Analysis and Geometric Interpretation of PDE-G-CNNs

Bon, Daan L.J.

Award date:
2022

Eindhoven University of Technology

Analysis and Geometric Interpretation of PDE-G-CNNs

Author: D.L.J. Bon

Supervisor: R. Duits

Co-supervisor: G. Bellaard

July 21, 2022

Contents

1 Introduction

Convolutional Neural Networks (CNNs) have been extensively researched and used in recent years. Ever since LeCun et al. showed how successful CNNs can be on real world image classification problems [1], the interest for CNNs has grown. This became especially true with the increase in hardware performance. However, CNNs suffer from a few flaws. For one, CNNs (and artificial neural networks in general) are often described as "black boxes": they lack interpretability in the sense that one does not know how the network reaches a conclusion. Furthermore, CNNs usually do not respect the inherent structure of the data they are trained on. Think of an image segmentation task. It is logical that when we rotate an image, its segmentation should also be rotated, as the rotation does not change the objects depicted. However, for CNNs, there is no guarantee that this actually happens.

These "invariances" of the data e.g., rotating an image does not change what it is depicting, are called symmetries. Requiring that the network respects certain symmetries is called equivariance of the network with respect to these symmetries. One possible way to achieve this equivariance is through data augmentation: add explicitly transformed data to the existing data set so that a networks can learn the symmetries. An alternative would be to alter the structure of the network such that invariance is guaranteed. Not only does this prevent the need for training on a (much) larger augmented data set, it also allows the network to "focus" its learning on the essence of the task, without also having to learn the symmetries involved. One way to include specific symmetries in CNNs is described in [2], where the authors introduce *Group Equivariant Convolutional Neural Networks* (G-CNNs). The networks presented are translation, rotation and reflection equivariant, as opposed to only the translation equivariance found in regular CNNs. A sample of papers exploring this topic further is [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23].

In [24], a PDE-based framework for CNNs is presented that generalizes G-CNNs. The authors present PDE-based Group CNNs (PDE-G-CNNs), in which the usual non-linearities that are present in CNNs, for example ReLU activation functions and max-pooling, are replaced by solvers for specific non-linear evolution equations. This idea is depicted in Figure 1, where we compare a traditional CNN layer with a PDE-G-CNN layer. Figure 2 shows more details about the evolution equation. There are several advantages to using these PDE-based layers. For one, the PDEs used in PDE-G-CNNs come from the world of geometric image analysis, and thus their effects are geometrically interpretable. They also generalize the usual operations that are used in CNNs, ReLUs and max-pooling. Lastly, the PDEs are not restricted to data living on $\mathbb{R}^2$, and can be formulated on so called *homogeneous spaces*. This allows for a natural way to require the network to be equivariant with respect to desired symmetries, based on the homogeneous space used. The main use cases of PDE-G-CNNs are image processing tasks that are inherently symmetric to translations and rotations, of which image classification is one. In order to achieve equivariance of the network, we will lift images to the *space of positions and orientations* $\mathbb{M}_2$, and do the processing there using PDE-G-CNNs. The overall architecture for such a network can be seen in Figure 3, where the task is to extract the continuous line from the image.

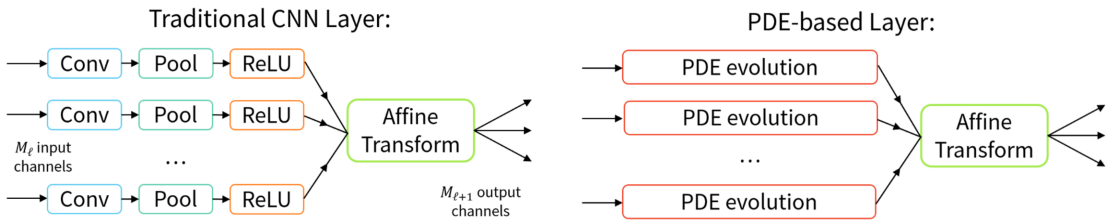


Figure 1: The difference between a traditional CNN layer and a PDE-G-CNN layer. Note that both still make use of linear combinations between layers. Each PDE has several parameters that can be trained by the network. Taken from [24] with permission.

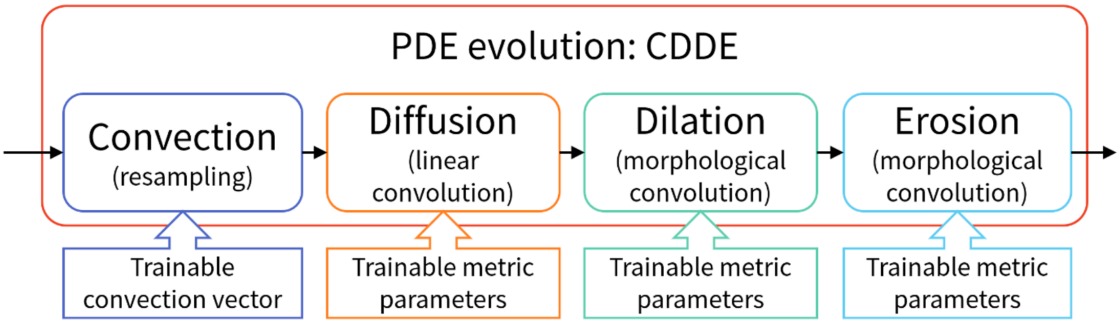


Figure 2: The used evolution PDE in more detail. It contains four terms, corresponding to the terms in the figure. This PDE is solved through operator splitting, which involves solving for each of the terms separately. We will primarily be concerned with convection, dilation and erosion is this report. Taken from [24] with permission.

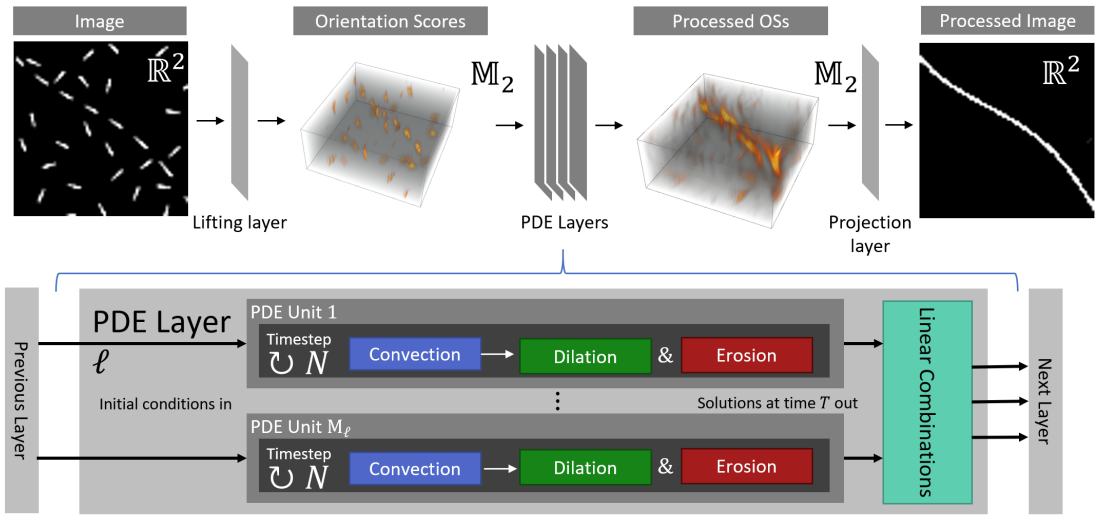


Figure 3: The overall architecture for a PDE-G-CNN performing image classification on the Line data set, which we will introduce later in this report. All of the processing using PDE layers happens on the homogeneous space $\mathbb{M}_2$.

Lifting an image to $\mathbb{M}_2$, which as a set is given by $\mathbb{R}^2 \times S^1$, means that for each position we determine locally how much the image is aligned with a specific orientation. The result of this is called the *orientation score* of the image. Since this score depends on position and orientation, it naturally lives in $\mathbb{M}_2$. See Figure 4 for an visualization of a lifted image. Next to allowing for equivariant processing using PDE-G-CNNs, working in $\mathbb{M}_2$ instead of in $\mathbb{R}^2$ also has the advantage that the networks have explicit access to more information about the image in the form of this orientation score. It has been shown that doing processing on $\mathbb{M}_2$, or its higher dimensional analogues, has advantages over regular image processing [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41].

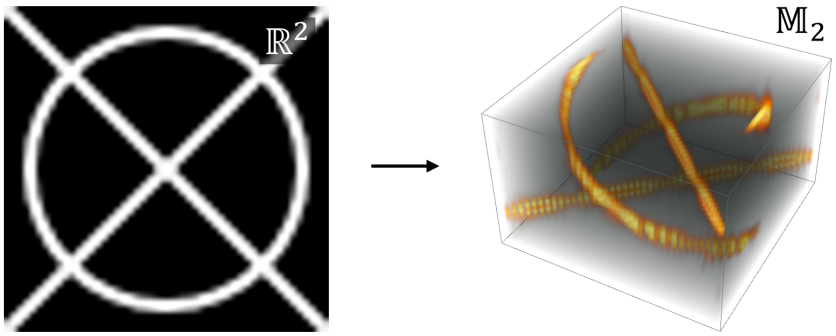


Figure 4: An example of an image together with its orientation score. We can see that the 2d image is transformed in a 3d object, the orientation score. These orientation scores are what the main part of the PDE-G-CNNs is processing, see Figure 3. Notice that the lines that are crossing in the left image are disentangled in the orientation score.

