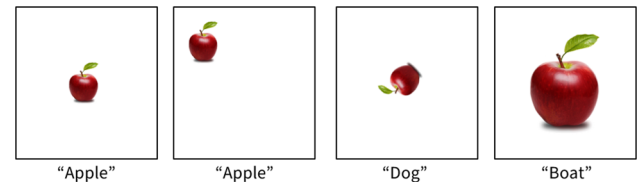


Fig. 4 Spatial CNNs, as used for image classification for example, are translation equivariant but not necessarily equivariant with respect to rotation, scaling and other transformations as the illustrative tags of the differently transformed apples images suggest. Building a G-CNN with the appropriately chosen group confers the network with all the equivariances appropriate for the chosen application. Our PDE-based approach is compatible with the group CNN approach [22] and so can confer the same symmetries

2.3 The Need for Equivariance

We require the layers of our network to be *equivariant*: a transformation of the input should lead to a corresponding transformation of the output, in other words: first transforming the input and then applying the network or first applying the network and then transforming the output should 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. 4. By guaranteeing equivariance of the network, the amount of data necessary or the need for data augmentation is reduced as the required symmetries are intrinsic to the network and need not be trained.

2.4 Related Work

G-CNNs After the introduction of G-CNNs by Cohen and Welling [13] 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 categorized as

- discrete G-CNNs [13–17],
- regular continuous G-CNNs [19–21, 29, 55, 56],
- and steerable continuous G-CNNs [22–25, 57] that rely on Fourier transforms on homogeneous spaces [50, 58].

Both regular and steerable G-CNNs naturally arise from linear mappings between functions on homogeneous spaces that are placed under equivariance constraints [22, 24, 55, 57]. 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. Although steerable operators have clear benefits in terms of computational efficiency and accuracy [59, 60], working with steerable representations puts constraints on nonlinear activations within the networks which limits the representation power of G-CNNs [57]. Like regular G-CNNs, the proposed PDE-G-CNNs do not suffer from this. In our proposed PDE-G-CNN framework, it is essential that we adopt the domain-extension viewpoint, as this allows to naturally and transparently construct scale space PDEs via left-invariant vector fields [12]. In general, this viewpoint entails that the domain of images is extended from the space of positions only, to a higher-dimensional homogeneous space, and originates from coherent state theory [61], orientation score theory [36], cortical perception models [39], G-CNNs [13, 20] and rigid-body motion scattering [62].

The proposed PDE-G-CNNs form a new, unique class of equivariant neural networks, and we show in Sect. 6 how regular continuous G-CNNs arise as a special case of our PDE-G-CNNs.

Probabilistic-CNNs Our geometric PDEs relate to α -stable Lévy processes [50] and cost-processes akin to [26], but then on \mathbb{M}_d rather than \mathbb{R}^d . This relates to probabilistic equivariant numerical neural networks [63] that use anisotropic convection-diffusions on \mathbb{R}^d .

In contrast with these networks, the PDE-G-CNNs that we propose allow for *simultaneous* spatial and angular diffusion on \mathbb{M}_d . Furthermore, we include nonlinear Bellman processes [26] for max pooling over Riemannian balls.

KerCNNs An approach to introducing horizontal connectivity in CNNs that does not require a Lie group structure

was proposed by Montobbio et al. [64, 65] in the form of KerCNNs. In this biologically inspired metric model, a diffusion process is used to achieve intra-layer connectivity.

While our approach does require a Lie group structure, it is not restricted to diffusion and also includes dilation/erosion.

Neural Networks and Differential Equations The connection between neural networks and differential equations became widely known in 2017, when Weinan [66] explicitly explained the connection between neural networks and dynamical systems, especially in the context of the ultradeep ResNet [67]. This point of view was further expanded by Lu et al. [68], 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. [69], 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 also explored by the early works of Chen et al. [70] 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 [71] with their PDE-Net. More recent work in PDE inspired neural networks includes [72, 73].

Basing neural network computations on PDEs formulated on manifolds also makes the processing independent with respect to the choice of coordinates on the manifold in the fashion of Weiler et al. [74].

More recent work in this direction includes integrating equivariant partial differential operators in steerable CNNs [75], drawing a strong analogy between deep learning and physics.

A useful aspect of the connection between neural networks and differential equations is the observation that the stability of the differential equation can give into the stability and generalization ability of the neural network [76]. Moreover, there are intriguing analogies with numerical PDE-approximations and specific network architectures (e.g. ResNets), as can be seen in the comprehensive overview article by Alt et al. [77].

The main contribution of our work in the field of PDE-related neural networks is that we implement and analyze geometric PDEs on homogeneous spaces, to obtain general *group equivariant PDE-based CNNs* whose implementations just require linear and morphological convolutions with new analytic approximations of scale space kernels.

3 Equivariance: Groups and 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.

3.1 The General Case

A layer in a neural network (or indeed the whole network) can be viewed as an operator from a space of real-valued functions defined on a space X to a space of real-valued 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 *connected Lie group* G . Think for instance of a group of translations which shift the domain into different directions. The Lie group being connected excludes transformations such as reflections, which we want to avoid for the sake of simplicity. We further assume that the Lie group G acts smoothly on both spaces 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 $g, h \in G$,

$$\rho_X(gh, x) = \rho_X(g, \rho_X(h, x))$$

and

$$\rho_Y(gh, x) = \rho_Y(g, \rho_Y(h, x)),$$

making ρ_X and ρ_Y group actions on their respective spaces.

Additionally we will assume that the group G acts *transitively* on the spaces, meaning that for any two elements of these spaces 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* [78]. In particular, this means that after selecting a reference element $x_0 \in X$ we can make the following isomorphism:

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

using the mapping

$$x \mapsto \{g \in G \mid \rho_X(g, x_0) = x\}, \quad (2)$$

which is a bijection due to transitivity and the fact that

$$\text{Stab}_G(x_0) := \{g \in G \mid \rho_X(g, x_0) = x_0\}$$

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

In this article, we will restrict ourselves to those homogeneous spaces that correspond to those quotients G/H where the subgroup H is compact and connected. Restricting ourselves to compact and connected subgroups simplifies many constructions and still covers several interesting cases such as the rigid body motion groups $SE(d)$.

The elements of the quotient G/H consist of subsets of G which we will 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

$$g_1 \sim g_2 \iff g_1^{-1}g_2 \in H. \iff g_1H = g_2H.$$

Under this notation, the group G consists of the disjoint union

$$G = \bigsqcup_{p \in G/H} p. \quad (3)$$

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 $p_0 := H$ when we want to think of it as an atomic entity rather than a set in its own right.

We will denote quotient map from G to G/H with π :

$$\pi(g) := gp_0 := gH. \quad (4)$$

Remark 2 (Principal homogeneous space) Observe that by choosing $H = \{e\}$ we get $G/H \cong G$, 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 numerical experiments we perform in this paper are on the principal homogeneous space $\mathbb{R}^2 \times S^1$ of $SE(2)$.

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

$$L_g p := gp \quad \text{for all } p \in G/H. \quad (5)$$

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

$$(\mathcal{L}_g f)(p) := f(g^{-1}p). \quad (6)$$

A neural network layer is itself an operator (from functions on G/H_X to functions on G/H_Y), and we require the operator

to be equivariant with respect to the actions on these function spaces.

Definition 1 (*Equivariance*) Let G be a Lie group with homogeneous spaces G/H_X and G/H_Y . Let Φ be an operator from functions (of some function class) on G/H_X to functions on G/H_Y , then we say that Φ is equivariant with respect to G if for all functions f (of that class) we have that:

$$\forall g \in G : (\Phi \circ \mathcal{L}_g) f = (\mathcal{L}_g \circ \Phi) f, \quad (7)$$

or in words: the neural network commutes with transformations.

Most of the time we will have $H_X = H_Y$ in our proposed neural networks; only the initial lifting layer and the final projection layer will be between different homogeneous spaces, as we will see later on.

3.2 Vector and Metric Tensor Fields

The particular operators that we will base our framework on are vector and tensor fields, if these basic building blocks are equivariant then our processing will be equivariant. We explain what left invariance means for these objects next.

For $g \in G$ and $p \in G/H$, let $T_p(G/H)$ be the tangent space at point p , then the pushforward

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

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

$$((L_g)_* \mathbf{v}) f := \mathbf{v}(f \circ L_g). \quad (8)$$

Remark 3 (Tangent vectors as differential operators) Other than 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 (8). This algebraic viewpoint defines the action of the tangent vector $\dot{\gamma}(t)$ on a differentiable function f as

$$\dot{\gamma}(t)f := \frac{\partial}{\partial s} f(\gamma(s)) \Big|_{s=t}.$$

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 [79, §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

$$\forall g \in G, \forall p \in G/H : \mathbf{v}(p)f = \mathbf{v}(gp)[\mathcal{L}_g f], \quad (9)$$

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

Definition 2 (*G-invariant vector field on a homogeneous space*) A vector field \mathbf{v} on G/H is invariant with respect to G if it satisfies

$$\forall g \in G, \forall p \in G/H : \mathbf{v}(gp) = (L_g)_* \mathbf{v}(p). \quad (10)$$

It is straightforward to check that (9) and (10) are equivalent and that these imply the following.

Corollary 3 (Properties of G -invariant vector fields) *On a homogeneous space G/H , a G -invariant vector field \mathbf{v} has the following properties:*

1. *it is fully determined by its value $\mathbf{v}|_H \in T_H(G/H)$ in H ,*
2. *$\forall h \in H : (L_h)_* \mathbf{v}|_H = \mathbf{v}|_H$.*

We also introduce G -invariant metric tensor fields.

Definition 4 (*G-invariant metric tensor field on G/H*) A $(0, 2)$ -tensor field \mathcal{G} on G/H is G -invariant if and only if

$$\forall g \in G, \forall p \in G/H, \forall \mathbf{v}, \mathbf{w} \in T_p(G/H) : \mathcal{G}|_p(\mathbf{v}, \mathbf{w}) = \mathcal{G}|_{gp}((L_g)_* \mathbf{v}, (L_g)_* \mathbf{w}). \quad (11)$$

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

Corollary 5 (Properties of G -invariant metric tensor fields) *On a homogeneous space G/H , a G -invariant metric tensor field \mathcal{G} has the following properties:*

1. *it is fully determined by its metric tensor $\mathcal{G}|_{p_0}$ at $p_0 = H$,*
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 (tensor) has to be symmetric with respect to the subgroup H .

A (positive definite) metric tensor field yields a Riemannian metric in the usual manner, as we recall next.

Definition 6 (Metric on G/H) Let $p_1, p_2 \in G/H$ then:

$$d_G(p_1, p_2) := d_{G/H, \mathcal{G}}(p_1, p_2) := \inf_{\substack{\beta \in \text{Lip}([0,1], G/H) \\ \beta(0)=p_1, \beta(1)=p_2}} \int_0^1 \sqrt{\mathcal{G}|_{\beta(t)}(\dot{\beta}(t), \dot{\beta}(t))} dt.$$

As metrics and their smoothness play a role in our construction, we need to take into account where that smoothness fails.

Definition 7 The cut locus $\text{cut}(p) \subset G/H$ or $\text{cut}(g) \subset G$ is the set of points, respectively, group elements where the distance map from p resp. g is not smooth (excluding the point p and group element g themselves).

As long as we stay away from the cut locus, the infimum from Definition 6 gives a unique geodesic.

Being derived from a G -invariant tensor field gives the metric d_G the same symmetries.

Proposition 8 (G -invariance of the metric on G/H) Let $p_1, p_2 \in G/H$ away from each other's cut locus, then we have:

$$\forall g \in G : d_G(p_1, p_2) = d_G(gp_1, gp_2).$$

Proof We observe that we can make a bijection from the set of Lipschitz curves between p_1 and p_2 and between gp_1 and gp_2 simply by left multiplication by g one way and g^{-1} the other way. Due to (11), multiplying a curve with a group element preserves its length; hence, if $\gamma : [0, 1] \rightarrow G/H$ is the geodesic from p_1 to p_2 , then $g\gamma$ is the geodesic from gp_1 to gp_2 , both having the same length. \square

A metric tensor field on the homogeneous space has a natural counterpart on the group.

Definition 9 (Pseudometric tensor field on G) A G -invariant metric tensor field \mathcal{G} on G/H induces a (pullback) pseudometric tensor field $\tilde{\mathcal{G}}$ on G that is left invariant:

$$\tilde{\mathcal{G}} := \pi^* \mathcal{G}, \quad (12)$$

where π^* is the pullback of the quotient map π from (4). This is equivalent to saying that for all $\mathbf{v}, \mathbf{w} \in T_g G$:

$$\tilde{\mathcal{G}}|_g(\mathbf{v}, \mathbf{w}) := \mathcal{G}|_{\pi(g)}(\pi_* \mathbf{v}, \pi_* \mathbf{w}),$$

where π_* is the pushforward of π .

This tensor field $\tilde{\mathcal{G}}$ is left invariant by virtue of \mathcal{G} being G -invariant. It is also degenerate in the direction of H and so yields a seminorm on TG .

Definition 10 (Seminorm on TG) Let $\mathbf{v} \in T_g G$. Then, the metric tensor field \mathcal{G} on G/H induces the following seminorm:

$$\|\mathbf{v}\|_{\tilde{\mathcal{G}}} := \sqrt{\tilde{\mathcal{G}}|_g(\mathbf{v}, \mathbf{v})} := \sqrt{\mathcal{G}|_{gp_0}(\pi_* \mathbf{v}, \pi_* \mathbf{v})}. \quad (13)$$

In the same fashion, we have an induced pseudometric on G from the pseudometric tensor field on G .

Definition 11 (Pseudometric on G) Let $g_1, g_2 \in G$. Then, we define:

$$d_{\tilde{\mathcal{G}}}(g_1, g_2) := d_{G, \tilde{\mathcal{G}}}(g_1, g_2) := \inf_{\substack{\gamma \in \text{Lip}([0,1], G) \\ \gamma(0)=g_1, \gamma(1)=g_2}} \int_0^1 \sqrt{\tilde{\mathcal{G}}|_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt. \quad (14)$$

This pseudometric has the property that $d_{\tilde{\mathcal{G}}}(h_1, h_2) = 0$ for all $h_1, h_2 \in H$; in fact for all $p \in G/H$ we have that $d_{\tilde{\mathcal{G}}}(g_1, g_2) = 0$ for all $g_1, g_2 \in p$.

By requiring G and H to be connected, we get the following strong correspondence between the metric structure on the homogeneous space and the pseudometric structure on the group.

Lemma 12 Let $g_1, g_2 \in G$ so that $\pi(g_2)$ is away from the cut locus of $\pi(g_1)$, then:

$$d_{\tilde{\mathcal{G}}}(g_1, g_2) = d_G(\pi(g_1), \pi(g_2)).$$

Moreover, if γ is a minimizing geodesic in the group G connecting g_1 with g_2 , then $\pi \circ \gamma$ is the unique minimizing geodesic in the homogeneous space G/H that connects $\pi(g_1)$ with $\pi(g_2)$.

Proof Assuming it exists, let $\gamma \in \text{Lip}([0, 1], G)$ be a minimizing geodesic connecting $\gamma(0) = g_1$ with $\gamma(1) = g_2$ and let $\beta \in \text{Lip}([0, 1], G/H)$ be the unique minimizing geodesic connecting $\beta(0) = \pi(g_1)$ with $\beta(1) = \pi(g_2)$. Because of the pseudometric on G , minimizing geodesics are not unique, i.e. γ is not unique. On G/H , we have a full metric and so staying away from the cut locus means β is both unique and minimizing.

Denote the length functionals with:

$$\text{Len}_G(\gamma) := \int_0^1 \sqrt{\tilde{\mathcal{G}}|_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt,$$

$$\text{Len}_{G/H}(\beta) := \int_0^1 \sqrt{\mathcal{G}|_{\beta(t)}(\dot{\beta}(t), \dot{\beta}(t))} dt.$$

Observe that by construction of the pseudometric tensor field $\tilde{\mathcal{G}}$ on G , we have: $\text{Len}_G(\gamma) = \text{Len}_{G/H}(\pi \circ \gamma)$.