The inspiration for using orientation scores in image analysis comes from biology. The Nobel laureates Hubel and Wiesel discovered that certain visual cells in the striate cortex of cats have a directional preference, firing more when the viewed image contains the specific orientation that is preferred [42]. It has turned out that the majority of the visual cortex exhibits such a preference for specific orientations [43].

Moreover, it turns out that neurons that fire for a specific orientation excite neighboring neurons with an orientation that "aligns". Petitot proposed a model for the distribution of the orientation preference and this exciting of neighbors based on sub-Riemannian geometry in [36]. In this paper he also relates the phenomenon of preference of aligned orientations to the concept of *association fields* [44], which model how a specific local orientation puts expectations on surrounding local orientations in human vision. Such activations of local orientations around a given local orientation have also been measured [43] on the visual cortex of mammalians. Figure 5 shows what these measured association fields look like. In [28] the authors propose an model for these association fields using (horizontal) exponential curves on $SE(2)$.

These association field lines are more closely approximated by (projected) sub-Riemannian geodesics in the space of positions and orientations $SE(2)$ [45]. For now, we can think of $SE(2)$ as the same thing as $\mathbb{M}_2$. Moreover, [46] showed that certain anisotropic Riemannian geodesics converge to these sub-Riemannian geodesics for increasing anisotropy of the Riemannian metric. [47] showed that this convergence is in fact very rapid, so that in practical applications anisotropic Riemannian geometry on $SE(2)$ is a good model for association fields. In Figure 6 we can see how a Riemannian metric on $SE(2)$ gives rise to association fields.

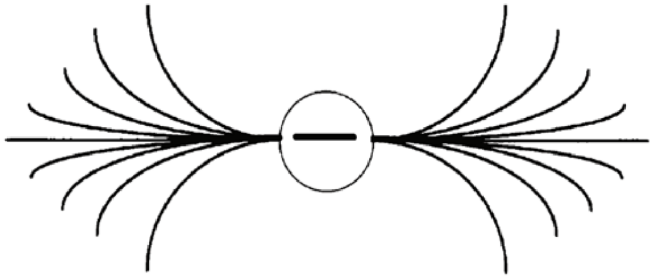


Figure 5: Association field lines from neurogeometry [36, 44] are empirical line propagation models for contour perception. Finding a local orientation at the center (within the circle) imposes visual expectations of local orientations $p = (x, y, \theta)$ in the surroundings along the depicted field lines. Such field lines are either modeled by projected (horizontal) exponential curves (co-circularity [28]) or by cusp-free projected sub-Riemannian geodesics in $\mathbb{M}_2$ [45]. There is something to say for both models: 'shortest' curves in the sub-Riemannian manifold are geodesics whereas the 'straight' curves w.r.t. Cartan connection are exponential curves [48, Thm.1]. See Fig. 6 where sub-Riemannian balls on $\mathbb{M}_2$ and projected horizontal exponential curves are depicted.

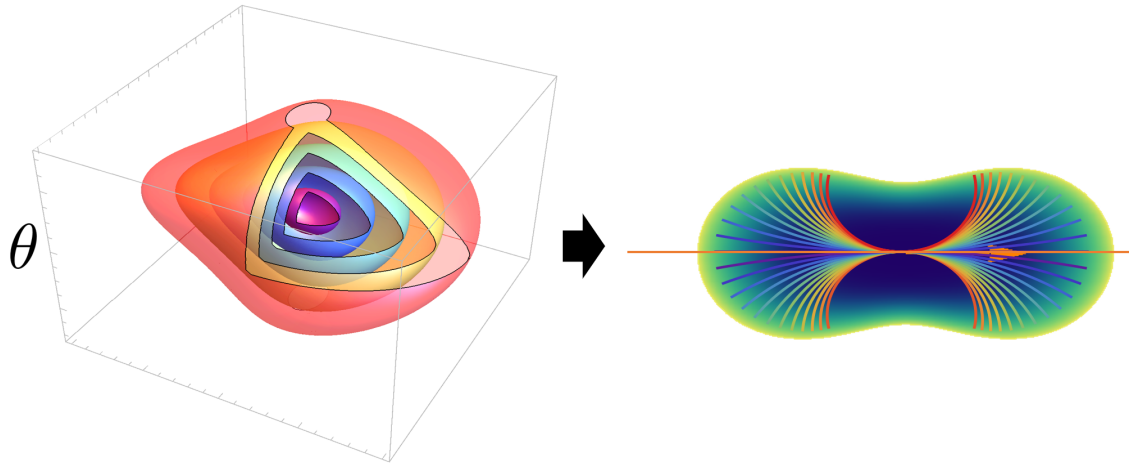
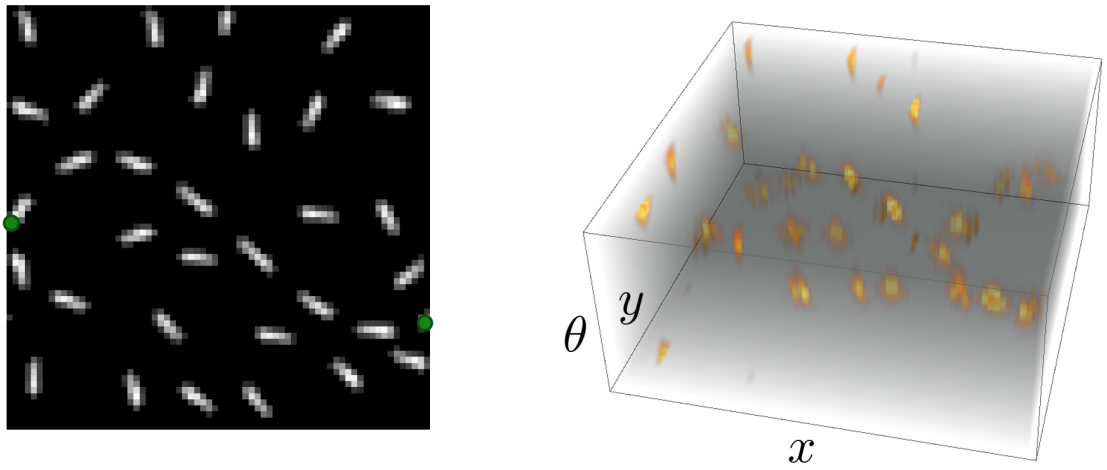
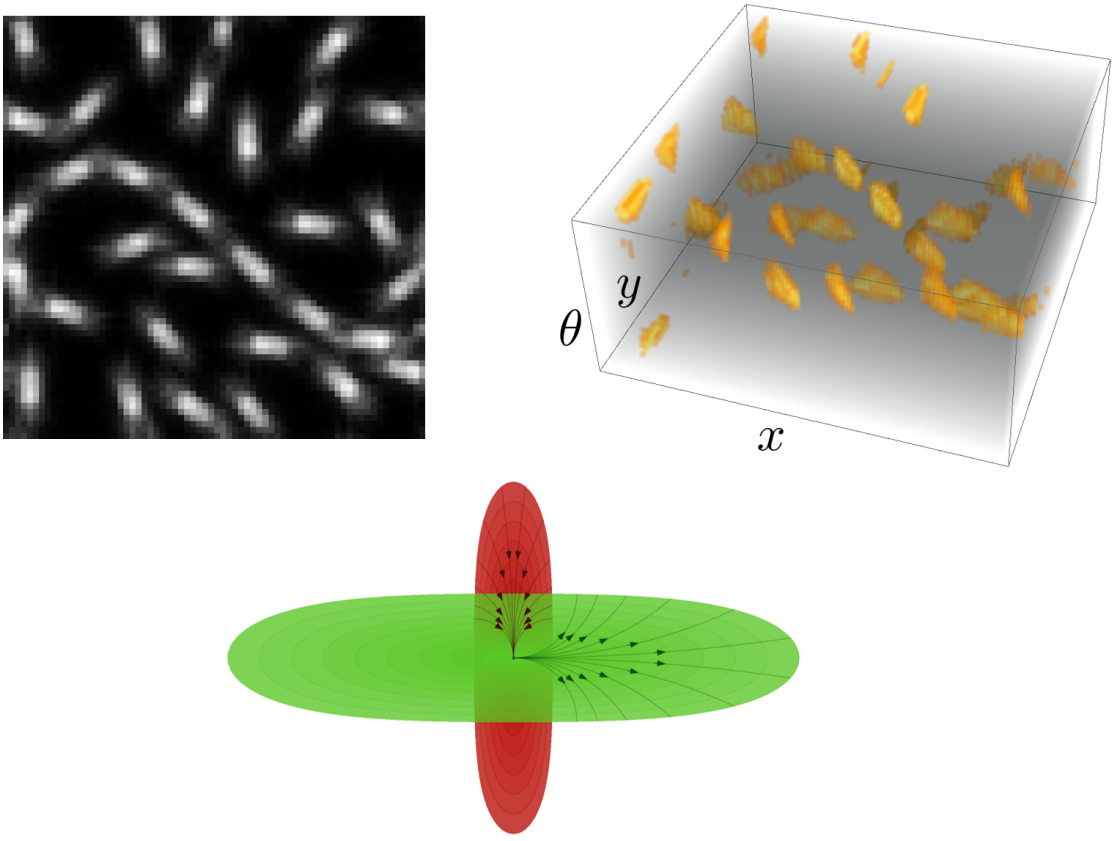


Figure 6: On the left we see level sets of a Riemannian distance in $\mathbb{M}_2$. On the right we have plotted a projected level set, together with horizontal exponential curves, which are a model for the association fields (See Fig. 5) in the visual system of mammalians, cf. [36, 44, 43, 42]

In PDE-G-CNNs we train such Riemannian geometries for the processing of the image, recall Figure 2. In other words we train association fields. The Riemannian metrics that are trained are part of two effects in the PDE evolution, dilation (related to "excitation" from neurogeometry) and erosion (related to "inhibition"). Dilation and erosion are related to the gray-scale morphology operations by the same names [49]. These operations mimic the effect of excitation and inhibition of neighboring positions and orientations, depending on whether they are aligned or not. This alignment now depends on the trained metric in the network. The hope is that PDE-G-CNNs are able to combine these learned association fields to higher level geometric units, called *geometric units describing shapes* (geons) [50]. Our leading example of this will be line-detection in images, like in Figure 7a. We see that applying dilation and erosion once has the ability to enhance lines present, see Figure 7b. The general idea of this is depicted in Figure 8, where elementary geometrical units, the association fields, combine into a global line detector. Examples of this can be seen in trained PDE-G-CNNs in Figure 9, and will be explored later on in this report.



(a) On the left we have the original image, which has a line running through it. We can see this line by connecting the two green dots, following the local orientation of the line segments. On the right we have the orientation score of the image on the left.



(b) On the right we now see the orientation score of the image in (a), after having done dilation and erosion. On the left we see this orientation score projected down to $\mathbb{R}^2$. Note how this dilation and erosion in $\mathbb{M}_2$ allows us to find the outline of the line in $\mathbb{R}^2$. The Riemannian metrics (association fields) we used to apply dilation and erosion are visualized below as the green and red kernels. This visualization will be explained in Section 5.

Figure 7: An example of how dilation and erosion in $\mathbb{M}_2$ can lead to line completion in images. In PDE-G-CNNs, these operations are done by all channels in each layer.

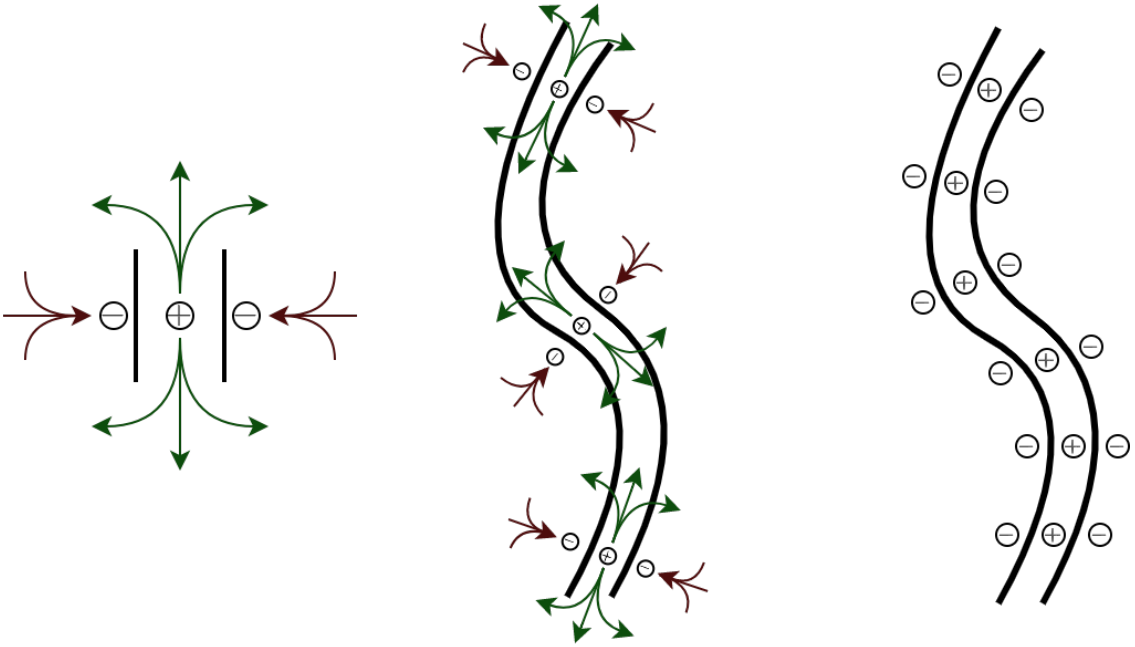


Figure 8: An example of how elementary geometric neurons combine to form geometric units describing shape (geons). On the left we see an association field, responding to a certain local orientation, i.e. a point (x, y, θ); here (x, y) is the center position and $\theta = \frac{\pi}{2}$ the orientation. In the middle we see how several of these basic units (the association fields) might interact in order to get a curved line filter, as on the right. In green we depict 'excitation' (line extrapolation) that we will model later by dilation PDEs on $\mathbb{M}_2$. In red we see 'inhibition' (line sharpening) that we will model later by erosion PDEs on $\mathbb{M}_2$.

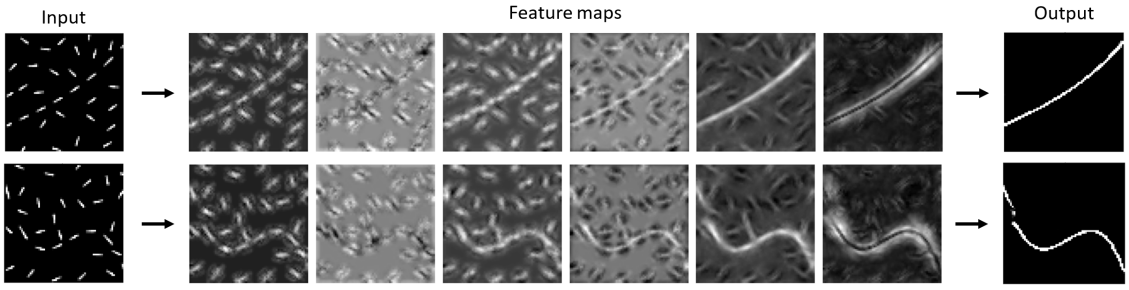


Figure 9: Showing the principle of Figure 8 in a real application of PDE-G-CNNs (where we depict spatially projected feature maps computed on each orientation score node in the PDE-G-CNN network), line detection in an image. This will be explained in greater detail in Section 5.

1.1 Research Objectives

PDE-G-CNNs seem to have many advantages over regular CNNs: the performance is better despite serious reductions of network complexity, fewer parameters and less training data are needed (as recent experiments show), and they are equivariant to more symmetries than standard CNNs [24].

However, there are still several pertinent questions that we want to address. Next we list them as the three main objectives of this master project, with references to the answers further on in the report in blue.

A) **Improve the Assessment of the Quality of Analytic Approximations for PDE-G-CNNs**.
The solver for the evolution PDE uses certain approximations, specifically for the dilation and erosion part. The approximations are motivated by a rough asymptotic expansion [24, App.A] and supported with a few numerical verifications in [24, Fig.9] with limited spatial anisotropy. Since in the sub-Riemannian case (where spatial anisotropy tends to infinity) one must resort to different analytic approximations [29, ch.5.4], we know that there are ranges of parameter settings where the approximations breakdown. There are three sub-goals that we tackle in our assessment:

1. Identify the region where the analytic kernel approximations of PDE-G-CNNs in [24, 51] are guaranteed to be sufficiently accurate.
   Remark (4.13)
2. Check whether the trained parameters indeed satisfy the requirements under 1.
   Section (5.2)
3. Provide sharper bounds and more accurate asymptotic expansions than in [24, Lemma 5.7,App.A].
   Theorem (4.8) and Theorem (4.12), and Theorem 4.9

B) **Visualize the PDE-G-CNN network-architecture (and its trained analytic kernels).**
We want to do this with the following sub-goals in mind for the visualizations:

1. Visualize the separate modules in a PDE-G-CNNs layer (dilation, erosion and convection) in a geometrically interpretable way.
   Section 5.3, Figure 29
2. Visualize the full network in such a way that allows us to confirm or deny geometric expectations (from geometric processing on orientation scores).
   Section 5.3, Figure 30
3. We can use the visualizations in such a way to identify essential parts of the network.
   Section 5.3, Figure 30

C) **Study the functioning and application of (sub-)Riemannian PDE-G-CNN models on test data.**
Here we address two sub-goals:

1. Compare the classification performances of (Riemannian and sub-Riemannian) PDE-G-CNNs and CNNs.
   Section 5.2, Figure 21 and Section 5.4, Figure 31
2. A PDE-G-CNN essentially applies an association field in each connection between two nodes (orientation scores) in the network (recall Figure 8). When we iterate applying association field on association field can we see the creation of geons in the feature maps of test images? Is it indeed more intuitive than with a CNN?
   Section 5.2, Figures 23,24, 25 and Section 5.4, Figures 32, 33

1.2 Structure of the Report

This report has the following structure. Section 2 contains the theoretical background for the remainder of this report. This primarily consists of some Riemannian geometry and Lie group theory. Section 3 contains a more elaborate introduction to PDE-G-CNNs, where we go into more detail about how they work. Section 4 contains the new results about the quality of the approximate distances used in the PDE solver in PDE-G-CNNs. Section 5 contains new visualizations and geometric interpretations of PDE-G-CNNs, partly based on the new results from the previous section. Lastly, Section 6 contains the conclusion and future research directions.