Now we assume $\pi \circ \gamma \neq \beta$. Then, since β is the unique geodesic, we have

$$\text{Len}_{G/H}(\beta) < \text{Len}_{G/H}(\pi \circ \gamma) = \text{Len}_G(\gamma).$$

But then we can find some $\gamma_{\text{lift}} \in \text{Lip}([0, 1], G)$ that is a preimage of β , i.e., $\pi \circ \gamma_{\text{lift}} = \beta$. The potential problem is that while $\gamma_{\text{lift}}(0) \in \pi(g_1)$ and $\gamma_{\text{lift}}(1) \in \pi(g_2)$, γ_{lift} does not necessarily connect g_1 to g_2 . But since the coset $\pi(g_1)$ is connected, we can find a curve wholly contained in it that connects g_1 with $\gamma_{\text{lift}}(0)$, call this curve $\gamma_{\text{head}} \in \text{Lip}([0, 1], \pi(g_1))$. Similarly we can find a $\gamma_{\text{tail}} \in \text{Lip}([0, 1], \pi(g_2))$ that connects $\gamma_{\text{lift}}(1)$ to g_2 . Both these curves have zero length since π maps them to a single point on G/H , i.e., $\text{Len}_G(\gamma_{\text{head}}) = \text{Len}_G(\gamma_{\text{tail}}) = 0$.

Now we can compose these three curves:

$$\gamma_{\text{new}}(t) := \begin{cases} \gamma_{\text{head}}(3t) & \text{if } t \in [0, 1/3], \\ \gamma_{\text{lift}}(3t - 1) & \text{if } t \in [1/3, 2/3], \\ \gamma_{\text{tail}}(3t - 2) & \text{if } t \in [2/3, 1]. \end{cases}$$

This new curve is again in $\text{Lip}([0, 1], G)$ and connects g_1 with g_2 , but also:

$$\text{Len}_G(\gamma_{\text{new}}) = \text{Len}_G(\gamma_{\text{lift}}) = \text{Len}_{G/H}(\beta) < \text{Len}_G(\gamma),$$

which is a contradiction since γ is a minimizing geodesic between g_1 and g_2 . We conclude $\pi \circ \gamma = \beta$ and thereby:

$$d_{\tilde{G}}(g_1, g_2) = \text{Len}_G(\gamma) = \text{Len}_{G/H}(\beta) = d_{\tilde{G}}(\pi(g_1), \pi(g_2)).$$

□

This result allows us to more easily translate results from Lie groups to homogeneous spaces.

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

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

Our main example and Lie group of interest are 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

$$\mathbb{M}_d = SE(d) / (\{0\} \times SO(d-1)). \quad (15)$$

This homogeneous space and group will enable the construction of roto-translation equivariant networks. For the experiments in this paper, we restrict ourselves to $d = 2$, but we include the case $d = 3$ in some of our theoretical preliminaries.

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, 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

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

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

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

What the G -invariant vector field and metric tensor fields look like on \mathbb{M}_d is different for $d = 2$ than for $d > 2$. We first look at $d > 2$.

Proposition 13 *Let $d > 2$ and let $\partial_{\mathbf{a}} \in T_{p_0}(\mathbb{M}_d)$ be the tangent vector in the reference point in the main direction $\mathbf{a} \in S^{d-1}$, specifically:*

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

where $f : \mathbb{M}_d \rightarrow \mathbb{R}$ is smooth in an open neighborhood of $p_0 = (\mathbf{0}, \mathbf{a})$; then, all $SE(d)$ -invariant vector fields are spanned by the vector field:

$$p \mapsto \mathcal{A}_1|_p := (L_{g_p})_* \partial_{\mathbf{a}}, \quad (17)$$

with $g_p \in p \in \mathbb{M}_d$.

Proof For $d > 3$, we can see that (17) are the only left-invariant vector fields since for all $h \in H$ we have $(g_p h) p_0 = p$, 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 $T_{p_0}(\mathbb{M}_d)$. □

Proposition 14 *For $d > 2$, the only Riemannian metric tensor fields on \mathbb{M}_d that are $SE(d)$ -invariant are of the form:*

$$\mathcal{G}|_{(\mathbf{x}, \mathbf{n})}((\dot{\mathbf{x}}, \dot{\mathbf{n}}), (\dot{\mathbf{x}}, \dot{\mathbf{n}})) \\ = w_M |\dot{\mathbf{x}} - \mathbf{n}|^2 + w_L \|\dot{\mathbf{x}} \wedge \mathbf{n}\|^2 + w_A \|\dot{\mathbf{n}}\|^2, \quad (18)$$

with $w_M, w_L, w_A > 0$ 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 the second condition of Corollary 5 at the tangent space $T_{(x,n)}$ of a particular (x, n) the metric tensor needs to be symmetric with respect to rotations about n both spatially and angularly (i.e., we require isotropy in all angular and lateral directions) which leads to the three degrees of freedom contained in (18) irrespective of d . \square

For $d = 2$, we represent the 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, so that $n = (\cos \theta, \sin \theta)^T$. The reference element is then simply denoted by $(0, 0, 0)$.

It may be counter-intuitive but decreasing the number of dimensions to 2 gives more freedom to the G -invariant vector and metric tensor fields compared to $d > 2$. This is a consequence of the subgroup H being trivial, and so the symmetry conditions from Corollaries 3 and 5 also become trivial. The $SE(2)$ -invariant vector fields are given as follows.

Proposition 15 *On \mathbb{M}_2 , the $SE(2)$ -invariant vector fields are spanned by the following basis:*

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

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

In a similar manner, $SE(2)$ -invariant metric tensors are then given as follows.

Proposition 16 *On \mathbb{M}_2 , the $SE(2)$ -invariant metric tensor fields are given by:*

$$\mathcal{G}|_{(x,y,\theta)}(v, w) = \mathcal{G}|_{(0,0,0)}\left(\left(L_{(x,y,\theta)}^{-1}\right)_* v, \left(L_{(x,y,\theta)}^{-1}\right)_* w\right),$$

for any choice of inner product $\mathcal{G}|_{(0,0,0)}$ at e .

Proof Since $SE(2) \equiv \mathbb{M}_2$, the G -invariant metric tensor fields are again exactly the left-invariant metric tensor fields. \square

This gives $SE(2)$ -invariant metric tensor fields 6 degrees of freedom and hence 6 trainable parameters on \mathbb{M}_2 . Remarkably, the case $d = 2$ allows for more degrees of freedom than the case $d = 3$ where Proposition 14 applies. In our experiments so far, we have restricted ourselves to those metric tensors that are diagonal with respect to the frame from Proposition 15. A diagonal metric tensor would have just 3 degrees of freedom and have the same general form as (18), specifically:

$$\begin{aligned} \mathcal{G}|_{(x,y,\theta)}((\dot{x}, \dot{y}, \dot{\theta}), (\dot{x}, \dot{y}, \dot{\theta})) \\ = w_M |\dot{x} \cos \theta + \dot{y} \sin \theta|^2 \\ + w_L |-\dot{x} \sin \theta + \dot{y} \cos \theta|^2 \\ + w_A |\dot{\theta}|^2. \end{aligned} \quad (20)$$

We will expand into non-diagonal metric tensors in future work.

4 Architecture

4.1 Lifting and Projecting

The key ingredient 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. 5.

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 live 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 $SE(2)$ is found in [20,80].

Remark 4 (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 G/H_1 and G/H_2 of a Lie group G ; let $f : G/H_1 \rightarrow \mathbb{R}$ and $k : G/H_2 \rightarrow \mathbb{R}$ with the following property:

$$\forall h \in H_1, q \in G/H_2 : k(hq) = k(q),$$

where H_1 is compact. Then, the operator \mathcal{T} defined by

$$\forall q \in G/H_2 : (\mathcal{T}f)(q) := \int_G k(g^{-1}q) f(gH_1) d\mu_G(g) \quad (21)$$

transforms f from a function on G/H_1 to a function on G/H_2 in an equivariant manner (assuming f and k are such that the

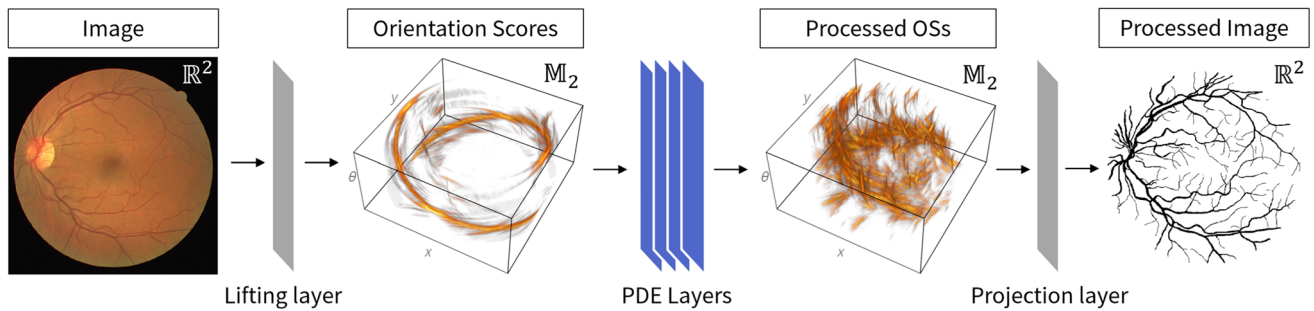


Fig. 5 Illustrating the overall architecture of a PDE-G-CNN (example: retinal vessel segmentation). An input image is lifted to a homogeneous space from which point on it can be fed through subsequent PDE layers

(each PDE layer follow the structure of Fig. 2) that replace the convolution layers in conventional CNNs. Finally, the result is projected back to the desired output space

integral exists). Here the kernel k is the trainable part and μ_G is the left-invariant Haar measure on the group.

Moreover, it can be shown via the Dunford–Pettis [81] theorem that (under mild restrictions) all linear transforms between homogeneous spaces are of this form.

Remark 5 (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 *Invertible Orientation Score Transform* [36] or a trainable lift [20] in the style of Remark 4.

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 [20]. A variation on the above projection is detailed in [80, Ch. 3.3.3].

4.2 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 $T > 0$. Afterward, we take these solutions at time T and take affine combinations (really batch normalized linear combinations in practice) 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 the PDE solvers that each apply the PDE evolution to their respective input channel and the affine combination unit. This design is illustrated in Fig. 1, but let us formalize it.

Let $(U_{\ell,c})_{c=1}^{M_\ell}$ be the inputs of the ℓ -th layer (i.e., some functions on G/H); let $a_{\ell ij}$ and $b_{\ell i} \in \mathbb{R}$ be the coefficients of the affine transforms for $i = 1 \dots M_{\ell+1}$ and $j = 1 \dots M_\ell$. Let each PDE be parametrized by a set of parameters $\theta_{\ell j}$. Then, the action of a PDE layer is described as:

$$U_{\ell+1,i} = \sum_{j=1}^{M_\ell} a_{\ell ij} \Phi_{T, \theta_{\ell j}}(U_{\ell j}) + b_{\ell i}, \quad (23)$$

where $\Phi_{T, \theta}$ is the evolution operator of the PDE at time $T \geq 0$ and parameter set θ . We define the operator $\Phi_{t, \theta}$ so that $(p, t) \mapsto (\Phi_{t, \theta} U)(p)$ satisfies the Hamilton–Jacobi type PDE that we introduce in just a moment. In this layer formula, the parameters $a_{\ell ij}$, $b_{\ell i}$ and $\theta_{\ell j}$ are the trainable weights, but the evolution time T we keep fixed.

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

The PDE solver that we will consider in this article, illustrated in Fig. 2, computes the approximate solution to the PDE

$$\left\{ \begin{array}{ll} \frac{\partial W}{\partial t}(p, t) = \begin{array}{l} -cW(p, t) \\ -(-\Delta_{\mathcal{G}_1})^\alpha W(p, t) \\ + \|\nabla_{\mathcal{G}_2^+} W(p, t)\|_{\mathcal{G}_2^+}^{2\alpha} \\ - \|\nabla_{\mathcal{G}_2^-} W(p, t)\|_{\mathcal{G}_2^-}^{2\alpha} \end{array} & \begin{array}{l} \text{(convection)} \\ \text{(diffusion)} \\ \text{(dilation)} \\ \text{(erosion)} \end{array} \\ W(p, 0) = U(p) & \begin{array}{l} \text{for } p \in G/H, t \geq 0, \\ \text{for } p \in G/H. \end{array} \end{array} \right. \quad (24)$$

Here, c is a G -invariant vector field on G/H (recall (17) and our use of tangent vectors as differential operators per Remark 3), $\alpha \in [1/2, 1]$, \mathcal{G}_1 and \mathcal{G}_2^\pm are G -invariant metric tensor fields on G/H , 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 four terms have distinct effects:

- convection: moving data around,
- (fractional) diffusion: regularizing data (which relates to subsampling by destroying data),
- dilation: pooling of data,
- erosion: sharpening of data.

This is also why we refer to a layer using this PDE as a CDDE layer. Summarized parameters of this PDE are given by $\theta = (\mathbf{c}, \mathcal{G}_1, \mathcal{G}_2^+, \mathcal{G}_2^-)$. The geometric interpretation of each of the terms in (24) is illustrated in Fig. 6 for $G = \mathbb{R}^2$ and in Fig. 7 for $G = \mathbb{M}_2$.

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

4.3 Training

Training the PDE layer comes down to adapting the parameters in the PDEs 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 G invariant, the parameter space is finite-dimensional as a consequence of Corollaries 3 and 5 if we restrict ourselves to Riemannian metric tensor fields.

For our main application on \mathbb{M}_2 , each PDE unit would have the following 12 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 each to specify the dilation and erosion metric tensor fields \mathcal{G}_2^\pm ,

where the metric tensor fields are of the form (20) that are diagonal with respect to the frame from Proposition 15.

Surprisingly for higher dimensions, \mathbb{M}_d has less trainable parameters than for $d = 2$. This is caused by the $SE(d)$ -invariant vector fields on \mathbb{M}_d for $d \geq 3$ being spanned by a single basis element (per Proposition 13) 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 $d \geq 3$.

In our own experiments, we always use some form of stochastic gradient descent (usually ADAM) with a small

amount of L^2 regularization applied uniformly over all the parameters. Similarly, we stick to a single learning rate for all the parameters. Given that in our setting different parameters have distinct effects, treating all of them the same is likely far from optimal, however, we leave that topic for future investigation.

5 PDE Solver

Our PDE solver will consist of an iteration of time step units, each of which is a composition of convection, diffusion, dilation and erosion substeps. 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 convection, diffusion and dilation/erosion steps are implemented with, respectively, a shifted resample, linear convolution and two morphological convolutions, as illustrated in Fig. 8.

The composition of the substeps does not solve (24) 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

$$\begin{cases} \frac{\partial W_1}{\partial t}(p, t) = -\mathbf{c}(p)W_1(\cdot, t) & \text{for } p \in G/H, t \geq 0, \\ W_1(p, 0) = U_1(p) & \text{for } p \in G/H. \end{cases} \quad (25)$$

The output of the layer is the solution of the PDE evaluated at time $t = T$, i.e., the output is the function $p \mapsto W_1(p, T)$.

Proposition 17 (Convection solution) *The solution of the convection PDE (25) is found by the method of characteristics and is given by*

$$\begin{aligned} W^1(p, t) &= (\mathcal{L}_{g_p^{-1}} U_1) (\gamma_{\mathbf{c}}(t)^{-1} p_0) \\ &= U_1 (g_p \gamma_{\mathbf{c}}(t)^{-1} p_0) \end{aligned} \quad (26)$$

$$= U_1 (g_p \gamma_{-\mathbf{c}}(t) p_0), \quad (27)$$

where $g_p \in p$ (i.e., $g_p p_0 = p$) and $\gamma_{\mathbf{c}} : \mathbb{R} \rightarrow G$ is the exponential curve that satisfies $\gamma_{\mathbf{c}}(0) = e$ and

$$\frac{\partial}{\partial t} (\gamma_{\mathbf{c}}(t) p) (t) = \mathbf{c}(\gamma_{\mathbf{c}}(t) p), \quad (28)$$

i.e., $\gamma_{\mathbf{c}}$ is the exponential curve in the group G that induces the integral curves of the G -invariant vector field \mathbf{c} on G/H when acting on elements of the homogeneous space.






\mathbb{R}^2	Transport	Regularization	Max pooling	Min pooling
PDE Term:	$-cW$	$-(-\Delta_{g_1})^\alpha W$	$+\ \nabla_{g_2^+} W\ _{g_2^+}^{2\alpha}$	$-\ \nabla_{g_2^-} W\ _{g_2^-}^{2\alpha}$
Parameters:	transport vector	Riemannian metric tensor	Riemannian metric tensor	Riemannian metric tensor
	$\begin{pmatrix} 100 \\ 120 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 5 & 11 \\ 11 & 40 \end{pmatrix}$
Operation:	resample with offset	convolution with kernel	dilation with kernel	erosion with kernel
				
				

Fig. 6 Geometric interpretation of the terms of the PDE (24) illustrated for \mathbb{R}^2 . In this setting, the G -invariant vector field c is the constant vector field given by two translation parameters. For the other terms, we use Riemannian metric tensors parametrized by a positive definite 2×2

matrix in the standard basis. The kernels used in the diffusion, dilation and erosion terms are functions of the distance map induced by the metric tensors

Note that this exponential curve existing is a consequence of the vector field c being G -invariant, such exponential curves do not exist for general convection vector fields.