All of the code used for this project can be found at `https://github.com/d-bon/master_project`. Parts of this report are being adapted into a paper for publishing.

2 Preliminary Theory

Since PDE-G-CNNs make use of Riemannian geometry and some Lie group and homogeneous space theory, we will start with a small introduction to these subjects. We will present some basic results, and where relevant provide a proof of the presented statements. We will introduce the basic concepts of Riemannian geometry, including metric tensors, distances, geodesics, connections, the curvature tensor and end with normal coordinates. For Lie groups we introduce left-invariant vector fields, the Lie algebra, the exponential map, homogeneous spaces of Lie groups and we show how Riemannian geometry on Lie groups can interact with the additional structure. We end with introducing the spaces of positions and orientations $\mathbb{M}_d$, with particular details for the two-dimensional case.

For a general introduction to the subject of Riemannian geometry see for example [52] and [53]. For a general introduction to Lie groups see for example [54] and several chapters in [55]. We assume the reader is familiar with the basic concepts of differential geometry, such as manifolds, tangent spaces, functions between manifolds and their derivatives. For a general introduction to this subject, see for example [55].

2.1 Riemannian Geometry

In the remainder of this section, we let M denote a smooth manifold of dimension d, and $p \in M$ is some arbitrary point on M. We will denote the space of smooth vector fields on M by $\mathfrak{X}(M)$. We will make use of the Einstein summation convention.

We would like to have some concept of geometry on M, i.e. have some concept of distances between two points on the manifold. This is done by defining an inner product on each tangent space.

Definition 2.1 (Riemannian metric)**.** A Riemannian metric $\mathcal{G}$ is a smooth covariant 2-tensor field on M, which when restricted to $p \in M$ is an inner product on T_pM. A Riemannian manifold is a smooth manifold M together with a Riemannian metric $\mathcal{G}$.

This already gives us a concept of geometry on a fixed tangent space. If $X, Y \in T_pM$, we will use the notation $\mathcal{G}(X, Y)$ to denote the inner product between X and Y, or $\mathcal{G}_p(X, Y)$ if we want to stress the point in which we are looking. We let $||X||_{\mathcal{G}} = \sqrt{\mathcal{G}(X, X)}$ denote the length of a vector. Given some local coordinate chart $x = (x^1, \cdots, x^d)$ for $U \subset M$, we can write the metric $\mathcal{G}$ in the form

$$\mathcal{G} = g_{ij} dx^i \otimes dx^j,$$

where $g_{ij} : U \to \mathbb{R}$ are called the *metric components* (w.r.t. the coordinate chart x). We would now like to use this construction to define a distance between points on the manifold itself, not just on the tangent spaces. One way to do this is by looking at smooth curves on M, and defining length for these objects, using our metric $\mathcal{G}$. If $[a, b]$ is some closed interval, a smooth function $\gamma : [a, b] \to M$ is called a curve in M. Assuming for convenience that we can express this in one local coordinate system, $\gamma(t) = (x^1(t), \cdots, x^d(t))$, we have that

$$\frac{d}{dt}\gamma(t) = \sum_i \dot{x}_i(t) \partial_{x_i} \in T_{\gamma(t)}M.$$

Using our metric we can measure the velocity of this curve at each point. Hence it makes sense to define the *length* of the curve γ, denoted by $L(\gamma)$, as

$$L(\gamma) := \int_a^b ||\frac{d}{dt}\gamma(t)||_{\mathcal{G}} \, dt.$$

Furthermore we define the *energy* of γ, denoted by $E(\gamma)$, as

$$E(\gamma) := \frac{1}{2}\int_a^b ||\frac{d}{dt}\gamma(t)||_G^2 dt.$$

Using these definitions we can define a distance on our manifold. Let $p, q \in M$, then the distance between these points is given by

$$\boxed{d(p,q) := \inf\left\{L(\gamma) | \gamma : [a,b] \to M \text{ is a piece-wise smooth curve with } \gamma(a) = p, \gamma(b) = q\right\}.} \quad (1)$$

For convenience we assume here that our manifold M is path-connected, so that we can actually find such a piece-wise smooth curve connecting our two points. All of the examples of manifolds we will use in this report are path-connected. This distance is a true distance, in the sense that it satisfies the usual axioms of a distance function.

Since we will need it later on, we present some basic relations between the length and the energy of a curve. The proofs of the following two lemmas are straightforward.

Lemma 2.2. *Let $\gamma : [a,b] \to M$ be a smooth curve. If $s : [\alpha, \beta] \to [a,b]$ is a reparametrization, then*

$$L(\gamma \circ s) = L(\gamma),$$

i.e. the length functional is invariant under reparametrizations.

Lemma 2.3. *Let $\gamma : [a,b] \to M$ be a smooth curve. We have that*

$$L(\gamma)^2 \leq 2(b-a)E(\gamma),$$

with equality if and only if $||\frac{d}{dt}\gamma||$ is constant on $[a,b]$.

In (1) it suffices to look at regular curves, i.e. curves that do not have a vanishing derivative. By Lemma 2.2 we can thus only consider curves parameterized by arc length. Using Lemma 2.3 this means we can also minimize energy instead of length for such curves. For the energy functional, we can compute the Euler-Lagrange equations.

Lemma 2.4. *The Euler-Lagrange equations for the energy functional E are given by*

$$\ddot{x}^i + \Gamma^i_{jk}(x(t))\,\dot{x}^j(t)\dot{x}^k(t) = 0, \quad i = 1, \cdots, d.$$

Here the dot denotes the time derivative, and Γ^i_{jk} are called the Christoffel symbols. They are given in terms of the metric components

$$\Gamma^i_{jk} = \frac{1}{2}g^{il}\left(g_{jl,k} + g_{kl,j} - g_{jk,l}\right),$$

where g^{ij} denotes the inverse matrix of g_{ij}, and $g_{ij,k} = \partial_{x^k} g_{ij}$.

The above system of differential equations are also called the *geodesic equations*, and curves satisfying them are called *geodesics*. Geodesics have constant speed, as a straight forward computation shows. Because we can always reparametrize the geodesics, whenever we use geodesics we will assume that they are parametrized by arc length. By the Picard-Lindelöf Theorem we have the following result.

Theorem 2.5. *Let $(M, \mathcal{G})$ be a Riemannian manifold, $(p, v) \in TM$. Then there exists an $\varepsilon > 0$ and an unique geodesic $\gamma : [0, \varepsilon] \to M$ such that $\gamma(0) = p, \dot{\gamma}(0) = v$. In addition, γ depends smoothly on p and v.*

Definition 2.6. Let $p \in M$. Define $V_p := \{v \in T_pM : \gamma_v \text{ is defined on } [0, 1]\}$. Then

$$\exp_p : V_p \to M, \quad v \mapsto \gamma_v(1),$$

is called the (Riemannian) exponential map.

We want to note that V_p always contains some neighborhood of $0 \in T_pM$, but in general shall not be equal to T_pM. Using the exponential map, we can construct a convenient set of local coordinates around a fixed point p called normal coordinates. They are constructed in such a way that geodesics on the manifold correspond to straight lines in this (local) coordinate system. Moreover, as we will see, they allow for a convenient expansion of the metric around p. Before defining these coordinates we need another result about the exponential map.

Theorem 2.7. *Let $p \in M$ be some arbitrary point. The exponential map $\exp_p$ maps some neighborhood of $0 \in T_pM$ diffeomorphically onto a neighborhood of p.*

Proof.
We are going to show that the differential of $\exp_p$ in 0 has full rank, from which we can conclude that it is a local diffeomorphism around 0 by the inverse function theorem for manifolds. Because T_pM is a linear space, the differential of $\exp_p$ at 0 can be seen as a map $d\exp_p : T_pM \to T_pM$. For $v \in T_pM$ we have that

$$\begin{aligned} d\exp_p(0)(v) &= \frac{d}{dt}\gamma_{tv}(1)|_{t=0} \\ &= \frac{d}{dt}\gamma_v(t)|_{t=0} \\ &= \dot{\gamma}_v(0) \\ &= v. \end{aligned}$$

So, $d\exp_p(0)$ is the identity map on T_pM, and thus has full rank.

□

Now we can define the normal coordinates we discussed above, and prove some important properties of them.

Definition 2.8 (normal coordinates, neighborhood)**.** Let $p_0 \in M$ be some arbitrary point, and let $\exp : T_{p_0}M \to M$ be the Riemannian exponential map. Let V be a star-shaped neighborhood of p_0 such that exp restricted to this neighborhood is a diffeomorphism. Then $U = \exp(V) \subset M$ is called a *normal neighborhood* of p_0. Given an orthonormal basis e_i for $T_{p_0}M$ we get the canonical identification $B : \mathbb{R}^n \to T_{p_0}M$. We can then combine this with the inverse of the exponential map to get a coordinate chart $y = B^{-1} \circ (\exp|_V)^{-1} : U \to \mathbb{R}^n$. Such coordinates are called *normal coordinates*. We usually specify the pair (U, y) as a normal neighborhood of p_0.

Lemma 2.9. *Let $p_0 \in M$, and let (U, y) be a normal neighborhood of p_0. Then*

a) The metric components w.r.t. the normal coordinates in p_0 are $g_{ij} = \delta_{ij}$.

b) Geodesics starting from p_0 to $p \in U$ in the normal coordinates are straight lines, i.e.

$$\gamma(t) = t(y^1(p), \cdots, y^n(p)).$$

c) The Christoffel symbols in p_0 all vanish.

d) The partial derivatives of g_{ij} at p_0 are 0.

Proof.
For a), let e_i be the orthonormal basis for $T_{p_0}M$. Note that if y is the normal coordinate chart, we have that

$$y^{-1} = \exp|_V \circ B,$$

and so the differential of this map at $0 \in \mathbb{R}^n$ is given by

$$d(y^{-1})_0 = d(\exp|_V)_0 \circ dB_0 = I \circ B = B.$$

This follows from the fact that $d(\exp)_0 = I$ and because B is linear. Hence we have that

$$\partial_i|_{p_0} = dy_0^{-1}(\partial_i|_0) = B(\partial_i|_0) = e_i.$$

So at p_0, the coordinate vectors and the chosen orthonormal basis coincide, hence expressing our metric in these coordinates, we have that $g_{ij}(p_0) = \delta_{ij}$.

For b), we recall the property of the exponential map that for $v \in TM$, the geodesic γ_v is given by

$$\gamma_v(t) = \exp(tv),$$

whenever this is defined. So, composing this with our coordinate chart we get that the coordinates of the geodesic are

$$y \circ \gamma_v(t) = B^{-1} \circ \exp^{-1} \circ \exp(tv) = tB^{-1}(v).$$

For c), let $v \in T_pM$ be arbitrary, and consider the geodesic $\gamma_v(t) = tv$. The geodesic equation in this coordinate system simplifies to

$$\Gamma^i_{jk}(tv)v^j v^k = 0.$$

Setting $t = 0$ we thus see that $\Gamma^i_{jk}(0)v^j v^k = 0$. If we choose $v = e_k$ for some k we get that $\Gamma^i_{kk}(0) = 0$. Now for $k \neq l$, consider the vectors $v_1 = e_k + e_l$ and $v_2 = e_k - e_l$. Substituting these into the geodesic equation we get

$$\Gamma^i_{kl}(0) + \Gamma^i_{lk}(0) = 0, \;\; \Gamma^i_{kl}(0) - \Gamma^i_{lk} = 0.$$

Subtracting these from each other allows us to conclude $\Gamma^i_{kl}(0) = 0$. Thus all Christoffel symbols vanish at p_0.

For *d*), note that since $g_{ij}(0) = \delta_{ij}$, we have that $g^{ij} = \delta^{ij}$, so that

$$\Gamma^i_{jk}(0) = \frac{1}{2}\left(g_{ji,k}(0) + g_{ki,j}(0) - g_{jk,i}(0)\right).$$

Next we calculate, making use of the symmetry $g_{ij} = g_{ji}$,

$$\Gamma^i_{jk}(0) - \Gamma^k_{ij}(0) = g_{ij,k} - g_{jk,i} = 0.$$

Making use of this property allows us to conclude that

$$\Gamma^i_{jk} = \frac{1}{2}(2g_{ij,k}) = 0,$$

so that we see that $g_{ij,k} = 0$.

□

As mentioned above, these coordinates allow for an expansion of the metric in a convenient way. It relates the geometry of the manifold to how much the metric deviates from being a euclidean metric. The above lemma already implies that $g_{ij}(p) = \delta_{ij} + \mathcal{O}(||y(p)||^2)$ via a Taylor expansion. We now aim to refine this by calculating the next term in the series. Before we do this though, we have to introduce another very important concept in Riemannian geometry, the curvature tensor, which will turn up in this series expansion. To do this we first introduce connections.

Definition 2.10 (Connection)**.** An affine connection on the tangent bundle is a map $\nabla : C^\infty(TM) \times C^\infty(TM) \to C^\infty(TM)$, written as

$$(X, Y) \mapsto \nabla_X Y,$$

satisfying the properties

a) The map is $C^\infty(M)$ linear in its first argument

$$\nabla_{f_1 X_1 + f_2 X_2} Y = f_1 \nabla_{X_1} Y + f_2 \nabla_{X_2} Y.$$

b) The map is linear (NOT $C^\infty(M)$ linear) in the second argument,

c) It satisfies a product rule. For $f \in C^\infty(M)$ we have

$$\nabla_X (fY) = f \nabla_X Y + (Xf)Y.$$

Given a connection, we define its Christoffel symbols, Γ^k_{ij}, relative to some chart $x = (x^1, \cdots, x^n)$ by

$$\nabla_{\partial_{x^i}} \partial_{x^j} = \Gamma^k_{ij} \partial_{x^k}.$$

Connections serve as a generalization of directional derivatives and can be used for defining a (equivalent) notion of shortest paths on a Riemannian manifold. Note that in this definition we do not use the additional structure of a Riemannian manifold. Placing requirements on the connection which makes it interact nicely with this additional structure allows us to single out an unique connection, called the *Levi-Civita connection*. These two requirements are *torsion freeness* and *metric compatibility*.

Definition 2.11 (Torsion free connection)**.** A connection ∇ is called torsion free (or symmetric) if it satisfies

$$\nabla_X Y - \nabla_Y X = [X, Y] =: XY - YX.$$

Definition 2.12 (Metric compatible)**.** A connection ∇ is *metric compatible* with a Riemannian metric $\mathcal{G}$ if it satisfies

$$\mathcal{G}(\nabla_X Y, Z) + \mathcal{G}(Y, \nabla_X Z) = X(\mathcal{G}(Y, Z)).$$

Metric compatibility resembles a differentiation rule that the Euclidean inner product satisfies. For the Levi-Cevita connection, the definition of its Christoffel symbols above and the previous definition given coincide, which is why we used the same name and symbol. Next we introduce the *Riemannian curvature tensor* $R : \mathfrak{X}(M) \times \mathfrak{X}(M) \times \mathfrak{X}(M) \to \mathfrak{X}(M)$ define by

$$R(X,Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z. \tag{2}$$

This is a $(1,3)$ tensor field on M. We can thus write this in terms of local coordinates as

$$R = R^l_{ijk} dx^i \otimes dx^j \otimes dx^k \otimes \partial_{x^l},$$

with $R(\partial_i, \partial_j)\partial_k = R^l_{ijk}\partial_l$. We have the following formula for the components in terms of the Christoffel symbols

Proposition 2.13. *Let ∇ be the Levi-Cevita connection of some Riemannian manifold $(M, \mathcal{G})$, and Γ^k_{ij} its Christoffel symbols. In some local coordinate system $x = (x^1, \cdots, x^n)$ we have that the curvature tensor can be written as*

$$R^l_{ijk} = \Gamma^l_{jk,i} - \Gamma^l_{ik,j} + \Gamma^m_{jk}\Gamma^l_{im} - \Gamma^m_{ik}\Gamma^l_{jm}.$$

Proof.
Let us use the notation $\partial_i = \partial_{x^i}$ for convenience. We write out $R(\partial_i, \partial_j)\partial_k$. This gives us

$$\begin{aligned}
R(\partial_i, \partial_j)\partial_k &= \nabla_{\partial_i}\nabla_{\partial_j}\partial_k - \nabla_{\partial_j}\nabla_{\partial_i}\partial_k - \nabla_{[\partial_i,\partial_j]}\partial_k \\
&= \nabla_{\partial_i}\nabla_{\partial_j}\partial_k - \nabla_{\partial_j}\nabla_{\partial_i}\partial_k \\
&= \nabla_{\partial_i}\Gamma^m_{jk}\partial_m - \nabla_{\partial_j}\Gamma^m_{ik}\partial_m \\
&= \partial_m\Gamma^m_{jk,i} + \Gamma^m_{jk}\nabla_{\partial_i}\partial_m - \partial_m\Gamma^m_{ik,j} - \Gamma^m_{ik}\nabla_{\partial_j}\partial_m \\
&= \partial_m\Gamma^m_{jk,i} - \partial_m\Gamma^m_{ik,j} + \Gamma^m_{jk}\Gamma^l_{im}\partial_l - \Gamma^m_{ik}\Gamma^l_{jm}\partial_l.
\end{aligned}$$

So, comparing this new expression with $R^l_{ijk}\partial_l$ allows us to conclude that

$$R^l_{ijk} = \Gamma^l_{jk,i} - \Gamma^l_{ik,j} + \Gamma^m_{jk}\Gamma^l_{im} - \Gamma^m_{ik}\Gamma^l_{jm}.$$

□

Lastly we define the *curvature function* $\kappa : C^\infty(M) \times C^\infty(M) \to C^\infty(M)$ by

$$\kappa(X,Y) = \mathcal{G}(R(X,Y)Y, X). \tag{3}$$

This is closely related to the *sectional curvature* $K_p : T_p(M) \times T_p(M) \to \mathbb{R}$, which is a normalized variant of the curvature function

$$K_p(X|_p, Y|_p) = \frac{\kappa(X|_p, Y|_p)}{|X|_p|^2|Y|_p|^2 - g(X|_p, Y|_p)}. \tag{4}$$

We are now finally ready to prove the last result about general Riemannian geometry we will need for this report. In the next lemma we will calculate the next term in the expansion $g_{ij}(p) = \delta_{ij} + \mathcal{O}(||y(p)||^2)$.