Proof

$$\begin{aligned}
 & \frac{\partial W_1}{\partial t}(p, t) \\
 &= \lim_{h \rightarrow 0} \frac{W_1(p, t+h) - W_1(p, t)}{h} \\
 &= \lim_{h \rightarrow 0} \frac{U_1(g_p \gamma_c(t+h)^{-1} p_0) - U_1(g_p \gamma_c(t)^{-1} p_0)}{h} \\
 &= \lim_{h \rightarrow 0} \frac{U_1(g_p \gamma_c(t)^{-1} \gamma_c(h)^{-1} p_0) - U_1(g_p \gamma_c(t)^{-1} p_0)}{h},
 \end{aligned}$$

now let $\bar{U} := \mathcal{L}_{\gamma_c(t) g_p^{-1}} U_1$, then

$$\begin{aligned}
 &= \lim_{h \rightarrow 0} \frac{\bar{U}(\gamma_c(h)^{-1} p_0) - \bar{U}(p_0)}{h} \\
 &= -c(p_0) \bar{U} \\
 &= -(L_{g_p})_* c(p_0) \mathcal{L}_{g_p} \bar{U}
 \end{aligned}$$

due to the G -invariance of c this yields

$$\begin{aligned}
 &= -c(p) \mathcal{L}_{g_p} \mathcal{L}_{\gamma_c(t) g_p^{-1}} U_1 \\
 &= -c(p) \left[p \mapsto U_1(g_p \gamma_c(t)^{-1} g_p^{-1} p) \right] \\
 &= -c(p) \left[p \mapsto U_1(g_p \gamma_c(t)^{-1} p_0) \right] \\
 &= -c(p) W_1(\cdot, t).
 \end{aligned}$$

□

In our experiments, Eq. (27) is numerically implemented as a resampling operation with trilinear interpolation to account for the off-grid coordinates.

5.2 Fractional Diffusion

The (fractional) diffusion step solves the PDE

$$\begin{cases} \frac{\partial W_2}{\partial t} = -(-\Delta_{g_1})^\alpha W_2(p, t) & \text{for } p \in G/H, t \geq 0, \\ W_2(p, 0) = U_2(p) & \text{for } p \in G/H. \end{cases} \quad (29)$$

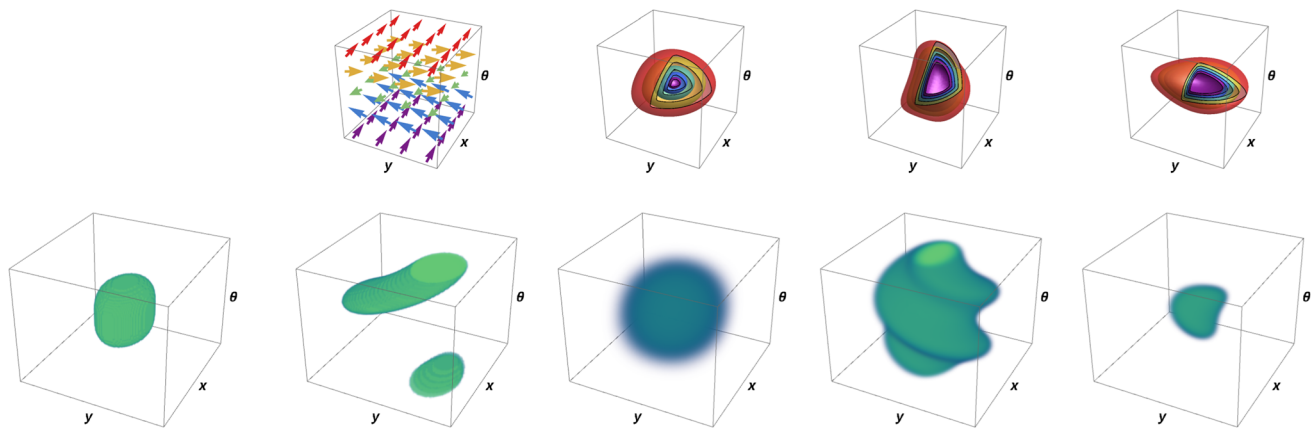
\mathbb{M}_2	Transport	Regularization	Max pooling	Min pooling
PDE Term:	$-cW$	$-(\Delta_{\mathcal{G}_1})^\alpha W$	$+\ \nabla_{\mathcal{G}_2^+} W\ _{\mathcal{G}_2^+}^{2\alpha}$	$-\ \nabla_{\mathcal{G}_2^-} W\ _{\mathcal{G}_2^-}^{2\alpha}$
Parameters:	G-invariant vector field $\begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$	Riemannian metric tensor $\begin{pmatrix} 4 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	Riemannian metric tensor $\begin{pmatrix} 4 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}$	Riemannian metric tensor $\begin{pmatrix} 6 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1.5 \end{pmatrix}$
Operation:	resample with offset	convolution with kernel	dilation with kernel	erosion with kernel
				

Fig. 7 Geometric interpretation of the terms of the PDE (24) illustrated for \mathbb{M}_2 . In this setting, the G -invariant vector field c is a left-invariant vector field given by two translation and one rotation parameter. For the other terms, we use Riemannian metric tensors parametrized by a positive definite 3×3 matrix in the left-invariant basis (the matrix does not

need to be diagonal, but we keep that for future work). The kernels used in the diffusion, dilation and erosion terms are functions of the distance map induced by the metric tensors and are visualized by partial plots of their level sets

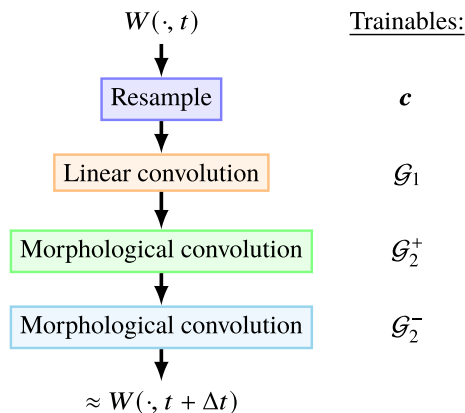


Fig. 8 Evolving the PDE through operator splitting, each operation corresponds to a term of (24)

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

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

called the fundamental solution of the α -diffusion equation, such that for every initial condition U_2 ; the solution to the PDE (29) is given by the convolution of the function U_2 with the fundamental solution K_t^α :

$$W^2(p, t) = (K_t^\alpha *_{G/H} U_2)(p). \quad (30)$$

The convolution $*_{G/H}$ on a homogeneous space G/H is specified by the following definition.

Definition 18 (Linear group convolution) Let $p_0 = H$ be compact, let $f \in L^2(G/H)$ and $k \in L^1(G/H)$ such that:

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

then, we define:

$$(k *_{G/H} f)(p) := \frac{1}{\mu_H(H)} \int_G k(g^{-1}p) f(gp_0) d\mu_G(g), \quad (31)$$

where μ_H and μ_G are the left-invariant Haar measures (determined up to scalar multiplication) on H , respectively, G .

Remark 6 In the remainder of this article, we refer to the left-invariant Haar measure on G as ‘the Haar measure on G ’ as right-invariant Haar measures on G do not play a role in our framework.

Remark 7 Compactness of H is crucial as otherwise the integral in the right-hand side of (31) does not converge. To this end, we note that one can always decompose (by Weil’s integral formula [82, Lem. 2.1]) the Haar measure μ_G on the group as a product of a measure on the quotient G/H times the measure on the subgroup H . As Haar measures are determined up to a constant, we take the following convention: we normalize the Haar measure μ_G such that

$$\mu_G(\pi^{-1}(A)) = \mu_H(H) \mu_{\mathcal{G}}(A), \quad \forall A \subset G/H, \quad (32)$$

where $\mu_{\mathcal{G}}$ is the Riemannian measure induced by \mathcal{G} and μ_H is a choice of Haar measure on H . Thereby, (32) boils down to Weil’s integration formula:

$$\mu_H(H) \int_{G/H} f(p) d\mu_{\mathcal{G}}(p) = \int_G f(gH) d\mu_G(g) \quad (33)$$

whenever f is measurable. Since H is compact, we can indeed normalize the Haar measure μ_H so that $\mu_H(H) = 1$.

In general, an exact analytic expression for the fundamental solution K_t^α requires complicated steerable filter operators [50, Thm. 1 & 2], and for that reason we contend ourselves with more easily computable approximations. For now, let us construct our approximations and address their quality and the involved asymptotics later.

Remark 8 In the approximations, we will make use of logarithmic map as the inverse of the Lie group exponential map \exp_G . Locally, such inversion can always be done by the inverse function theorem. Specifically, there is always a neighborhood $V \subset T_e G$ of the origin so that $\exp_G|_V$ is a diffeomorphism between V and $W = \exp_G(V) \subset G$, where W is a neighborhood of e . Then we define the logarithmic map $\log_G : W \rightarrow V$ by $\exp_G \circ \log_G = \text{id}_W$ and $\log_G \circ \exp_G|_V = \text{id}_V$. For the moment, for simplicity, we assume $V = T_e(G)$ in the general setting.¹

The idea is that instead of basing our kernels on the metric $d_{\mathcal{G}}$ (which is hard to calculate [83]) we approximate it using the seminorm from Definition 10 (which is easy to calculate). We can use this seminorm on elements of the homogeneous space by using the group’s logarithmic map \log_G . We can take

¹ In our primary case of interest $G = SE(2)$, we have $V = \{\sum_{k=1}^3 c^k A_k \mid c^3 \in [-\pi, \pi)\}$.

the group logarithm of all the group elements that constitute a particular equivalence class of G/H and then pick the group element with the lowest seminorm:

$$d_{\mathcal{G}}(p_0, p) \approx \inf_{g \in p} \|\log_G g\|_{\tilde{\mathcal{G}}}. \quad (34)$$

Henceforth, we write this estimate as $d_{\mathcal{G}}(p_0, p) \approx \rho_{\mathcal{G}}(p)$ relying on the following definition.

Definition 19 (Logarithmic metric estimate) Let \mathcal{G} be a G -invariant metric tensor field on the homogeneous space G/H , then we define

$$\begin{aligned} \rho_{\mathcal{G}}(p) &:= \inf_{g \in p} \|\log_G g\|_{\tilde{\mathcal{G}}} \\ &:= \inf_{g \in p} \sqrt{\mathcal{G}(\pi_* \log_G g, \pi_* \log_G g)}, \end{aligned} \quad (35)$$

where π_* is the push-forward of the projection map π given by (4).

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 constant velocity according to the seminorm in Definition 10 and use that velocity as the distance estimate.

Summarizing, Definition 19 and Eq. (34), can be intuitively reformulated as: ‘instead of the length of the geodesic connecting two points of G/H , we take the length of the shortest exponential curve connecting those two points.’

The following lemma quantifies how well our estimate approximates the true metric.

Lemma 20 (Bounding the logarithmic metric estimate) For all $p \in G/H$ sufficiently close to p_0 , we have

$$d_{\mathcal{G}}(p_0, p)^2 \leq \rho_{\mathcal{G}}(p)^2 \leq d_{\mathcal{G}}(p_0, p)^2 + O(d_{\mathcal{G}}(p_0, p)^4),$$

which has as a weaker corollary that for all compact neighborhoods of p_0 there exists a $C_{\text{metr}} > 1$ so that

$$d_{\mathcal{G}}(p_0, p) \leq \rho_{\mathcal{G}}(p) \leq C_{\text{metr}} d_{\mathcal{G}}(p_0, p)$$

for all p in that neighborhood. Note that the constant C_{metr} depends on both the choice of compact neighborhood and the metric tensor field.

The proof of this lemma can be found in Appendix A.1.

Remark 9 (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

$$\rho_{\mathcal{G}}(g) = \|\log_G g\|_{\mathcal{G}|_e};$$

hence, we see that this construction generalizes the logarithmic estimate, as used in [84,85], to homogeneous spaces other than the principal.

Remark 10 (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 (20), we formulate the metric estimate in terms of the following auxiliary functions called the exponential coordinates of the first kind:

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

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

$$\rho_G(x, y, \theta) = \sqrt{w_M c^1(x, y, \theta)^2 + w_L c^2(x, y, \theta)^2 + w_A c^3(x, y, \theta)^2};$$

this estimate is illustrated in Fig. 9 where it is contrasted against the exact metric.

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 Definition 18.

Proposition 21 (Kernel compatibility of ρ_G) *Let \mathcal{G} be a G -invariant metric tensor field on G/H , then we have*

$$\forall p \in G/H, \forall h \in H : \rho_G(hp) = \rho_G(p). \quad (36)$$

Note that, since we use left cosets, $ph = p$ but $hp \neq p$ in general, this requirement is not trivial. Proof of this proposition is included in Appendix A.2.

Now that we have developed and analyzed the logarithmic metric estimate; we can use it to construct an approximation to the diffusion kernel for $\alpha = 1$.

Definition 22 (Approximate $\alpha = 1$ kernel)

$$K_t^{1, \text{appr}}(p) := \eta_t \exp \left(-\frac{\rho_G(p)^2}{4t} \right) \quad (37)$$

where η_t is a normalization constant for a given t ; this can either be the L^1 normalization constant or in the case of groups of polynomial growth one typically sets $\eta_t = \mu_G(B(p_0, \sqrt{t}))^{-1}$, see the definition of polynomial growth below.

On Lie groups of polynomial growth this approximate kernel be bounded from above and below by the exact kernels.

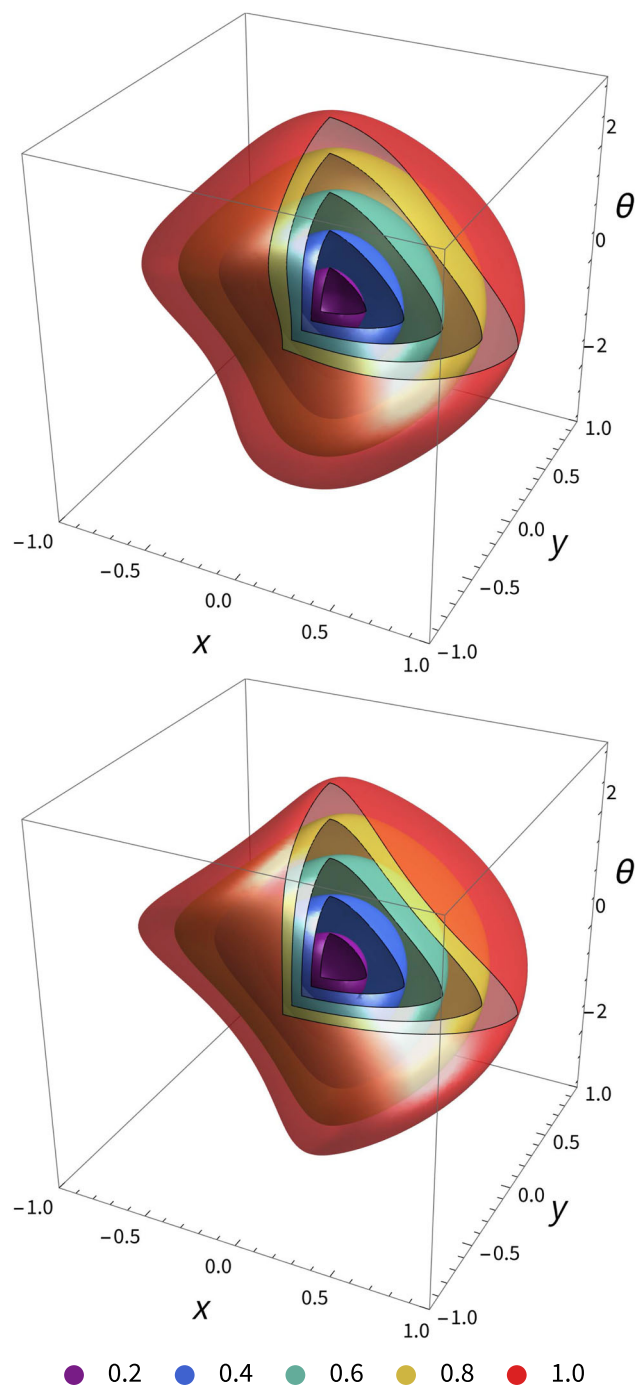


Fig. 9 Comparing the ‘exact’ Riemannian distance (top) obtained through numerically solving the Eikonal equation [29] versus the logarithmic metric estimate (bottom) on $SE(2)$ endowed with a left-invariant Riemannian metric tensor field (20) with $w_M = 1$, $w_L = 2$, $w_A = 1/\pi$. The relative L^1 error in the plotted volume is 0.20

Definition 23 (Polynomial growth) A Lie group G with left-invariant Haar measure μ_G is of polynomial growth when the volume of a sphere of radius r around $g \in G$:

$$B(g, r) = \{g' \in G \mid d_{\tilde{G}}(g, g') < r\},$$

can be polynomially bounded as follows: there exist constants $\delta > 0$ and $C_{\text{grow}} > 0$ so that

$$\frac{1}{C_{\text{grow}}} r^\delta \leq \mu_G(B(g, r)) \leq C_{\text{grow}} r^\delta, \quad r \geq 1,$$

take note that the exponent δ is the same on both the lower and upper bound. Since μ_G is left invariant, the choice of g does not matter.