Remark 2.14. *In order to define the Hessian of a function $f : M \to \mathbb{R}$, we need to choose a connection ∇. Connections can uniquely be extended to act on all tensor fields on M. We will not go into detail on this here as we will not need this further. The Hessian of f is then defined by letting the connection act on the differential of f, $Hf = \nabla df$, which in local coordinates looks like*

$$\nabla df = \left(\partial_{x^i}\partial_{x^j} f - \partial_{x^k} f \Gamma_{ij}^k\right) dx^i \otimes dx^j,$$

with Γ_{ij}^k the Christoffel symbols of ∇.

Because we will be working with the Levi-Cevita connection below, for which we have proven that the Christoffel symbols vanish in normal coordinates, we do not have to worry about the additional effect of the connection and we can just compute the Hessian as we would normally do.

Lemma 2.15. *Let $p_0 \in M$ be given, and let (U, y) be a normal neighborhood of p_0. For $p \in U$ the components of the metric tensor relative to this coordinate system can be written as (with $y = y(p)$)*

$$g_{ij}(y) = \delta_{ij} - \frac{1}{3}\sum_{k,l=1}^{n} R_{ikjl}\, y^k y^l + \mathcal{O}(||y||^3). \tag{5}$$

Proof.
As mentioned we show this using a Taylor expansion. Lemma 2.9 already shows that the zeroth and first order terms are correct. We now calculate the Hessian of g_{ij}. Taking the above remark into consideration, we can calculate the Hessian as we are used to.

We start by establishing the identity $g_{ij,k} = \Gamma_{ki}^l g_{lj} + \Gamma_{kj}^l g_{il}$. Notice that we can write $\partial_k g_{ij}$ as $\partial_k \mathcal{G}(\partial_i, \partial_j)$, so that we can apply the metric compatibility of the Levi-Cevita connection to conclude

$$\begin{aligned}
\partial_k g_{ij} &= \mathcal{G}(\nabla_{\partial_k}\partial_i, \partial_j) + \mathcal{G}(\partial_i, \nabla_{\partial_k}\partial_j) \\
&= \mathcal{G}(\Gamma_{ki}^n \partial_n, \partial_j) + \mathcal{G}(\partial_i, \Gamma_{kj}^n \partial_n) \\
&= \Gamma_{ki}^n g_{nj} + \Gamma_{kj}^n g_{in}.
\end{aligned}$$

We now calculate $g_{ij,kl}(0)$ using this characterization of $g_{ij,k}$. Making use of the fact that $g_{ij}(0) = \delta_{ij}$ and $g_{ij,k}(0) = 0$ we get that

$$g_{ij,kl}(0) = \Gamma_{ki,l}^j(0) + \Gamma_{kj,l}^i(0). \tag{6}$$

Next, we claim that $\Gamma_{ij,l}^k(0) + \Gamma_{jl,i}^k(0) + \Gamma_{li,j}^k(0) = 0$. To show this consider some vector $v \in U$, and the geodesic passing through v, given by $\gamma_v(t) = ty(v)$. We saw in the lemma above that the geodesic equations reduce to

$$\Gamma_{ij}^k(tv) v_i v_j = 0.$$

If we differentiate this to t, and set $t = 0$ we get that

$$\Gamma_{ij,l}^k(0) v_l v_i v_j = 0.$$

This is a homogeneous polynomial of degree 3 in the coefficients v_i, that vanishes at least on some open set. This means that each coefficient in front of $v_i v_j v_l$ must be equal to 0. This is exactly

equal to all permutations of i, j, l in $\Gamma^k_{ij,l}$. Furthermore from the symmetry $g_{ij} = g_{ji}$ we get that $\Gamma^k_{ij} = \Gamma^k_{ji}$, so that we can conclude that

$$\Gamma^k_{ij,l}(0) + \Gamma^k_{jl,i}(0) + \Gamma^k_{li,j}(0) = 0. \tag{7}$$

Now from proposition 2.13, combined with lemma 2.9, we know that $R^l_{ijk}(p_0) = R_{ijkl}(p_0) = \Gamma^l_{jk,i}(0) - \Gamma^l_{ik,j}(0)$. Combining this fact together with (7) we get that

$$R_{iklj}(p_0) + R_{ilkj}(p_0) = 3\Gamma^j_{kl,i}(0).$$

Finally, we can use the above identity in (6) to conclude that

$$\begin{aligned} g_{ij,kl}(0) &= \Gamma^j_{ki,l}(0) + \Gamma^i_{kj,l}(0) \\ &= \frac{1}{3}\left(R_{lkij} + R_{likj} + R_{lkji} + R_{ljki}\right) \\ &= \frac{1}{3}\left(R_{lkij} + R_{likj} - R_{lkij} + R_{ljki}\right) \\ &= -\frac{1}{3}\left(R_{lijk} + R_{kijl}\right). \end{aligned}$$

This proves the lemma.

□

The last thing we want to show is how to solve advection PDEs on smooth manifolds, using the method of characteristics. Consider the evolution equation below, with f some smooth function on M and $\boldsymbol{c}$ a smooth vector field on M.

Lemma 2.16. *Let f be a smooth function on M and $\boldsymbol{c}$ be a smooth vector field on M. The solution to the following advection PDE*

$$\begin{cases} \frac{\partial W}{\partial t}(\boldsymbol{p}, t) &= -\boldsymbol{c}W(\boldsymbol{p}, t) \\ W(\boldsymbol{p}, 0) &= f(\boldsymbol{p}), \end{cases} \tag{8}$$

is given by

$$W(p, t) = f(\gamma_p(-t)),$$

where $\gamma_p : \mathbb{R} \to M$ is an integral curve of the vector field $\boldsymbol{c}$, starting at p.

Proof.
From the method of characteristics, we know that $\frac{d}{dt}W(\gamma_p(t), t) = 0$, which can be verified by

$$\begin{aligned} \frac{d}{dt}W(\gamma_p(t), t) &= \frac{\partial W}{\partial t}(\gamma(t), t) + \langle dW(\cdot, t)|_{\gamma_p(t)}, \dot{\gamma}_p(t)\rangle \\ &= \frac{\partial W}{\partial t}(\gamma(t), t) + \langle dW(\cdot, t)|_{\gamma_p(t)}, \boldsymbol{c}|_{\gamma_p(t)}\rangle \\ &= \frac{\partial W}{\partial t}(\gamma(t), t) + \boldsymbol{c}W(p, t) \\ &= 0. \end{aligned}$$

We thus see that

$$W(\gamma_p(t), t) = W(\gamma_p(0), 0) = f(p).$$

Now using the fact that $\gamma_{\gamma_p(t)}(-t) = p$, which we visualized in Figure 10, we get that the solution is given by

$$W(p, t) = f(\gamma_p(-t)).$$

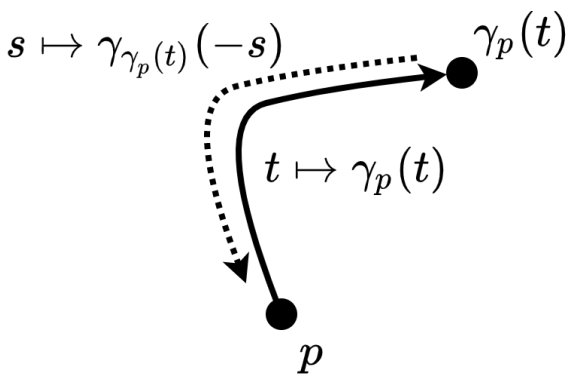


Figure 10: A figure illustrating the idea that $\gamma_{\gamma_p(t)}(-t) = p$. We first walk along the curve γ, to then walk backwards again for the same amount of time.

□

Remark 2.17. *We want to remark that in general, integral curves will not exist for all $t \in \mathbb{R}$. However, we will see below that for our purpose, this (implicit) assumption will be sufficient.*

2.2 Lie Groups

In this section we collect some basic facts about Lie groups, with the aim of calculating the curvature of a Lie group endowed with a left-invariant metric. In particular, we present where possible both general theory and apply this to the example of interest, the group $SE(2)$ of rigid body motions in the plane. The aim is to provide some basic facts about $SE(2)$, ending with calculating the curvature for particular left-invariant metrics on this Lie group.

2.2.1 Lie Groups

Lie groups are differentiable or smooth manifolds endowed with a group structure that respects the smooth manifold structure.

Definition 2.18 (Lie group)**.** A Lie group is a smooth manifold G, together with a group structure on the set G such that the functions

$$\mu : (x, y) \mapsto x \cdot y \ (= xy), \quad \iota : x \mapsto x^{-1},$$

are smooth.