Lemma 24 *Let G be of polynomial growth, and let K_t^1 be the fundamental solution to the $\alpha = 1$ diffusion equation on G/H , then there exist constants $C \geq 1$, $D_1 \in (0, 1)$ and $D_2 > D_1$ so that for all $t > 0$ the following holds:*

$$\frac{1}{C} K_{D_1 t}^1(p) \leq K_t^{1, \text{appr}}(p) \leq C K_{D_2 t}^1(p). \quad (38)$$

for all $p \in G/H$.

Proof On a group of polynomial growth, we have $\eta_t = \mu_G(B(p_0, \sqrt{t}))^{-1}$. If G is of polynomial growth, we can apply [86, Thm. 2.12] to find that there exist constants $C_1, C_2 > 0$, and for all $\varepsilon > 0$ there exist a constant C_ε so that:

$$\begin{aligned} C_1 \eta_t \exp\left(-\frac{d_G(p_0, p)^2}{4C_2 t}\right) &\leq K_t^1(p) \\ &\leq C_\varepsilon \eta_t \exp\left(-\frac{d_G(p_0, p)^2}{4(1+\varepsilon)t}\right). \end{aligned}$$

□

Remark 11 (Left vs. right cosets) Note that Maheux [86] uses right cosets while we use left cosets. We can translate the results easily by inversion in view of $(gH)^{-1} = H^{-1}g^{-1} = Hg^{-1}$. We then apply the result of Maheux to the correct (invertible) G -invariant metric tensor field on G/H .

Also note the different (but equivalent) way Maheux relates distance on the group with distance on the homogeneous space. While we use a pseudometric on G induced by a metric on G/H , Maheux uses a metric on G/H induced by a metric on G by:

$$\begin{aligned} d_{G/H}^{\text{maheux}}(p_1, p_2) &= \inf_{g_1 \in p_1} \inf_{g_2 \in p_2} d_G^{\text{maheux}}(g_1, g_2) \\ &= \inf_{g_2 \in p_2} d_G^{\text{maheux}}(q_1, g_2), \end{aligned} \quad (39)$$

for any choice of $q_1 \in p_1$. We avoid having to minimize inside the cosets as in (39) thanks to the inherent symmetries in our pseudometric.

Now using the inequalities from Lemma 20, we obtain:

$$\begin{aligned} C_1 \eta_t \exp\left(-\frac{\rho_G(p)^2}{4C_2 t}\right) &\leq K_t^1(p) \\ &\leq C_\varepsilon \eta_t \exp\left(-\frac{\rho_G(p)^2}{4C_{\text{metr}}^2(1+\varepsilon)t}\right), \end{aligned}$$

which leads to:

$$\begin{aligned} C_1 \frac{\eta_t}{\eta_{C_2 t}} K_{C_2 t}^{1, \text{appr}}(p) &\leq K_t^1(p) \\ &\leq C_\varepsilon \frac{\eta_t}{\eta_{C_{\text{metr}}^2(1+\varepsilon)t}} K_{C_{\text{metr}}^2(1+\varepsilon)t}^{1, \text{appr}}(p). \end{aligned}$$

The group G being of polynomial growth also implies G/H is a doubling space [86, Thm. 2.17]. Using the volume doubling and reverse volume doubling property of doubling spaces [87, Prop. 3.2 and 3.3], we find that there exist constants $C_3, C_4, \beta, \beta' > 0$ so that:

$$\begin{aligned} \frac{\eta_t}{\eta_{C_2 t}} &\geq C_3 \left(\frac{\sqrt{t}}{\sqrt{C_2 t}}\right)^\beta = C_3 C_2^{-\beta/2}, \\ \frac{\eta_t}{\eta_{C_{\text{metr}}^2(1+\varepsilon)t}} &\leq C_4 \left(\frac{\sqrt{t}}{\sqrt{C_{\text{metr}}^2(1+\varepsilon)t}}\right)^{\beta'} \\ &= C_4 \left(C_{\text{metr}}^2(1+\varepsilon)\right)^{-\beta'/2}. \end{aligned}$$

Applying these inequalities, we get:

$$C'_1 := C_1 C_3 C_2^{-\beta/2}$$

and

$$C'_\varepsilon := C_\varepsilon C_4 \left(C_{\text{metr}}^2(1+\varepsilon)\right)^{-\beta'/2}$$

; we obtain:

$$C'_1 K_{C_2 t}^{1, \text{appr}}(p) \leq K_t^1(p) \leq C'_\varepsilon K_{C_{\text{metr}}^2(1+\varepsilon)t}^{1, \text{appr}}(p).$$

Reparametrizing t in both inequalities gives:

$$\frac{1}{C'_\varepsilon} K_{t/(C_{\text{metr}}^2(1+\varepsilon))}^1(p) \leq K_t^{1, \text{appr}}(p) \leq \frac{1}{C'_1} K_{C_2^{-1}t}^1(p).$$

Finally, we fix $\varepsilon > 0$ and relabel constants:

$$C := \max\{C_1^{-1}, C'_\varepsilon, 1\},$$

$$D_1 := \frac{1}{C_{\text{metr}}^2(1+\varepsilon)},$$

$$D_2 := \frac{1}{C_2};$$

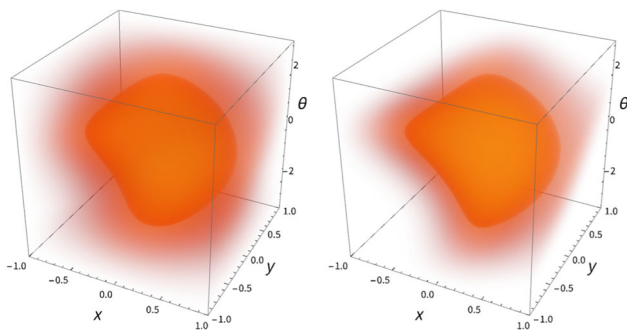


Fig. 10 Comparing the numerically computed heat kernel K_t^1 (left) with our approximation $K_t^{1,appr}$ based on the logarithmic norm estimate (right) for $G/H = SE(2)$. Shown here at $t = 1$ with the same metric as in Fig. 9. Especially in deep learning applications where discretization is very coarse, our approximation is sufficiently accurate as long as the spatial anisotropies w_M/w_L and w_L/w_M do not become too high. In this case, with $w_L/w_M = 2$ we have a relative L^2 error of 0.23 in the plotted volume

observe that since $\varepsilon > 0$ and $C_{\text{metr}} \geq 1$, we have $0 < D_1 < 1$. \square

Depending on the actually achievable constants, Lemma 24 provides a very strong or very weak bound on how much our approximation deviates from the fundamental solution. Fortunately in the $SE(2)$ case, our approximation is very close to the exact kernel in the vicinity of the origin, as shown in Fig. 10. In our experiments, we sample the kernel on a grid around the origin; hence, this approximation is good for reasonable values of the metric parameters, which we may expect from Lemma 20 providing a second-order relative error.

Now let us develop an approximation for values of α other than 1. From semi-group theory [88], it follows that semi-groups generated by taking fractional powers of the generator (in our case $\Delta_G \rightarrow -(-\Delta_G)^\alpha$) amounts to the following key relation between the α -kernel and the diffusion kernel:

$$K_t^\alpha(p) := \int_0^\infty q_{t,\alpha}(\tau) K_\tau^1(p) d\tau, \quad (40)$$

for $\alpha \in (0, 1)$ and $t > 0$ where $q_{t,\alpha}$ is defined as follows.

Definition 25 Let \mathcal{L}^{-1} be the inverse Laplace transform then

$$q_{t,\alpha}(\tau) := \mathcal{L}^{-1} \left(r \mapsto e^{-tr^\alpha} \right) (\tau) \quad \text{for } \tau \geq 0.$$

For explicit formulas of this kernel, see [88, Ch. IX:11 eq. 17]. Since e^{-tr^α} is positive for all r , it follows that $q_{t,\alpha}$ is also positive everywhere.

Now instead of integrating K_t^1 to obtain the exact fundamental solution, we can replace it with our approximation $K_t^{1,appr}$ to obtain an approximate α -kernel.

Definition 26 (Approximate $\alpha \in (0, 1)$ kernel) Akin to (40), we set $\alpha \in (0, 1)$, $t > 0$ and define:

$$K_t^{\alpha,appr}(p) := \int_0^\infty q_{t,\alpha}(\tau) K_\tau^{1,appr}(p) d\tau \geq 0, \quad (41)$$

for $p \in G/H$.

The bounding of K_t^1 we obtained in Lemma 24 transfers directly to our approximation for other α .

Theorem 27 Let G be of polynomial growth and let K_t^α be the fundamental solution to the $\alpha \in (0, 1]$ diffusion equation on G/H , then there exists constants $C \geq 1$, $D_1 \in (0, 1)$ and $D_2 > D_1$ so that for all $t > 0$ and $p \in G/H$ the following holds:

$$\frac{1}{C} K_{D_1 t}^\alpha(p) \leq K_t^{\alpha,appr}(p) \leq C K_{D_2 t}^\alpha(p). \quad (42)$$

Proof This is a consequence of Lemma 24 and the fact that $q_{t,\alpha}$ is positive, applying the integral from (40) yields:

$$\begin{aligned} K_t^{\alpha,appr}(p) &= \int_0^\infty q_{t,\alpha}(\tau) K_\tau^{1,appr}(p) d\tau \\ &\stackrel{\text{(Lemma 24)}}{\leq} C \int_0^\infty q_{t,\alpha}(\tau) K_{D_2 \tau}^1(p) d\tau \\ &\stackrel{(\tau' = D_2 \tau)}{=} C \int_0^\infty \frac{1}{D_2} q_{t,\alpha} \left(\frac{\tau'}{D_2} \right) K_{\tau'}^1(p) d\tau' \\ &\stackrel{\text{(Bromwich integral)}}{=} C \int_0^\infty q_{D_2 t, \alpha}(\tau') K_{\tau'}^1(p) d\tau' \\ &= C K_{D_2 t}^\alpha(p). \end{aligned}$$

The other inequality works the same way. \square

Although the approximation (41) is helpful in the proof above it contains some integration and is not an explicit expression. Our initial experiments with diffusion for $\alpha = 1$ showed that (for the applications under consideration at least) adding diffusion did not improve performance. For that reason, we chose not to focus further on diffusion in this work. We leave developing a more explicit and computable approximation for diffusion kernels for $0 < \alpha < 1$ for future work.

5.3 Dilation and Erosion

The dilation/erosion step solves the PDE

$$\begin{cases} \frac{\partial W_3}{\partial t}(p, t) = \pm \left\| \nabla_{\mathcal{G}_2^\pm} W_3(p, t) \right\|_{\mathcal{G}_2^\pm}^{2\alpha} & \text{for } p \in G/H, \\ & t \geq 0, \\ W_3(p, 0) = U_3(p) & \text{for } p \in G/H. \end{cases} \quad (43)$$

By a generalization of the Hopf–Lax formula [89, Ch.10.3], the solution is given by morphological convolution

$$W_3(p, t) = -(k_t^\alpha \square_G - U_3)(p) \quad (44)$$

for the (+) (dilation) variant and

$$W_3(p, t) = (k_t^\alpha \square_G U_3)(p) \quad (45)$$

for the (−) (erosion) variant, where the kernel k_t^α (also called the structuring element in the context of morphology) is given as follows.

Definition 28 (Dilation/erosion kernels) The morphological convolution kernel k_t^α for small times t and $\alpha \in (1/2, 1]$ is given by

$$k_t^\alpha(p) := v_\alpha t^{-\frac{1}{2\alpha-1}} d_{\mathcal{G}_2}(p_0, p)^{\frac{2\alpha}{2\alpha-1}}, \quad (46)$$

with $v_\alpha := \left(\frac{2\alpha-1}{(2\alpha)^{2\alpha/(2\alpha-1)}}\right)$ and for $\alpha = 1/2$ by

$$k_t^{1/2}(p) = \begin{cases} 0 & \text{if } d_{\mathcal{G}_2}(p_0, p) \leq t, \\ \infty & \text{if } d_{\mathcal{G}_2}(p_0, p) > t. \end{cases} \quad (47)$$

In the above definition and for the rest of the section, we write \mathcal{G}_2 for either \mathcal{G}_2^+ or \mathcal{G}_2^- depending on whether we are dealing with the dilation or erosion variant. The morphological convolution \square_G (alternatively: the infimal convolution) is specified as follows.

Definition 29 (Morphological group convolution) Let $f \in L^\infty(G/H)$, let $k : G/H \rightarrow \mathbb{R} \cup \{\infty\}$ be proper (not everywhere ∞), then we define:

$$\begin{aligned} (k \square_G f)(p) &:= \inf_{g \in G} \{k(g^{-1}p) + f(gp_0)\} \\ &= \inf_{g \in G} \{k(g^{-1}p) + f(gH)\}. \end{aligned}$$

Remark 12 (Grayscale morphology) Morphological convolution is related to the grayscale morphology operations \oplus (dilation) and \ominus (erosion) on \mathbb{R}^d as follows:

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

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 .

Combining morphological convolution with the structuring element k_t^α allows us to solve (43).

Theorem 30 Let G be of polynomial growth, let $\alpha \in (1/2, 1]$ and let $U_3 : G/H \rightarrow \mathbb{R}$ be Lipschitz. Then, $W_3 : G/H \times (0, \infty) \rightarrow \mathbb{R}$ given by

$$W_3(p, t) := (k_t^\alpha \square_G U_3)(p)$$

is Lipschitz and solves the (−)-variant, the erosion variant, of the system (43) in the sense of Theorem 2.1 in [90], while

$$W_3(p, t) := -(k_t^\alpha \square_G - U_3)(p)$$

is Lipschitz and solves the (+)-variant, the dilation variant, of system (43) in the sense of Theorem 2.1 in [90]. The kernels satisfy the semigroup property

$$k_t^\alpha \square_G k_s^\alpha = k_{t+s}^\alpha$$

for all $s, t \geq 0$ and $\alpha \in (1/2, 1]$.

Proof The Riemannian manifold $(G/H, \mathcal{G}_2)$ is a proper length space, and therefore, the theory of [90] applies. Moreover, since G is of polynomial growth, we have that G/H is a doubling space [86, Thm. 2.17] and also admits a Poincaré constant [86, Thm. 2.18]. So we meet the additional requirements of [90, Thm. 2.3 (vii) and (viii)].

The Hamiltonian $\mathcal{H} : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ in [90] is given by $\mathcal{H}(x) = x^{2\alpha}$. This Hamiltonian is indeed superlinear, convex, and satisfies $\mathcal{H}(0) = 0$. The corresponding Lagrangian $\mathcal{L} : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ becomes

$$\mathcal{L}(x) = v_\alpha x^{\frac{2\alpha}{2\alpha-1}}.$$

According to [90], the solution (in the sense of their Theorem 2.1) to the (−)-variant of system (43) is given by

$$\begin{aligned} W_3(p, t) &= \inf_{x \in G/H} \left\{ t \mathcal{L} \left(\frac{d_{\mathcal{G}_2}(p, x)}{t} \right) + U_3(x) \right\} \\ &= \inf_{g \in G} \left\{ t \mathcal{L} \left(\frac{d_{\mathcal{G}_2}(p, gp_0)}{t} \right) + U_3(gp_0) \right\} \\ &= \inf_{g \in G} \left\{ t \mathcal{L} \left(\frac{d_{\mathcal{G}_2}(g^{-1}p, p_0)}{t} \right) + U_3(gp_0) \right\} \\ &= \inf_{g \in G} \left\{ v_\alpha \frac{d_{\mathcal{G}_2}(g^{-1}p, p_0)^{\frac{2\alpha}{2\alpha-1}}}{t^{\frac{2\alpha}{2\alpha-1}-1}} + U_3(gp_0) \right\} \\ &= \inf_{g \in G} \left\{ v_\alpha t^{1-\frac{2\alpha}{2\alpha-1}} d_{\mathcal{G}_2}(g^{-1}p, p_0)^{\frac{2\alpha}{2\alpha-1}} + U_3(gp_0) \right\} \\ &= \inf_{g \in G} \left\{ v_\alpha t^{\frac{-1}{2\alpha-1}} d_{\mathcal{G}_2}(g^{-1}p, p_0)^{\frac{2\alpha}{2\alpha-1}} + U_3(gp_0) \right\} \\ &= (k_t^\alpha \square_G U_3)(p). \end{aligned}$$

The (+)-variant is proved analogously.

The semigroup property follows directly from [90, Thm 2.1(ii)]. \square

Remark 13 (Solution according to Balogh et al.) This theorem builds on the work by Balogh et al. [90] who provide a solution concept that is (potentially) different from the strong, weak or viscosity solution. The point of departure is to replace the norm of the gradient (i.e., the dual norm of the differential) with a metric subgradient, i.e., we replace $\|\nabla_{\mathcal{G}_2} W(p, t)\|_{\mathcal{G}_2}$ by:

$$\limsup_{p' \rightarrow p} \frac{\max(W(p, t) - W(p', t), 0)}{d_{\mathcal{G}_2}(p, p')},$$

and we get a solution concept in terms of this slightly different notion of a gradient.

Remark 14 Unique viscosity solutions For the case $\alpha = 1/2$, we lose the superlinearity of the Hamiltonian and can no longer apply Balogh et al.'s approach [90]. The solution for $\alpha > 1/2$ (46) converges pointwise to the solution for $\alpha = 1/2$ (47) as $\alpha \downarrow 1/2$. However, the solution concept changes from that of Balogh et al. to that of a viscosity solution [91,92]. In the general Riemannian homogeneous space setting, the result by Azagra [92, Thm 6.24] applies. It states that viscosity solutions of Eikonal PDEs on complete Riemannian manifolds are given by the distance map departing from the boundary of a given open and bounded set. As Eikonal equations directly relate to geodesically equidistant wavefront propagation on manifolds ([93, ch. 3], [29, ch. 4, app. E], [89]) one expects that the solutions (44),(45) of (43) are indeed the viscosity solutions (for resp. the + and --case) for $\alpha = 1/2$.

In many matrix Lie group quotients, like the Heisenberg group $H(2d + 1)$ studied in [94], or in our case of interest: the homogeneous space \mathbb{M}_d of positions and orientations) this is indeed the case. One can describe G -invariant vector fields via explicit coordinates and transfer HJB systems on G/H directly toward HJB-systems on \mathbb{R}^n or $\mathbb{R}^d \times S^q$, with $n = d + q = \dim(G/H)$. Then, one can directly apply results by Dragoni [91, Thm.4] and deduce that *our solutions, the dilations in (44) resp. erosions in (45), are indeed the unique viscosity solutions of HJB-PDE system (43) for the + and --case, for all $\alpha \in [1/2, 1]$.* Details are left for future research.

To get an idea of how the kernel in (46) operates in conjunction with morphological convolution, we take $G = G/H = \mathbb{R}$ and see how the operation evolves simple data, the kernels and results at $t = 1$ are shown in Fig. 12. Observe that with α close to $1/2$ (kernel and result in red) we obtain what amounts to an equivariant version of max/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. 11; for $\alpha = 1/2$ these are the stencils over which we would perform min/max pooling.

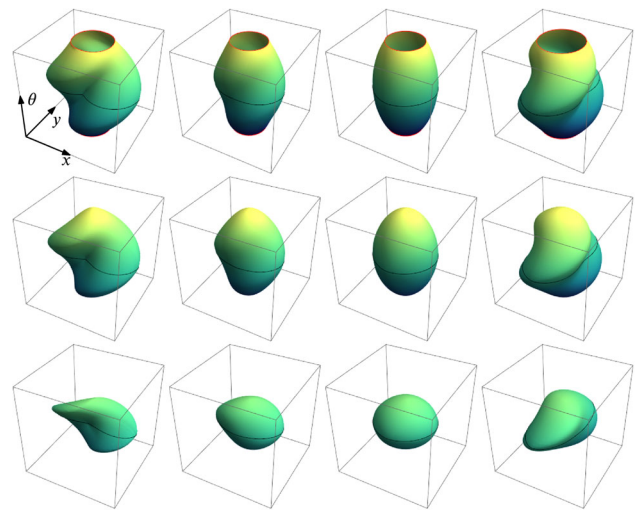


Fig. 11 Shapes of the level sets of the kernels on \mathbb{M}_2 for solving fractional diffusion (K_t^α) and dilation/erosion (k_t^α) for various values of the trainable metric tensor field parameters w_M , w_L and w_A . This shape is essentially what is being optimized during the training process of a metric tensor field on \mathbb{M}_2

Remark 15 The level sets in Fig. 11 are balls in $G/H = \mathbb{M}_2$ that do not depend on α . It is only the growth of the kernel values when passing through these level sets that depends on α . As such, the example $G/H = \mathbb{R}$ and Fig. 12 is very representative to the general G/H case. In the general G/H case, Fig. 11 still applies when one replaces the horizontal \mathbb{R} -axis with a signed distance along a minimizing geodesic in G/H passing through the origin. In that sense, $\alpha \in [1/2, 1]$ regulates soft-max pooling over Riemannian balls in G/H .

We can now define a more tangible approximate kernel by again replacing the exact metric $d_{\mathcal{G}_2}$ with the logarithmic approximation $\rho_{\mathcal{G}_2}$.

Definition 31 (Approximate dilation/erosion kernel) The approximate morphological convolution kernel $k_t^{\alpha, \text{appr}}$ for small times t and $\alpha \in [1/2, 1]$ is given by

$$k_t^{\alpha, \text{appr}}(p) := v_\alpha t^{-\frac{1}{2\alpha-1}} \rho_{\mathcal{G}_2}(p)^{\frac{2\alpha}{2\alpha-1}}, \quad (48)$$

with $v_\alpha := \left(\frac{2\alpha-1}{(2\alpha)^{2\alpha/(2\alpha-1)}} \right)$ and for $\alpha = 1/2$ by

$$k_t^{1/2, \text{appr}}(p) = \begin{cases} 0 & \text{if } \rho_{\mathcal{G}_2}(p) \leq t, \\ \infty & \text{if } \rho_{\mathcal{G}_2}(p) > t \end{cases}. \quad (49)$$

We used this approximation in our parallel GPU-algorithms (for our PDE-G-CNNs experiments in Sect. 7). It is highly preferable over the ‘exact’ solution based on the true distance as this would require Eikonal PDE solvers ([29,95]) which would not be practical for parallel GPU implementations of

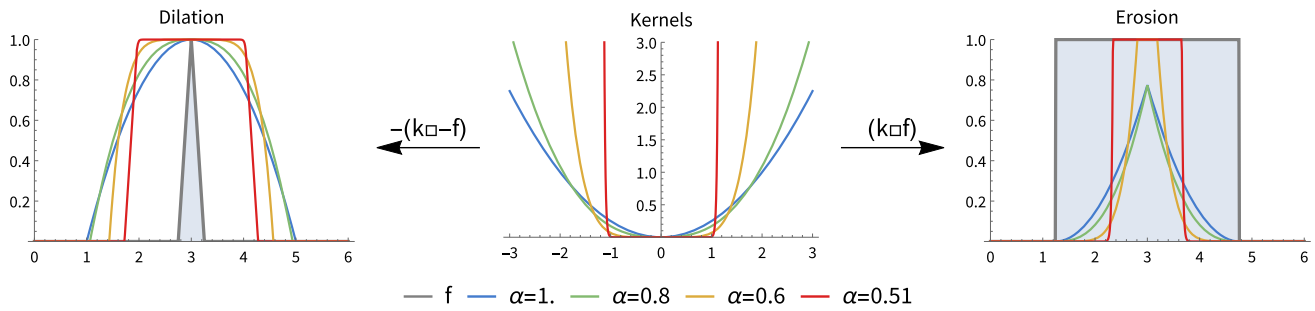


Fig. 12 In the center, we have kernels of the type (46) in \mathbb{R} (or the signed distance on a manifold of choice) for some $\alpha \in (1/2, 1]$ and $t = 1$, which solves dilation/erosion. For $\alpha \rightarrow 1/2$, this kernel converges to the type in (47), i.e., the solution is obtained by max/min pooling. On the left, we morphologically convolve a spike (in gray) with a few of these ker-

nels; 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 effects of these operations in the image processing context can also be seen in the last two columns of Fig. 6

PDE-G-CNNs. Again the approximations are reasonable as long as the spatial anisotropy does not get too high, see Fig. 9 for an example.

Next we formalize the theoretical underpinning of the approximations in the upcoming corollary.

An immediate consequence of Definition 31 and Lemma 20 (keeping in mind that the kernel expressions in Definition 31 are monotonic w.r.t. $\rho := \rho_{G_2}(p)$) is that we can enclose our approximate morphological kernel with the exact morphological kernels in the same way as we did for the (fractional) diffusion kernel in Theorem 27. This proves the following Corollaries:

Corollary 32 *Let $\alpha \in (1/2, 1]$, then for all $t > 0$*

$$k_t^\alpha(p) \leq k_t^{\alpha, \text{appr}}(p) \leq C_{\text{metr}}^{\frac{2\alpha}{2\alpha-1}} k_t^\alpha(p) \quad \text{for } p \in G/H.$$

For the case $\alpha = 1/2$, the approximation is exact in an inner and outer region:

$$\begin{aligned} k_t^{1/2, \text{appr}}(p) &= k_t^{1/2}(p) = 0 & \text{if } \rho_{G_2}(p)^{2\alpha} \leq t, \\ k_t^{1/2, \text{appr}}(p) &= k_t^{1/2}(p) = \infty & \text{if } d_{G_2}(p_0, p)^{2\alpha} > t, \end{aligned}$$

but in the intermediate region where $\rho_{G_2}(p)^{2\alpha} > t$ and $d_{G_2}(p_0, p)^{2\alpha} \leq t$ we have $k_t^{1/2, \text{appr}} = \infty$ while $k_t^{1/2} = 0$.

Alternatively, instead of bounding by value we can bound in time, in which case we do not need to distinguish different cases of α .

Corollary 33 *Let $\alpha \in [1/2, 1]$, $t > 0$ then for all $p \in G/H$ one has*

$$k_t^\alpha(p) \leq k_t^{\alpha, \text{appr}}(p) \leq k_{C_{\text{metr}}^{-2\alpha}t}^\alpha(p)$$

With these two bounds on our approximate morphological kernels, we end our theoretical results.

6 Generalization of (Group-)CNNs

In this section, we point out the similarities between common (G-)CNN operations and our PDE-based approach. Our goal here is not so much claiming that our PDE approach serves as a useful model for analyzing (G-)CNNs, but that modern CNNs already bear some resemblance to a network of PDE solvers. Noticing that similarity, our approach is then just taking the next logical step by structuring a network to explicitly solve a set of PDEs.

6.1 Discrete Convolution as Convection and Diffusion

Now that we have seen how PDE-G-CNNs are designed we show how they generalize conventional G-CNNs. Starting with an initial condition U , we show how group convolution with a general kernel k can be interpreted as a superposition of solutions (27) of convection PDEs:

$$\begin{aligned} (k *_{G/H} U)(p) &= \frac{1}{\mu_G(H)} \int_G k(g^{-1}p) U(gp_0) d\mu_G(g) \\ &= \frac{1}{\mu_G(H)} \int_G k(g^{-1}g_p p_0) U(gp_0) d\mu_G(g), \end{aligned}$$

for any $g_p \in p$, now change variables to $q = g_p^{-1}g$ and recall that μ_G is left invariant:

$$= \frac{1}{\mu_G(H)} \int_G k(q^{-1}p_0) U(g_p q p_0) d\mu_G(q).$$

In this last expression, we recognize (27) and see that we can interpret $p \mapsto U(g_p q p_0)$ as the solution of the convection PDE (25) at time $t = 1$ for a convection vector field c that has flow lines given by $\gamma_c(t) = \exp_G(-t \log_G q) p_0$ so that $(\gamma_c(1))^{-1} p_0 = q p_0$. As a result, the output $k *_{G/H} U$ can

then be seen as a weighted sum of solutions over all possible left invariant convection vector fields.

Using this result, we can consider what happens in the discrete case where we take the kernel k to be a linear combination of displaced diffusion kernels K_t^α (for some choice of α) as follows:

$$k(p) = \sum_{i=1}^n k_i K_{t_i}^\alpha (g_i^{-1} p), \quad (50)$$

where for all i we fix a weight $k_i \in \mathbb{R}$, diffusion time $t_i \geq 0$ and a displacement $g_i \in G$. Convolving with this kernel yields:

$$\begin{aligned} (k *_{G/H} U)(p) &= \int_G \sum_{i=1}^n k_i K_{t_i}^\alpha (g_i^{-1} g^{-1} p) U(gp_0) d\mu_G(g) \\ &= \sum_{i=1}^n k_i \int_G K_{t_i}^\alpha (g_i^{-1} g^{-1} p) U(gp_0) d\mu_G(g), \end{aligned}$$

we change variables to $h = g g_i$:

$$\begin{aligned} &= \sum_{i=1}^n k_i \int_G K_{t_i}^\alpha (h^{-1} p) U(h g_i^{-1} p_0) d\mu_G(h) \\ &= \sum_{i=1}^n k_i \left(K_{t_i}^\alpha *_{G/H} [q \mapsto U(g_i g_i^{-1} p_0)] \right) (p). \end{aligned}$$

Here again we recognize $q \mapsto U(g_i g_i^{-1} p_0)$ as the solution (27) of the convection PDE at $t = 1$ with flow lines induced by $\gamma_c(t) = \exp_G(t \log_G g_i)$. Subsequently, we take these solutions and convolve them with a (fractional) diffusion kernel with scale t_i , i.e., after convection we apply the fractional diffusion PDE with evolution time t_i and finally make a linear combination of the results.

We can conclude that G-CNNs fit in our PDE-based model by looking at a single discretized group convolution as a set of single-step PDE units working on an input, without the morphological convolution and with specific choices made for the convection vector fields and diffusion times.

6.2 Max Pooling as Morphological Convolution

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

Proposition 34 (Max pooling) *Let $f \in L^\infty(G/H)$; let $S \subset G/H$ be nonempty and define $k_S : G/H \rightarrow \mathbb{R} \cup \{\infty\}$ as:*

$$k_S(p) := \begin{cases} 0 & \text{if } p \in S, \\ \infty & \text{else.} \end{cases} \quad (51)$$

Then,

$$-(k_S \square - f)(p) = \sup_{g \in G: g^{-1} p \in S} f(gp_0). \quad (52)$$

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

Proof Filling in (51) into Definition 29 yields:

$$\begin{aligned} &-(k_S \square - f)(p) \\ &= -\inf \left\{ \inf_{g \in G: g^{-1} p \in S} -f(gp_0), \inf_{g \in G: g^{-1} p \notin S} \right. \\ &\quad \left. \times -f(gp_0) + \infty \right\} \\ &= -\inf_{g \in G: g^{-1} p \in S} -f(gp_0) \\ &= \sup_{g \in G: g^{-1} p \in S} f(gp_0). \end{aligned}$$

□

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

Corollary 35 (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:*

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

for all $x \in \mathbb{R}^n$.

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.

6.3 ReLUs as Morphological Convolution

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

Proposition 36 *Let f be a compactly supported continuous function on G/H . Then, dilation with the kernel*

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

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

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

Proof Filling in k into the definition of morphological convolution:

$$\begin{aligned} & -(k_{\text{ReLU},f} \square_G -f)(p) \\ &= -\inf_{g \in G} k_{\text{ReLU}}(g^{-1}p) - f(g \cdot p_0) \\ &= -\inf_{g \in G} \left\{ \inf_{g^{-1}p=p_0} -f(gp_0), \right. \\ & \quad \left. \times \inf_{g^{-1}p \neq p_0} -f(gp_0) + \sup_{y \in G/H} f(y) \right\} \\ &= \sup \left\{ f(p), \sup_{z \in G/H: z \neq p} f(z) - \sup_{y \in G/H} f(y) \right\}, \end{aligned}$$

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

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

□

We conclude that morphological convolution allows us to:

- do pooling in an equivariant manner with transformations other than translation,
- do *soft* pooling that is continuous under domain transformations (illustrated in Fig. 12),
- learn the pooling region by considering the kernel k as trainable,
- effectively fold the action of a ReLU into trainable nonlinearities.

6.4 Residual Networks

So-called *residual networks* [67] were introduced mainly as a means of dealing with the vanishing gradient problem in very deep networks, aiding trainability. These networks use so-called residual blocks, illustrated in Fig. 13, that feature a skip connection to group a few layers together to produce a delta-map that gets added to the input.