We denote left translation by some fixed element g of G with L_g, i.e. $L_g : G \to G$ with $L_g(h) = gh$, and similarly we denote right translation by R_g. We now have present a definition of an important construct on G:

Definition 2.19. A smooth vector field X on G is called left-invariant if we have that for all $g_1, g_2 \in G$

$$X|_{g_1 g_2} = (L_{g_1})_* X|_{g_2}. \tag{9}$$

The set of all left-invariant vector fields forms a linear subspace of $\mathfrak{X}(G)$, as taking linear combinations of such vector fields retains the left-invariance property. It turns out that this linear subspace has the same dimension as the tangent space at the identity of G, $T_e(G)$. This follows from the fact that the values of a left-invariant vector field are completely determined by its value in a single point. In particular, for $X \in T_e G$ we introduce the notation v_X for the vector field defined as

$$v_X(g) = (L_g)_* X,$$

which obviously is a left-invariant vector field by construction.

On the space of vector fields we furthermore have the commutator bracket, which for general vector fields X, Y is given by $[X, Y] = XY - YX$. It turns out that this bracket also respects the left-invariance property, that is if X, Y are left-invariant, so is $[X, Y]$. This additional structure on the linear space of left-invariant vector fields is called a Lie algebra. We denote the linear space of left-invariant vector fields, together with the bracket, by $\mathfrak{g}$. A Lie algebra is abstractly defined as

Definition 2.20 (Lie algebra)**.** A Lie algebra is a linear space V equipped with a bracket operation, a bilinear map $[\cdot, \cdot] : V \times V \to V$ such that for $X, Y, Z \in V$ we have

a) $[X, Y] = -[Y, X]$

b) $[X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0$

The bracket of a Lie algebra can be concisely described by the *structure constants*, which are the coefficients c_{ij}^k such that

$$[e_i, e_j] = \sum_{i=1}^{n} c_{ij}^k e_k, \tag{10}$$

where $\{e_i\}$ is a basis for the Lie algebra. Note that the structure constants are anti-symmetric in the two lower indices, due to the anti-symmetry of the bracket.

Left translation also gives rise to an important group action, which we will need later on. Consider some space of functions defined on G, denoted by $F(G)$. This could for example be bounded functions, continuous functions, etc. We then define the following actions of G on $F(G)$.

Definition 2.21. Let G be a Lie group, and let $g \in G$. The right-regular representation $\mathcal{R} : G \to F(G)$ is defined by

$$\mathcal{R}_g U(h) = U(hg) \quad \text{for all} \quad h \in G, U \in F(G).$$

Similarly, the left-regular representation $\mathcal{L} : G \to F(G)$ is defined by

$$\mathcal{L}_g U(h) = U(g^{-1}h) \quad \text{for all} \quad h \in G, U \in F(G).$$

For $X \in T_eG$, we now consider the *maximal integral curve* α_X of the vector field v_X, with initial point e. From general results we know that integral curves always exist for some time interval $(-\varepsilon, \varepsilon)$, however for these integral curves we can prove they behave better, as the following lemma shows.

Lemma 2.22. *Let $X \in T_eG$. The maximal integral curve α_X has domain $\mathbb{R}$. For all $t, s \in \mathbb{R}$, it satisfies $\alpha_X(t+s) = \alpha_X(t)\alpha_X(s)$.*

Proof.
Let α be any integral curve for v_X, with an arbitrary initial point. For $g \in G$, define $\alpha_g(t) = L_g\alpha(t) = g\alpha(t)$. Differentiating α_g with respect to t we find that

$$\frac{d}{dt}\alpha_g(t) = T_{\alpha(t)}L_g \frac{d}{dt}\alpha(t) = T_{\alpha(t)}L_g v_X|_{\alpha(t)} = v_X|_{\alpha_g(t)},$$

which tells us that left-translations still result in integral curves of v_X. We will use this as follows. Let $I \subset \mathbb{R}$ be the interval of definition for α_X, the maximal integral curve, and fix some $t_1 \in I$. Put $x_1 = \alpha(t_1)$, and define $\alpha_1(t) = x_1\alpha(t)$. This is an integral curve for v_X with initial point x_1, and is defined for $t \in I$. However, the maximal integral curve of v_X with starting point x_1 is given by $\alpha_2(t) = \alpha_X(t+t_1)$, with domain $I - t_1$, by definition of α as maximal curve. Hence we need to have that $I \subset I - t_1$, or $s + t_1 \subset I$ for all $s, t_1 \in I$, implying that $I = \mathbb{R}$.

Now, fix some $s \in \mathbb{R}$. From the discussion above it follows that both $t \mapsto \alpha_X(s)\alpha_X(t)$ and $t \mapsto \alpha_X(s+t)$ are maximal integral curves for v_X with initial point $\alpha_X(s)$, hence we conclude they must be the same, giving us $\alpha_X(s)\alpha_X(t) = \alpha_X(s+t)$. □

We can now define the *Lie group exponential* as follows

Definition 2.23. Let $X \in T_eG$, and let α_X be defined as above. We then define $\exp : T_eG \to G$ by

$$\exp(X) = \alpha_X(1).$$

We will also use the notation $e^X := \exp(X)$.

We want to note that this is an entirely different notion from the Riemannian exponential map we have seen before. We will refer to both as the exponential map, and where confusion can arise specify which one is meant. Now, if we would want to calculate this mapping, this definition is not the most convenient to use, as it would require us to find the integral curve, which amounts to solving a system of coupled ODEs. In certain cases, like with matrix groups, this exponential map coincides with a well known function, for matrices this is the matrix exponential.

2.2.2 Homogeneous Spaces

For groups, the notion of an action of the group on some mathematical object is important, as it tells us something about symmetries present in this object. There is a natural analog of this notion for Lie groups, where we require compatibility with the smooth structure of the Lie group.

Definition 2.24. Let M be a smooth manifold, and G a Lie group. We say that G acts smoothly from the left on M if there exists a smooth map $\odot : G \times M \to M$ such that

a) $e \odot x = x$ for all $x \in M$,

b) $g \odot (h \odot x) = gh \odot x$ for all $g, h \in G, x \in M$.

Whenever the action is clear, we will omit the $\odot$ symbol.

Such an action is said to be *transitive* if for $x, y \in M$ there exists a $g \in G$ such that $x = g \odot y$. In words this means we can "translate" any point on M to any other point using our group action.

A smooth manifold M together with a smooth, transitive action of some Lie group G is called a *homogeneous space*.

Theorem 2.25. *Let M be a homogeneous space of G, and let $p \in M$ be arbitrary. The manifolds M and $G/stab_G(p)$ are diffeomorphic.*

The previous theorem shows that each homogeneous space can be identified with some Lie group quotient G/H, where H is a closed subgroup of G. Note that H needs to be closed, because $\text{stab}_G(p)$ is always closed, due to continuity of the action. We will usual adopt this viewpoint of homogeneous spaces, as quotient spaces, and thus if $p \in G/H$ we have to keep in mind that p is an equivalence class.

We briefly want to mention that on homogeneous spaces, we can also make sense of left-invariant vector fields. We do this by replacing the left-translation L_g in Definition 2.19 by the left-action of G on the homogeneous space.

2.2.3 Riemannian Geometry on Lie Groups

Note that all of the above definitions are for general Riemannian manifolds. We now wish to take a particular look at Lie groups that have a metric that "respects the group structure", i.e. left-invariant metrics.

Definition 2.26. A Riemannian metric $\mathcal{G}$ on a Lie group G is called *left-invariant* if all left translations are isometries. This means that for $g, h \in G$, $X, Y \in T_g(M)$ we have that

$$\mathcal{G}_g(X, Y) = \mathcal{G}_{hg}((L_h)_* X, (L_h)_* Y).$$

Note that, as was similar for left-invariant vector fields, a left-invariant metric is determined by the inner product we assign to the tangent space at the identity. We thus only need to specify one symmetric positive definite matrix to fix a left-invariant metric. Also note that this definition again translates naturally to a concept of left-invariant metrics on homogeneous spaces, similar to the left-invariant vector fields.

Left-invariant metrics satisfy the nice property that the distance they induce is invariant under left-translations, as the following lemma shows.

Lemma 2.27. *Let G be a lie group, $\mathcal{G}$ a left-invariant metric. We then have that for $g_1, g_2 \in G$*

$$d(g_1, g_2) = d(e, g_1^{-1} g_2) = d(g_1^{-1} g_2, e). \tag{11}$$

Proof.