This *identity + delta* structure is very reminiscent of a forward Euler discretization scheme. If we had an evolution equation of the type

$$\begin{cases} \frac{\partial U}{\partial t}(p, t) = \mathcal{F}(U(\cdot, t), p) & \text{for } p \in M, t \geq 0, \\ U(p, 0) = U_0(p) & \text{for } p \in M, \end{cases}$$

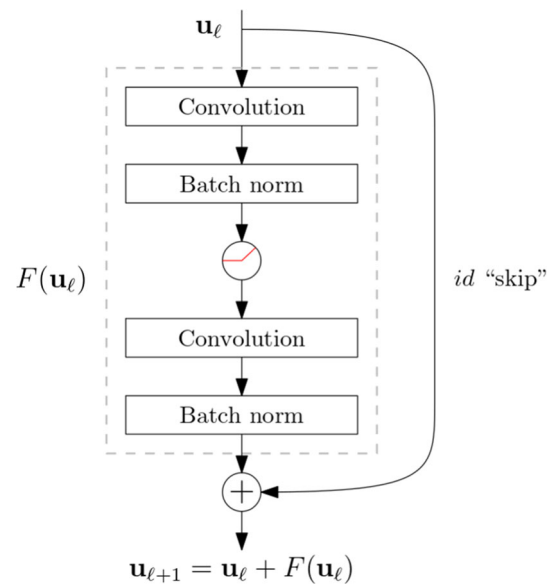


Fig. 13 A residual block, like in [67], note the resemblance to a forward Euler discretization scheme

with some operator $\mathcal{F} : L^\infty(M) \times M \rightarrow \mathbb{R}$, we could solve it approximately by stepping forward with:

$$U(p, t + \Delta t) = U(p, t) + \Delta t \mathcal{F}(U(\cdot, t), p),$$

for some time step $\Delta t > 0$. We see that this is very similar to what is implemented in the residual block in Fig. 13 once we discretize it.

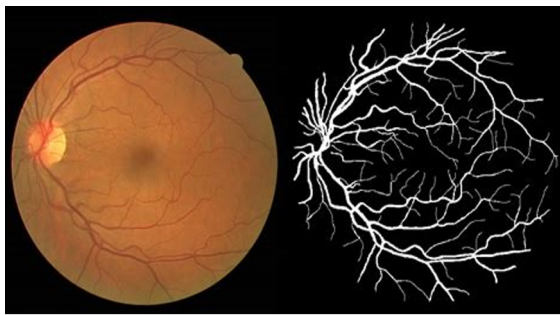
The correspondence is far from exact given that multiple channels are being combined in residual blocks, so we can not easily describe a residual block with a PDE. Still, our takeaway is that residual networks and skip connections have moved CNNs from networks that *change* data to networks that *evolve* data.

For this reason, we speculate that deep PDE-G-CNNs will not need (or have the same benefit from) skip connections; we leave this subject for future investigation. More discussion on the relation between residual networks and PDEs can be found in [77].

7 Experiments

To demonstrate the viability of PDE-based CNNs, we perform two experiments where we compare the performance of PDE-G-CNNs against G-CNNs and classic CNNs. We will be doing a vessel segmentation and digit classification problem: two straightforward applications of CNNs. Examples of these two applications are illustrated in Fig. 14.

The goal of the experiments is to compare the basic building blocks of different types of networks in clearly defined feed-forward network architectures. So we test networks of



(a) Example from the DRIVE [96] dataset, showing a retinal image and its vessel segmentation.



(b) Examples from the RotNIST [97] dataset.

Fig. 14 We perform a segmentation experiment on retinal vessel images and a classification experiment on rotation augmented digits

modest size only and do not just aim for the performance that would be possible with large-scale networks.

7.1 Implementation

We implemented our PDE-based operators in an extension to the PyTorch deep learning framework [98]. Our package is called *LieTorch* and is open source. It is available at <https://gitlab.com/bsmetsjr/lietorch>.

The operations we have proposed in the paper have been implemented in C++ for CPUs and CUDA for Nvidia GPUs but can be used from Python through PyTorch. Our package was also designed with modularity in mind: we provide a host of PyTorch modules that can be used together to implement the PDE-G-CNNs we proposed, but that can also be used separately to experiment with other architectures.

All the modules we provide are differentiable and so our PDE-G-CNNs are trainable through stochastic gradient descent (or its many variants) in the usual manner. In our experiments, we have had good results with using the ADAM [99] optimizer.

All the network models and training scripts used in the experiments are also available in the repository.

7.2 Design Choices

Several design choices are common to both experiments, we will go over these now.

First, we choose $G/H = \mathbb{M}_2$ for our G-CNNs and PDE-G-CNNs and so go for roto-translation equivariant networks. In all instances, we lift to 8 orientations.

Second, we use the convection, dilation and erosion version of (24); hence, we refer to these networks as PDE-CNNs of the CDE-type. Each PDE-layer is implemented as in Fig. 1 with the single-pass PDE solver from Fig. 2 without the convolution. So no explicit diffusion is used, and the layer consists of just resampling and two morphological convolutions. Since we do the resampling using trilinear interpolation, this does introduce a small amount of implicit diffusion.

Remark 16 (Role of diffusion) In these experiments, we found no benefit to adding diffusion to the networks. Diffusion likely would be of benefit when the input data is noisy but neither datasets we used are noisy and we have not yet performed experiments with adding noise. We leave this investigation for future work.

Third, we fix $\alpha = 0.65$. We came to this value empirically; the networks performed best with α -values in the range 0.6–0.7. Looking at Fig. 12, we can conjecture that $\alpha = 0.65$ is the “sweet spot” between sharpness and smoothness. When the kernel is too sharp (α close to $1/2$), minor perturbations in the input can have large effects on the output; when the kernel is too smooth (α close to 1) the output will be smoothed out too much as well.

Fourth, all our networks are simple feed-forward networks.

Finally, we use the ADAM optimizer [99] together with L^2 regularization uniformly over all parameters with a factor of 0.005.

7.3 DRIVE Retinal Vessel Segmentation

The first experiment uses the DRIVE retinal vessel segmentation dataset [96]. The object is the generate a binary mask indicating the location of blood vessels from a color image of a retina as illustrated in Fig. 14a.

We test 6- and 12-layer variants of a CNN, a G-CNN and a CDE-PDE-CNN. The layout of the 6-layer networks is shown in Fig. 15; the 12-layer networks simply add more convolution, group convolution or CDE layers. All the networks were trained on the same training data and tested on the same testing data.

The output of the network is passed through a sigmoid function to produce a 2D map a of values in the range $[0, 1]$ which we compare against the known segmentation map b

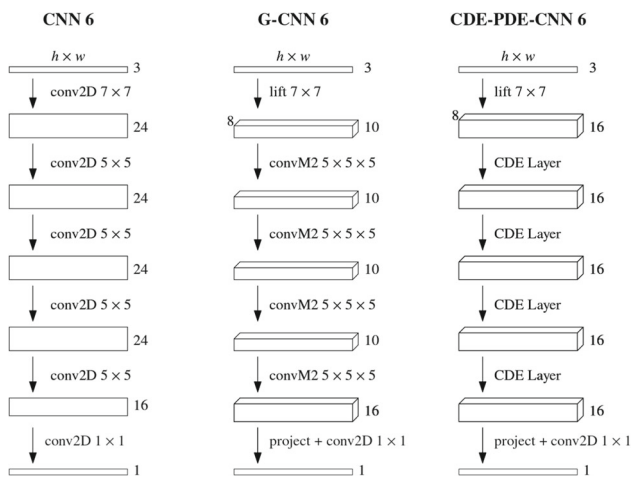


Fig. 15 Schematic of the 6-layer models used on our segmentation experiments. Kernel sizes and number of feature channels in each layer are indicated, depth indicates that the data lives on \mathbb{M}_2 . Omitted are activation functions, batch normalization, padding and dropout modules. The 12-layer models are essentially the same but with double the number of layers but with reduced number of channels per layer (i.e., reduced width) for the CDE-PDE-CNN (hence the reduction in parameters going from 6 to 12 layers)

with values in $\{0, 1\}$. We use the continuous DICE coefficient as the loss function:

$$\text{loss}(a, b) = 1 - \frac{2 \sum ab + \varepsilon}{\sum a + \sum b + \varepsilon},$$

where the sum \sum is over all the values in the 2D map. A relatively small $\varepsilon = 1$ is used to avoid divide-by-zero issues and the $a \equiv b \equiv 0$ edge case.

The 6-layer networks were trained over 60 epochs, starting with a learning rate of 0.01 that we decay exponentially with a gamma of 0.95. The 12-layer networks were trained over 80 epochs, starting from the same learning rate but with a learning rate gamma of 0.96.

We measure the performance of the network by the DICE coefficient obtained on the 20 images of the testing dataset. We trained each model 10 times, the results of which are summarized in Table 1 and Fig. 16a.

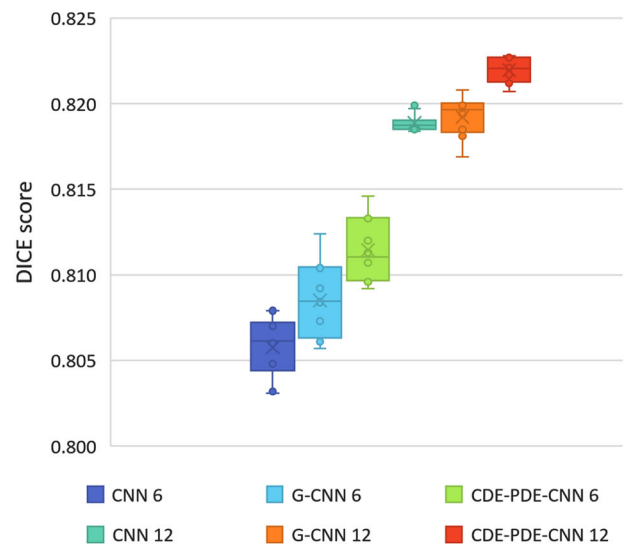
We achieve similar or better performance than CNNs or G-CNNs but with a vast reduction in parameters. Scaling from 6 to 12 layers even allows us to reduce the total number of parameters of the PDE-G-CNN while still increasing performance; this is achieved by reducing the number of channels (i.e., the width) of the network, see also Table 2.

7.4 RotNIST Digit Classification

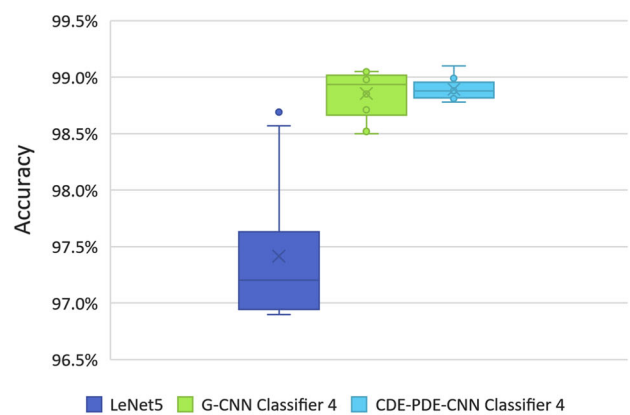
The second experiment we performed is the classic digit classification experiment. Instead of using the plain MNIST dataset, we did the experiment on the RotNIST dataset [97].

Table 1 Average DICE coefficient achieved on the 20 images of the testing dataset and the number of trainable parameters of each model. The G-CNNs and CDE-PDE-CNNs are roto-translation equivariant by construction. Note the vast reduction in parameters allowed by using PDE-based networks

Model	Parameters	DICE score \pm std.dev.
CNN 6	47352	0.8058 \pm 0.0017
G-CNN 6	39258	0.8085 \pm 0.0022
CDE-PDE-CNN 6	4128	0.8115 \pm 0.0018
CNN 12	129432	0.8189 \pm 0.0005
G-CNN 12	114378	0.8192 \pm 0.0012
CDE-PDE-CNN 12	3678	0.8220 \pm 0.0007



(a) Performance on retinal vessel segmentation. We test 6- and 12-layer variants of conventional CNNs, G-CNNs and our PDE-CNNs, each network is trained 10 times, the chart shows the distribution of DICE performances on the test dataset.



(b) Performance of digit classification on the RotNIST dataset. We compare the classic 5-layer LeNet against a 4-layer G-CNN and PDE-CNN. LeNet was trained for 120 epochs, the other two for 60 epochs.

Fig. 16 Comparison of PDE-based networks against conventional CNNs and group CNNs on segmentation and classification tasks

Table 2 Allocation of parameters for the 6- and 12-layer CDE-PDE-CNNs used in the vessel segmentation experiment. The added depth of the networks allows us to shrink the width. With the network having less channels over all we can also shrink the number of channels in the lifting layer, which drastically reduces the total number of parameters

Type of parameter	CDE-PDE-CNN 6	CDE-PDE-CNN 12
Lifting layer	2352	1470
Convection	192	300
Dilation	192	300
Erosion	192	300
Linear combinations	1040	1076
Batch normalization	160	232

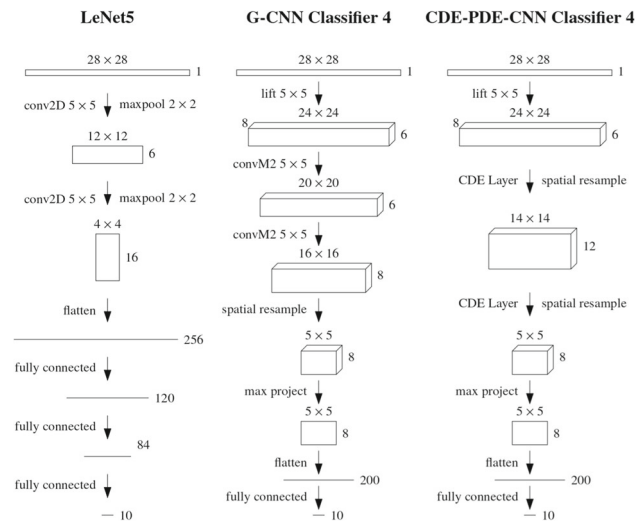


Fig. 17 Schematic of the three models tested with the RotNIST data. Kernel sizes and number of feature channels in each layer are indicated. Omitted are activation functions, batch normalization and dropout modules

RotNIST contains the same images as MNIST but rotated to various degrees. Even though classifying rotated digits is a fairly artificial problem, we include this experiment to show that PDE-G-CNNs also work in a context very different from the first segmentation experiment. While our choice of PDEs derives from more traditional image processing methods, this experiment shows their utility in a basic image classification context.

We tested three networks: the classic LeNet5 CNN [100] as a baseline, a 4-layer G-CNN and a 4-layer CDE-PDE-CNN. The architectures of these three networks are illustrated in Fig. 17.

Table 3 Accuracy of the digit classification models on the testing dataset and number of parameters for each model

Model	Parameters	Error rate \pm std.dev.
CNN (LeNet5)	44426	2.59% \pm 0.66%
G-CNN Classifier 4	12700	1.14% \pm 0.21%
CDE-PDE-CNN Classifier 4	2542	1.10% \pm 0.10%

All three networks were trained on the same training data and tested on the same testing data. We train with a learning rate of 0.05 and a learning rate gamma of 0.96. We trained the LeNet5 model for 120 epochs and the G-CNN and CDE-PDE-CNN models for 60 epochs.

We measure the performance of the network by its accuracy on the testing dataset. We trained each model 10 times, the results of which are summarized in Table 3 and Fig. 16b.

We manage to get better performance than classic or group CNNs with far fewer parameters.

7.5 Computational Performance

Care was taken in optimizing the implementation to show that PDE-based networks can still achieve decent running times despite their higher computational complexity. In Table 4, we summarized the inferencing performance of each model we experimented with.

Our approach simultaneously gives us equivariance, a decrease in parameters and higher performance but at the cost of an increase in flops and memory footprint. While our implementation is reasonably optimized, it has had far less development time dedicated to it than the traditional CNN implementation provided by PyTorch/cuDNN, so we are confident more performance gains can be found.

In comparison with G-CNNs, our PDE-based networks are generally a little bit faster. Our G-CNN implementation is, however, less optimized compared to our PDE-G-CNN implementation. Were our G-CNN implementation equally optimized we expect G-CNNs to be slightly faster than the PDE-G-CNNs in our experiments.

Table 4 Time in seconds it took to run each model on the testing dataset of its respective experiment. The DRIVE testing dataset contains 20 images while the RotNIST testing dataset contains 10000 digits

	CNN	G-CNN	PDE-CNN
DRIVE 6-layer	1.7s	6.5s	6.8s
DRIVE 12-layer	2.2s	14.1s	9.8s
RotNIST	0.1s	0.9s	0.7s

8 Conclusion

In this article, we presented the general mathematical framework of geometric PDEs on homogeneous spaces that underlies our PDE-G-CNNs. PDE-G-CNNs allow for a geometric and probabilistic interpretation of CNNs opening up new avenues for the study and development of these types of networks. We showed that additionally, PDE-G-CNNs have increased performance with a reduction in parameters.

PDE-G-CNNs ensure equivariance by design. The trainable parameters are geometrically relevant: they are left-invariant vector and tensor fields.

PDE-G-CNNs have three types of layers: convection, diffusion and erosion/dilation layers. We have shown that these layers implicitly include standard nonlinear operations in CNNs such as max pooling and ReLU activation.

To efficiently evaluate PDE evolution in the layers, we provided tangible analytical approximations to the relevant kernel operators on homogeneous spaces. In this article, we have underpinned the quality of the approximations in Theorems 27 and 30.

With two experiments we have verified that PDE-G-CNNs can improve performance over G-CNNs in the context of automatic vessel segmentation and digit classification. Most importantly, the performance increase is achieved with a vast reduction in the amount of trainable parameters.

Acknowledgements We gratefully acknowledge the Dutch Foundation of Science NWO for financial support (Duits: Geometric learning for Image Analysis, VI.C 202-031, Bekkers: Context Aware AI, VI nr.17290).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A

A.1 Proof of Lemma 20

The left inequality follows directly from the observation that $\rho_G(p)$ is exactly the Riemannian length of the curve

$$t \mapsto \exp_G(t \log_G(g_p))p_0$$

for $t \in [0, 1]$ and $g_p = \arg \min_{g \in p} \|\log_G g\|_{\tilde{G}}$. This continuous curve connects p_0 with p and as such has a greater length than the minimal length curve between those two points.

For the right inequality, consider the function $F : T_e G \rightarrow \mathbb{R}$ given by

$$F(v) := d_G(p_0, \pi \circ \exp_G(v))^2,$$

where we recall that $\pi : G \rightarrow G/H$ was given by (4). With the goal of making a Taylor expansion for this function, we note that:

- at the origin we have $F(0) = 0$,
- due to the chain rule applied to the squaring we have $dF|_0 = 0$.

Moreover, due to the G -invariance of d_G , the function F is even and consequently the 3rd order term of the Taylor expansion of F is zero.

For the second-order term, we are looking for the Hessian \mathcal{H} of F at $v = 0$. We split F into $F_1 := \pi \circ \exp_G$ and $F_2(p) := d_G(p_0, p)^2$ and find the Hessian of the composed function is

$$\begin{aligned} \mathcal{H}(F_2 \circ F_1)|_0(v, w) &= \mathcal{H}F_2|_{p_0}(dF_1|_0(v), dF_1|_0(w)) \\ &= 2\mathcal{G}|_{p_0}(d\pi|_e \circ d\exp_G|_0(v), d\pi|_e \circ d\exp_G|_0(w)) \\ &= 2\tilde{\mathcal{G}}|_e(d\exp_G|_0(v), d\exp_G|_0(w)) \\ &= 2\tilde{\mathcal{G}}|_e(v, w). \end{aligned}$$

Putting these facts together, we find:

$$F(v) = \tilde{\mathcal{G}}|_e(v, v) + O(\|v\|^4), \quad (53)$$

where $\|\cdot\|$ denotes some arbitrary norm on $T_e G$.

Now we take a linear subspace V of $T_e G$ that is independent from $T_e H$ but so that the span of $T_e H$ and V equals the entire $T_e G$, so that $T_e H \oplus V = T_e G$. Note that $\tilde{\mathcal{G}}|_e$ is only degenerated along $T_e H$, and so is a full norm when restricted to V , i.e., for all $v \in V$, $\tilde{\mathcal{G}}|_e(v, v) = 0$ only if $v = 0$. Therefore, there exists a $c > 0$ such that for all $v \in V$,

$$\tilde{\mathcal{G}}|_e(v, v) > c\|v\|^2,$$

and so by (53) we have:

$$F(v) = d_G(p_0, \pi \circ \exp_G(v))^2 > c\|v\|^2 + O(\|v\|^4). \quad (54)$$

Hence, for all $v \in V$ close enough to 0 we have:

$$d_G(p_0, \pi \circ \exp_G(v))^2 > \frac{c}{2}\|v\|^2. \quad (55)$$

In a neighborhood of the origin, the Lie group exponential map $\exp_G : T_e G \rightarrow G$ is a diffeomorphism to a neighborhood of e ; at the same time $\pi : G \rightarrow G/H$ is a smooth submersion by the Homogeneous Space Construction Theorem [101, Thm 21.17]. Consequently, $d(\pi \circ \exp_G)|_0 : V \rightarrow T_{p_0}(G/H)$ has full rank since $\pi \circ \exp_G$ is a local diffeomorphism between two spaces (V and G/H) with the same dimension; it follows by the inverse function theorem that there exists a neighborhood V_0 of 0 in V and a neighborhood P_0 of p_0 in G/H such that $\pi \circ \exp_G$ is a diffeomorphism from V_0 to P_0 . By possibly choosing V_0 smaller, we may assume by (54) that there exists a $C' > 0$ such that for all $v \in V_0$:

$$\tilde{G}|_e(v, v) \leq F(v) + C'\|v\|^4,$$

which, by using (55) yields

$$\tilde{G}|_e(v, v) \leq d_G(p_0, \pi \circ \exp_G(v))^2 + Cd_G(p_0, \pi \circ \exp_G(v))^4,$$

for all $v \in V_0$ and $C = C'\frac{c}{2} > 0$.

Now take a $p \in P_0$, then there exists a $w \in V_0$ so that

$$\pi \circ \exp_G w = p.$$

Call $g_p = \exp_G w$, then the previous inequality gives

$$\|\log_G g_p\|_{\tilde{G}}^2 \leq d_G(p, p_0)^2 + Cd_G(p, p_0)^4.$$

Clearly $g_p \in p$. Since $\rho_G(p)$ is the infimum of $\|\log_G g\|$ for all $g \in p$, it follows that $\rho_G(p)$ must also satisfy:

$$\rho_G(p)^2 \leq \|\log_G g_p\|_{\tilde{G}}^2 \leq d_G(p, p_0)^2 + Cd_G(p, p_0)^4,$$

for all $p \in P_0$, i.e., all p sufficiently close to p_0 .

As a corollary, we get that for any compact neighborhood $K \subset G/H$ of p_0

$$\rho_G(p) \leq C_{\text{metr}} d_G(p_0, p)$$

for all $p \in K$. We can see this by choosing $C_1^2 = 1 + C \sup_{p \in P_0} d_G(p_0, p)^2$, then for all by choosing $p \in P_0$ we have

$$\rho_G(p) \leq C_1 d_G(p_0, p).$$

Let $K \subset G/H$ be compact so that it contains P_0 . Then, on $\overline{K} \setminus P_0$ we have that both ρ_G and $d_G(p_0, \cdot)$ are strictly positive, continuous and so bounded functions. Consequently,

$$\rho_G(p) \leq \sup_{p \in K \setminus P_0} \rho_G(p) =: M < \infty,$$

and

$$d_G(p_0, p) \geq \sup_{p \in P_0} d_G(p_0, p) =: m > 0,$$

for all $p \in K \setminus P_0$, which leads to

$$\rho_G(p) \leq M \leq \frac{M}{d_G(p_0, p)} d_G(p_0, p) \leq \frac{M}{m} d_G(p_0, p).$$

for all $p \in K \setminus P_0$. Now choose $C_{\text{metr}} = \max\{C_1, C_2\}$, then we obtain the corollary. Remark that C_{metr} depends on both the parameters of the metric tensor field and the choice of K and so may become very large indeed. \square

A.2 Proof of Proposition 21

As a preliminary, we prove the following lemma.

Lemma For all $g \in G$, let $L_g : G \rightarrow G$ be the left group multiplication given by $L_g h = gh$ and let $R_g : G \rightarrow G$ be the right group multiplication given by $R_g h = hg$. Let H be a closed subgroup of G with the projection map $\pi : G \rightarrow G/H$ given by $\pi(g) = gH$.

Then, for all $h \in H$ we have the following relations for the push forwards:

1. $\pi_* \circ (R_h)_* = \pi_*$,
2. $(L_h)_* \circ \pi_* = \pi_* \circ (L_h)_*$.

Proof 1. $\pi \circ R_h = \pi$ since $ghH = gH$,
2. $L_h \circ \pi = \pi \circ L_h$ since $h(gH) = (hg)H$. \square

Now for the proof of Proposition 21. Consider the set of all exponential curves in the group whose action connects $p_0 \in G/H$ to $p \in G/H$:

$$\Gamma_{p_0, p} = \left\{ \gamma \in \text{Lip}([0, 1], G) \mid \begin{aligned} &\gamma(0) = e, \gamma(1)p_0 = p, \gamma(t+s) = \gamma(t)\gamma(s) \end{aligned} \right\}.$$

We can then restate ρ_G equivalently in terms of these curves as

$$\rho_G(p) := \inf_{g \in P} \|\log_G g\|_{\tilde{G}} = \inf_{\gamma \in \Gamma_{p_0, p}} \|\dot{\gamma}(0)\|_{\tilde{G}}$$

since for each $g \in p$ we have an exponential curve $t \mapsto \exp_G(t \log_G g)$ in $\Gamma_{p_0, p}$, and for each exponential curve γ in $\Gamma_{p_0, p}$ we have $\gamma(1) \in p$.

Let $\gamma \in \Gamma_{p_0, p}$ and let $h \in H$, then

1. $h\gamma(0)h^{-1} = heh^{-1} = e$,
2. $h\gamma(1)h^{-1}p_0 = h\gamma(1)p_0 = hp$,
3. $h\gamma(a+b)h^{-1} = h\gamma(a)\gamma(b)h^{-1} = (h\gamma(a)h^{-1})(h\gamma(b)h^{-1})$,

from which we conclude that $h\gamma(\cdot)h^{-1} \in \Gamma_{p_0, hp}$ and so there is a bijection between $\Gamma_{p_0, p}$ and $\Gamma_{p_0, hp}$ given by

$$\Gamma_{p_0, hp} = h\Gamma_{p_0, p}h^{-1}.$$

Moreover, the bijection preserves the seminorm due to the G-invariance of \mathcal{G} :

$$\begin{aligned} \|(h\gamma(\cdot)h^{-1})(0)\|_{\tilde{\mathcal{G}}} &= \|(L_h)_*(R_{h^{-1}})_*\dot{\gamma}(0)\|_{\tilde{\mathcal{G}}} \\ &= \|\pi_*(L_h)_*(R_{h^{-1}})_*\dot{\gamma}(0)\|_{\mathcal{G}} \end{aligned}$$

(using the previous lemma)

$$\begin{aligned} &= \|(L_h)_*\pi_*(R_{h^{-1}})_*\dot{\gamma}(0)\|_{\mathcal{G}} \\ &= \|(L_h)_*\pi_*\dot{\gamma}(0)\|_{\mathcal{G}} \end{aligned}$$

(using the G-invariance of \mathcal{G})

$$\begin{aligned} &= \|\pi_*\dot{\gamma}(0)\|_{\mathcal{G}} \\ &= \|\dot{\gamma}(0)\|_{\tilde{\mathcal{G}}}. \end{aligned}$$

It follows that

$$\begin{aligned} \rho_{\mathcal{G}}(p) &= \inf_{\gamma \in \Gamma_{p_0, p}} \|\dot{\gamma}(0)\|_{\tilde{\mathcal{G}}} \\ &= \inf_{\gamma \in \Gamma_{p_0, p}} \|h\dot{\gamma}(0)h^{-1}\|_{\tilde{\mathcal{G}}} \\ &= \inf_{\gamma \in \Gamma_{p_0, hp}} \|\dot{\gamma}(0)\|_{\tilde{\mathcal{G}}} \\ &= \rho_{\mathcal{G}}(hp). \end{aligned}$$

□

References

1. Welk, M., Weickert, J.: PDE evolutions for M-smoothers: from common myths to robust numerics. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 236–248. Springer (2019)
2. Fadili, J., Kutyniok, G., Peyré, G., Plonka-Hoch, G., Steidl, G.: Guest editorial: mathematics and image analysis. *J. Math. Imaging Vis.* **52**(3), 315–316 (2015)
3. Peyré, G., Péchaud, M., Keriven, R., Cohen, L.D.: Geodesic methods in computer vision and graphics. *Found. Trends. Comput. Graph. Vis.* **5**(3), 197–397 (2010)
4. Dubrovina-Karni, A., Rosman, G., Kimmel, R.: Multi-region active contours with a single level set function. *IEEE PAMI* **37**(8), 1585–1601 (2015)
5. Burger, M., Sawatzky, A., Steidl, G.: First Order Algorithms in Variational Image Processing. Springer, Cham (2016)
6. Duran, J., Moeller, M., Sbert, C., Cremers, D.: Collaborative total variation: a general framework for vectorial tv models. *SIAM SIIMS* **9**(1), 116–151 (2016)
7. Weickert, J., Grewenig, S., Schroers, C., Bruhn, A.: Cyclic schemes for PDE-based image analysis. *Int. J. Comput. Vis.* **118**(3), 275–299 (2016)
8. Sapiro, G.: Geometric Partial Differential Equations and Image Analysis. Cambridge University Press, Cambridge (2001). <https://doi.org/10.1017/CBO9780511626319>
9. Sethian, J.: Level Set Methods and Fast Marching Methods. Cambridge University Press, Cambridge (1999)
10. Weickert, J.: Theoretical foundations of anisotropic diffusion in image processing. *Comput. Suppl.* **11**, 221–236 (1996)
11. Morel, J., Solimini, S.: Variational Methods in Image Segmentation: With Seven Image Processing Experiments. Progress in Non-linear Differential Equations and Their Applications. Birkhäuser, Basel (1995)
12. Duits, R., Burgeth, B.: Scale spaces on Lie groups. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 300–312. Springer (2007)
13. Cohen, T.S., Welling, M.: Group equivariant convolutional networks. In: International of Conference on Machine Learning, pp. 2990–2999 (2016)
14. Dieleman, S., De Fauw, J., Kavukcuoglu, K.: Exploiting cyclic symmetry in convolutional neural networks (2016). arXiv preprint [arXiv:1602.02660](https://arxiv.org/abs/1602.02660)
15. Dieleman, S., Willett, K.W., Dambre, J.: Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Mon. Not. R. Astron. Soc.* **450**(2), 1441–1459 (2015)
16. Winkels, M., Cohen, T.S.: 3D G-CNNs for pulmonary nodule detection (2018). arXiv preprint [arXiv:1804.04656](https://arxiv.org/abs/1804.04656)
17. Worrall, D., Brostow, G.: Cubenet: Equivariance to 3D rotation and translation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 567–584 (2018)
18. Bekkers, E.J., Loog, M., ter Haar Romeny, B.M., Duits, R.: Template matching via densities on the roto-translation group. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(2), 452–466 (2017)
19. Oyallon, E., Mallat, S.: Deep roto-translation scattering for object classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2865–2873 (2015)
20. Bekkers, E.J., Lafarge, M.W., Veta, M., Eppenhof, K.A., Pluim, J.P., Duits, R.: Roto-translation covariant convolutional networks for medical image analysis. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 440–448. Springer (2018). [arXiv:1804.03393](https://arxiv.org/abs/1804.03393)
21. Weiler, M., Hamprecht, F.A., Storath, M.: Learning steerable filters for rotation equivariant cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 849–858 (2018)
22. Cohen, T.S., Geiger, M., Weiler, M.: A general theory of equivariant cnns on homogeneous spaces. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
23. Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J.: Harmonic networks: deep translation and rotation equivariance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5028–5037 (2017)
24. Kondor, R., Trivedi, S.: On the generalization of equivariance and convolution in neural networks to the action of compact groups.

- In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 2747–2755. PMLR, Stockholm, Sweden (2018). <http://proceedings.mlr.press/v80/kondor18a.html>
25. Esteves, C., Allen-Blanchette, C., Makadia, A., Daniilidis, K.: Learning SO(3) equivariant representations with spherical CNNs. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 52–68 (2018)
 26. Akian, M., Quadrat, J.P., Viot, M.: Bellman processes. In: 11th International Conference on Analysis and Optimization of Systems Discrete Event Systems, pp. 302–311. Springer (1994)
 27. Burgeth, B., Welk, M., Feddern, C., Weickert, J.: Morphological operations on matrix-valued images. In: European Conference on Computer Vision, pp. 155–167. Springer (2004)
 28. Duits, R., Haije, T.D., Creusen, E., Ghosh, A.: Morphological and linear scale spaces for fiber enhancement in DW-MRI. *J. Math. Imaging Vis.* **46**(3), 326–368 (2013)
 29. Bekkers, E.J., Duits, R., Mashtakov, A., Sanguinetti, G.R.: A PDE approach to data-driven sub-Riemannian geodesics in SE(2). *SIAM J. Imaging Sci.* **8**(4), 2740–2770 (2015)
 30. Haije, T.C.D., Duits, R., Tax, C.M.: Sharpening fibers in diffusion weighted MRI via erosion. In: Visualization and Processing of Tensors and Higher Order Descriptors for Multi-valued Data, pp. 97–126. Springer (2014)
 31. Duits, R., Smets, B., Bekkers, E.J., Portegies, J.M.: Equivariant deep learning via morphological and linear scale space PDEs on the space of positions and orientations. *LNCS* **12679**, 27–39 (2021)
 32. Citti, G., Franceschiello, B., Sanguinetti, G., Sarti, A.: Sub-Riemannian mean curvature flow for image processing. *SIAM J. Imaging Sci.* **9**(1), 212–237 (2016)
 33. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
 34. Chambolle, A., Pock, T.: Total roto-translational variation. *Numer. Math.* **142**(3), 611–666 (2019). <https://doi.org/10.1007/s00211-019-01026-w>
 35. Smets, B.M., Portegies, J., St-Onge, E., Duits, R.: Total variation and mean curvature PDEs on the homogeneous space of positions and orientations. *J. Math. Imaging Vis.* **63**(2), 237–262 (2021)
 36. Duits, R., Duits, M., van Almsick, M., ter Haar Romeny, B.: Invertible orientation scores as an application of generalized wavelet theory. *Pattern Recognit. Image Anal.* **17**(1), 42–75 (2007)
 37. Janssen, M.H.J., Janssen, A.J.E.M., Bekkers, E.J., Bescós, J.O., Duits, R.: Design and processing of invertible orientation scores of 3D images. *J. Math. Imaging Vis.* **60**(9), 1427–1458 (2018). <https://doi.org/10.1007/s10851-018-0806-0>
 38. Franceschiello, B., Mashtakov, A., Citti, G., Sarti, A.: Geometrical optical illusion via sub-Riemannian geodesics in the roto-translation group. *Differ. Geom. Appl.* **65**, 55–77 (2019)
 39. Citti, G., Sarti, A.: A cortical based model of perceptual completion in the roto-translation space. *J. Math. Imaging Vis.* **24**(3), 307–326 (2006)
 40. Duits, R., Franken, E.M.: 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). *Q. Appl. Math.* **68**, 255–292 (2010)
 41. Duits, R., Franken, E.M.: 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. *Q. Appl. Math.* **68**, 293–331 (2010)
 42. Zhang, J., Duits, R., ter Haar Romeny, B., Sanguinetti, G.: Numerical approaches for linear left-invariant diffusions on SE(2), their comparisons to exact solutions, and their applications in retinal imaging. *Numer. Math. Theory Methods Appl.* **9**(1), 1–50 (2016)
 43. Boscain, U., Chertovskih, R.A., Gauthier, J.P., Remizov, A.O.: Hypoelliptic diffusion and human vision: a semidiscrete new twist. *SIAM J. Imaging Sci.* **7**(2), 669–695 (2014)
 44. Bertalmio, M., Calatroni, L., Franceschi, V., Franceschiello, B., Prandi, D.: A cortical-inspired model for orientation-dependent contrast perception: a link with Wilson-Cowan equations. In: Lellmann, J., Burger, M., Modersitzki, J. (eds.) Scale Space and Variational Methods in Computer Vision, pp. 472–484. Springer, Cham (2019)
 45. Duits, R., Fuehr, H., Janssen, B., Florack, L., van Assen, H.: Evolution equations on Gabor transforms and their applications. *ACHA* **35**(3), 483–526 (2013)
 46. Barbieri, D., Citti, G., Cocci, G., Sarti, A.: A cortical-inspired geometry for contour perception and motion integration. *J. Math. Imaging Vis.* **49**(3), 511–529 (2014)
 47. Petitot, J.: The neurogeometry of pinwheels as a sub-Riemannian contact structure. *J. Physiol. Paris* **97**, 265–309 (2003)
 48. Felsberg, M., Forssen, P.E., Scharr, H.: Channel smoothing: efficient robust smoothing of low-level signal features. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**, 209–222 (2006)
 49. Savadjiev, P., Strijkers, G., Bakermans, A., Piuz, E., Zucker, S., Siddiqi, K.: Heart wall myofibers are arranged in minimal surfaces to optimize organ function. *PNAS* **109**(24), 9248–9253 (2012)
 50. Duits, R., Bekkers, E., Mashtakov, A.: Fourier transform on the homogeneous space of 3D positions and orientations for exact solutions to linear PDEs. *Entropy* **21**(1), 38 (2019)
 51. Momayyez-Siahkal, P., Siddiqi, K.: 3D stochastic completion fields for fiber tractography. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009). <https://doi.org/10.1109/CVPRW.2009.5204044>
 52. Duits, R., Führ, H., Janssen, B., Bruurmijn, M., Florack, L., van Assen, H.: Evolution equations on Gabor transforms and their applications. *Appl. Comput. Harmon. Anal.* **35**(3), 483–526 (2013)
 53. Boscain, U., Prandi, D., Sacchelli, L., Turco, G.: A bio-inspired geometric model for sound reconstruction. *J. Math. Neurosci.* **11**(1), 1–18 (2021)
 54. Baspinar, E., Citti, G., Sarti, A.: A geometric model of multi-scale orientation preference maps via Gabor functions. *J. Math. Imaging Vis.* **60**(6), 900–912 (2018)
 55. Bekkers, E.J.: B-Spline CNNs on Lie groups. In: International Conference on Learning Representations (2019)
 56. Finzi, M., Stanton, S., Izmailov, P., Wilson, A.G.: Generalizing convolutional neural networks for equivariance to Lie groups on arbitrary continuous data. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 3165–3176. PMLR (2020). <http://proceedings.mlr.press/v119/finzi20a.html>
 57. Weiler, M., Cesa, G.: General E(2)-equivariant steerable CNNs. In: Advances in Neural Information Processing Systems, pp. 14334–14345 (2019)
 58. Chirikjian, G.S., Kyatkin, A.B.: An operational calculus for the Euclidean motion group with applications in robotics and polymer science. *J. Fourier Anal. Appl.* **6**(6), 583–606 (2000)
 59. Franken, E., van Almsick, M., Rongen, P., Florack, L., ter Haar Romeny, B.: An efficient method for tensor voting using steerable filters. In: European Conference on Computer Vision, pp. 228–240. Springer (2006)
 60. Reiser, M.: Group integration techniques in pattern analysis: a kernel view. Ph.D. thesis, Albert-Ludwigs-Universität Freiburg (2008)
 61. Ali, S.T., Antoine, J.P., Gazeau, J.P., et al.: Coherent States, Wavelets and Their Generalizations, vol. 1. Springer, Berlin (2000)

62. Sifre, L., Mallat, S.: Rigid-motion scattering for texture classification (2014). arXiv preprint [arXiv:1403.1687](https://arxiv.org/abs/1403.1687)
63. Finzi, M., Bondesan, R., Welling, M.: Probabilistic numeric convolutional neural networks (2020). arXiv preprint [arXiv:2010.10876](https://arxiv.org/abs/2010.10876)
64. Montobbio, N.: A metric model of the visual cortex. Ph.D. thesis, Università di Bologna - Sorbonne Université (2019)
65. Montobbio, N., Bonnasse-Gahot, L., Citti, G., Sarti, A.: KerC-NNs: biologically inspired lateral connections for classification of corrupted images (2019). arXiv preprint [arXiv:1910.08336](https://arxiv.org/abs/1910.08336)
66. Weinan, E.: A proposal on machine learning via dynamical systems. *Commun. Math. Stat.* **5**(1), 1–11 (2017)
67. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
68. Lu, Y., Zhong, A., Li, Q., Dong, B.: Beyond finite layer neural networks: bridging deep architectures and numerical differential equations (2017). arXiv preprint [arXiv:1710.10121](https://arxiv.org/abs/1710.10121)
69. Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: *Advances in Neural Information Processing Systems*, pp. 6571–6583 (2018)
70. Chen, Y., Yu, W., Pock, T.: On learning optimized reaction diffusion processes for effective image restoration. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5261–5269 (2015)
71. Long, Z., Lu, Y., Ma, X., Dong, B.: PDE-net: Learning PDEs from data (2017). arXiv preprint [arXiv:1710.09668](https://arxiv.org/abs/1710.09668)
72. Ruthotto, L., Haber, E.: Deep neural networks motivated by partial differential equations. *J. Math. Imaging Vis.* **62**(3), 352–364 (2020)
73. Shen, Z., He, L., Lin, Z., Ma, J.: Pdo-econvs: partial differential operator based equivariant convolutions. In: *International Conference on Machine Learning*, pp. 8697–8706. PMLR (2020)
74. Weiler, M., Forré, P., Verlinde, E., Welling, M.: Coordinate independent convolutional networks—*isometry and gauge equivariant convolutions on Riemannian manifolds* (2021). arXiv preprint [arXiv:2106.06020](https://arxiv.org/abs/2106.06020)
75. Jenner, E., Weiler, M.: Steerable partial differential operators for equivariant neural networks (2021). arXiv preprint [arXiv:2106.10163](https://arxiv.org/abs/2106.10163)
76. Haber, E., Ruthotto, L.: Stable architectures for deep neural networks. *Inverse Prob.* **34**(1), 014004 (2017)
77. Alt, T., Peter, P., Weickert, J., Schrader, K.: Translating numerical concepts for PDEs into neural architectures (2021). arXiv preprint [arXiv:2103.15419](https://arxiv.org/abs/2103.15419)
78. Koda, T.: An introduction to the geometry of homogeneous spaces. In: *Proceedings of the Thirteenth International Workshop on Differential Geometry*, vol. 13, pp. 121–144 (2009)
79. Lee, J.M., Chow, B., Chu, S.C., Glickenstein, D., Guenther, C., Isenberg, J., Ivey, T., Knopf, D., Lu, P., Luo, F., et al.: Manifolds and differential geometry. *Topology* **643**, 658 (2009)
80. Smets, B.: Geometric image denoising and machine learning. Master thesis, TU Eindhoven (2019). <https://bmnsnets.com/publication/snets2019msc/>
81. Arendt, W., Bukhvalov, A.V.: Integral representations of resolvents and semigroups. *Forum Math.* **6**, 111–136 (1994)
82. Führ, H.: *Abstract Harmonic Analysis of Continuous Wavelet Transforms*. 1863. Springer, Berlin (2005)
83. Duits, R., Meesters, S.P., Mirebeau, J.M., Portegies, J.M.: Optimal paths for variants of the 2D and 3D Reeds–Shepp car with applications in image analysis. *J. Math. Imaging Vis.* **60**, 1–33 (2018)
84. Portegies, J., Sanguinetti, G., Meesters, S., Duits, R.: 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*, pp. 40–52. Springer (2015)
85. Ter Elst, A., Robinson, D.W.: Weighted subcoercive operators on Lie groups. *J. Funct. Anal.* **157**(1), 88–163 (1998)
86. Maheux, P.: Estimations du noyau de la chaleur sur les espaces homogènes. *ESAIM Control Optim. Calc. Var.* **8**(1), 65–96 (1998)
87. Grigor'yan, A., Hu, J., Lau, K.S.: Heat kernels on metric spaces with doubling measure. In: *Fractal Geometry and Stochastics IV*, pp. 3–44. Springer (2009)
88. Yosida, K.: *Functional Analysis*. Springer, Berlin (1968)
89. Evans, L.C.: *Partial Differential Equations*, vol. 19. American Mathematical Society, Providence (2010)
90. Balogh, Z.M., Engulatov, A., Hunziker, L., Maasalo, O.E.: Functional inequalities and Hamilton–Jacobi equations in geodesic spaces. *Potential Anal.* **36**(2), 317–337 (2012)
91. Dragoni, F.: Metric Hopf–Lax formula with semicontinuous data. *Discrete Contin. Dyn. Syst. A* **17**(4), 713 (2007)
92. Azagra, D., Ferrera, J., López-Mesas, F.: Nonsmooth analysis and Hamilton–Jacobi equations on Riemannian manifolds. *J. Funct. Anal.* **220**(2), 304–361 (2005)
93. Rund, H.: *The Hamilton–Jacobi Theory in the Calculus of Variations: Its Role in Mathematics and Physics*. Krieger Publishing Company, Malabar (1966)
94. Manfredi, J.J., Stroffolini, B.: A version of the Hopf–Lax formula in the Heisenberg group. *Commun. Partial Differ. Equ.* **27**, 1139–1159 (2002)
95. Mirebeau, J.M., Portegies, J.: Hamiltonian fast marching: a numerical solver for anisotropic and non-holonomic eikonal PDEs. *Image Process. On Line* **9**, 47–93 (2019)
96. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., Van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. *IEEE Trans. Med. Imaging* **23**(4), 501–509 (2004)
97. Baweja, C.: RotNIST (2018). <https://github.com/ChaitanyaBaweja/RotNIST>
98. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: an imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems*, pp. 8024–8035 (2019). www.pytorch.org
99. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014). arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
100. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1**(4), 541–551 (1989)
101. Lee, J.: *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics, 2nd edn. Springer, Berlin (2013). https://doi.org/10.1007/978-1-4419-9982-5_1

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

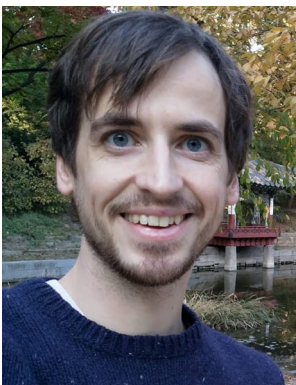


Bart M. N. Smets received his M.Sc. in applied mathematics at the TU/e in 2019. Now he is pursuing his Ph.D. under the supervision of Remco Duits and Jim Portegies, also at the TU/e. He created and teaches a course on the mathematics of neural networks. As part of his Ph.D. research, he is investigating how PDE-based image analysis techniques can be utilized to improve and gain a better understanding of how neural networks work. He is interested in all the ways that mathe-

matics can be used to understand and enhance deep learning.



Jim Portegies obtained his Ph.D. from the Courant Institute of Mathematical Sciences in 2014. Afterward, he was a postdoc at the Max Planck Institute for Mathematics in the Sciences in Leipzig. Since 2016, he is an assistant professor in mathematics in the Applied Analysis group at the Eindhoven University of Technology. Jim Portegies is a member of the Dutch Young Academy.



Erik J. Bekkers is an assistant professor in Geometric Deep Learning in the Machine Learning Lab of the University of Amsterdam. He did a post-doc in applied differential geometry at the Applied Mathematics Dept. of Technical University Eindhoven. In his Ph.D. thesis (cum laude, Biomedical Engineering, TU/e), he developed medical image analysis algorithms based on sub-Riemannian geometry in the Lie group $SE(2)$ using mathematical principles that underlie models of human visual

perception. His interest is in generalizing group convolutional networks and improving computational and representational efficiency through sparse (graphs) and adaptive learning mechanisms. Erik is a recipient of a MICCAI Young Scientist Award 2018, Philips Impact Award (MIDL 2018) and a personal VENI research grant awarded by the Dutch Research Council.



Remco Duits received his M.Sc. degree (cum laude) in Mathematics in 2001 at the TU/e, where he also received his Ph.D. (cum laude) at the Department of Biomedical Engineering. Now he is associate professor at the Department of Applied Mathematics & Computer Science. He leads a research group ‘Geometric Deep Learning and Differential Geometry’ within CASA (Center for Analysis, Scientific Computing and Applications). He has organized various large conferences

(6) and colloquia, coordinated the TU/e Mathematics department’s research assessment (2015–2021) and serves in program committees of international conferences (e.g., SSVI, GSI). He is editor for JMIV, and he has led a special issue on ‘Differential Geometry and Orientation Analysis in Image Processing.’ He has been scientific reviewer for many (inter)national research proposals, conference proceedings, books and a wide range of journals. His research interests subsume analysis, differential geometry, Lie group theory, PDEs, mathematical and medical image analysis. He has received two prestigious personal grants: an ERC-StG grant (2014–2019) ‘Lie Group Analysis for Medical Image Processing’ and an NWO-VICI grant (2021–2026) ‘Geometric Learning for Image Analysis’ and acted as supervisor/applicant in other EU-projects (EU-MC, FP7) and TU/e (High Potential, EAISI) projects.