Let $\gamma : [0, 1] \to G$ be a piece-wise smooth curve connecting g_1 and g_2. Then consider the curve $L_{g_1^{-1}} \circ \gamma$. This curve connects e with $g_1^{-1} g_2$, and by left-invariance of the metric we find that

$$\begin{aligned}
L(L_{g^{-1}} \circ \gamma) &= \int_0^1 ||\frac{d}{dt} L_{g_1^{-1}} \circ \gamma(t)||dt \\
&= \int_0^1 \sqrt{\mathcal{G}_{g_1^{-1}\gamma(t)} \left(\frac{d}{dt} L_{g_1^{-1}} \circ \gamma(t), \frac{d}{dt} L_{g_1^{-1}} \circ \gamma(t) \right)} dt \\
&= \int_0^1 \sqrt{\mathcal{G}_{g_1^{-1}\gamma(t)} \left((L_{g_1^{-1}})_* \dot{\gamma}(t), (L_{g_1^{-1}})_* \dot{\gamma}(t) \right)} dt \\
&= \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)} \left(\dot{\gamma}(t), \dot{\gamma}(t) \right)} dt \\
&= L(\gamma).
\end{aligned}$$

So we can identify the set of smooth curves connecting g_1 and g_2 with the set of smooth curves connecting e with $g_1^{-1}g_2$ in such a way that the length of the curves is the same. From this we thus conclude that $d(g_1, g_2) = d(e, g_1^{-1}g_2)$. The last equality follows from the fact that d is a distance, and thus it is symmetric in its arguments. □

From now on we assume that $\mathcal{G}$ on G is a left-invariant metric. Note that given two left-invariant vector fields X, Y the function $g \mapsto \mathcal{G}(X, Y)$ is constant, as its value is completely determined from the value at $e \in G$

$$\mathcal{G}(X|_g, Y|_g) = \mathcal{G}(T_e(L_g)X|_e, T_e(L_g)Y|_e) = \mathcal{G}(X|_e, Y|_e).$$

This means that for any other vector field Z, we have that

$$Z(\mathcal{G}(X, Y)) = 0. \tag{12}$$

Following [56], we can use this to show that in left-invariant frames, the curvature for left-invariant metrics is completely determined by the structure constants, equation 10. Given an orthonormal basis $\{e_i\}$ of $\mathfrak{g}$, we denote the structure constants by c_{ij}^k. We first claim that we can write the structure constants as

$$c_{ij}^k = \mathcal{G}([e_i, e_j], e_k), \tag{13}$$

which follows from the orthonormality of the frame. Using the fact that ∇ is torsion free and metric-compatible, equation (12) and the orthonormality of the frame we can obtain a formula for $\nabla_{e_i} e_j$ as follows

$$\nabla_{e_i} e_j = \sum_k \frac{1}{2} \left(c_{ij}^k - c_{jk}^i + c_{ki}^j \right) e_k. \tag{14}$$

Since the Riemannian curvature tensor is completely determined by ∇, we see that indeed only the structure constants will determine the curvature in this frame.

2.2.4 Cartan Connections

Although we see that the extra structure on the Lie group allows us to say something more about the Levi-Cevita connection then in the general case, we can look for a connection which respects the structure of our Lie group in a different way. This will lead to so called *Cartan connections*.

Definition 2.28 (Left-invariant connections)**.** A connection ∇ on G is called left-invariant if for all left-invariant vector fields X, Y we have that $\nabla_X Y$ is also left-invariant.

As was the case with left-invariant vector fields and metrics, left-invariant connections correspond to an object only living on the Lie algebra $\mathfrak{g}$, namely bi-linear forms.

Lemma 2.29. *There is a bijection between the set of left-invariant connections and the set of bi-linear forms on* $\mathfrak{g}$.

Proof.
Let ∇ be given, and let $X_v, Y_v \in \mathfrak{g}$ be arbitrary. Since ∇ is a connection, and $\nabla_X Y \in \mathfrak{g}$, $\nabla : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ is $\mathbb{R}$ linear, i.e. defines a bilinear form on the Lie algebra.

Let α be a bi-linear form on the Lie algebra. Let $\{X_i\}$ be some basis for $\mathfrak{g}$. Define $a_{ij} = \alpha(X_i, X_j)$. For general vector fields $Y = f^i X_i, Z = g^j X_j$ define the connection by

$$\nabla_Y Z = f^i \left((X_i g^j) X_j + g^j a_{ij} \right).$$

If Y and Z are left-invariant, that is f^i and g^j are constant, we see that we indeed get a left-invariant vector field again, so this connection is indeed left-invariant. □

Recall that if we require that a connection is torsion-free and metric compatible we fix an unique one, so we cannot expect that our left-invariant connections will satisfy these conditions. We wonder when such a left-invariant connection is torsion-free. By the lemma above we might as well characterize such connections by bi-linear forms. So let α be a bi-linear form, and ∇^α be the connection associated with it. First of, we can always write a bi-linear form as the sum of a symmetric and anti-symmetric form, $\alpha = \alpha_s + \alpha_a$. We then have that the torsion tensor of ∇^α is given by (evaluated on $X, Y \in \mathfrak{g}$)

$$T(X,Y) = \nabla^\alpha_X Y - \nabla^\alpha_Y X - [X,Y] = \alpha(X,Y) - \alpha(Y,X) - [X,Y] = 2\alpha_a(X,Y) - [X,Y].$$

Hence we see that a connection ∇^α is symmetric/torsion-free iff $\alpha_a(X,Y) = \frac{1}{2}[X,Y]$.

We call a left-invariant connection that additionally satisfies that exponential curves are auto-parallel a Cartan connection [57]. Since we are working on a Lie algebra, we have already have a bi-linear form on this space. We denote the connection that is induced by the bi-linear form $\alpha_\nu(X,Y) = \nu[X,Y]$ by $\nabla^{[\nu]}$, and call these the *Lie-Cartan connections*. We thus see that if $\nu = \frac{1}{2}$, this connection is symmetric.

The curvature relative to this connection, which is obtained by replacing the Levi-Cevita connection in 2 by this connection, is given by (again with $X, Y, Z \in \mathfrak{g}$)

$$\begin{aligned} R(X,Y)Z &= \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z \\ &= \nabla_X \nu[Y,Z] - \nabla_Y \nu[X,Z] - \nu[[X,Y],Z] \\ &= \nu^2[X,[Y,Z]] - \nu^2[Y,[X,Z]] + \nu[Z,[X,Y]] \\ &= \nu(1-\nu)[Z,[X,Y]]. \end{aligned}$$

Relative to a left-invariant frame we thus have that the components of the curvature tensor are given by

$$R^k_{ijl} = \nu(1-\nu) \sum_{q=1}^{n} c^k_{iq} c^q_{jl}, \tag{15}$$

where c^k_{ij} are the structure constants of the frame, from equation (10).

Remark 2.30. *Recall from the introduction that association field lines are either modeled by exponential curves or by geodesics. The first have parallel velocity w.r.t. Cartan connection whereas the geodesics have parallel momentum. For an intuitive visualisation see [57, Fig.7].*

2.3 The homogeneous Spaces $\mathbb{M}_d$

Of particular interest in image analysis and to us in this report are the homogeneous spaces of positions and orientations, which we will be introduced below.

For this we first define the *Special Euclidean group*, $SE(d)$, which is the Lie group consisting of rigid body motions. As a set, $SE(d) = \mathbb{R}^d \times SO(d)$, however it does not carry the direct product group operation, but its group operation is instead given by

$$(x_1, R_1)(x_2, R_2) = (x_1 + R_1 x_2, R_1 R_2).$$

One readily verifies this is a group product with inverse given by

$$(x, R)^{-1} = (-R^{-1}x, R^{-1}),$$

where R^{-1} denotes the inverse of the matrix element $R \in SO(d)$. The way we construct the homogeneous space of positions and orientations begins with fixing some reference axis in S^{d-1}, denoted by $\boldsymbol{a}$. The Lie group $SO(d)$ naturally acts on $S^{d-1} \subset \mathbb{R}^d$ via matrix multiplication. Consider the subgroup of $SO(d)$ that leaves $\boldsymbol{a}$ invariant. We can identify this subgroup with $SO(d-1)$. The homogeneous space $\mathbb{M}_d$ is then given by the quotient $SE(d)/H$, with $H = \{0\} \times SO(d-1)$, seeing $SO(d-1)$ as a subgroup of $SO(d)$ using the reference axis $\boldsymbol{a}$. As a set, $\mathbb{M}_d = \mathbb{R}^d \times S^{d-1}$. For calculations with distances, we often need a reference element in $\mathbb{M}_d$, for which we choose $p_0 = (0, \boldsymbol{a})$.

2.3.1 The 2-dimensional Case $\mathbb{M}_2$ and $SE(2)$

All of the applications in this report will be concerned with gray-scale two-dimensional image data, so that we have to consider the particular case $\mathbb{M}_2$. We will need some details about $\mathbb{M}_2$ for our applications, which are collected here.

Since for the two dimensional case $H = \{0\} \times SO(1) = \{0\} \times \{e\}$ is the trivial subgroup, we have that $\mathbb{M}_2 = SE(2)/\{e\} \cong SE(2)$, i.e. they are diffeomorphic manifolds. Although we will use both of these manifolds throughout this report, we need to mention that there is still a distinction between $\mathbb{M}_2$ and $SE(2)$, as strictly speaking they do not carry the same structure. In particular, $SE(2)$ carries a group structure while $\mathbb{M}_2$ does not. For our fixed reference element in $\mathbb{M}_2$, we will choose $p_0 = (0, 0, 0) \in \mathbb{R}^2 \times S^1$.

We now provide some details about $SE(2)$. Since as a set, $SE(2) = \mathbb{R}^2 \times SO(2)$, its elements have the form (x, y, R_θ) with $x, y \in \mathbb{R}$ and $R_\theta \in SO(2) \subset \mathbb{R}^{2\times 2}$. However, because $SO(2) \cong (\mathbb{R}/2\pi\mathbb{Z})$, we can also identify $SE(2)$ with $\mathbb{R}^2 \times (\mathbb{R}/2\pi\mathbb{Z})$, and thus elements can also be written as $g = (x, y, \theta \mod 2\pi)$ with $\theta \in \mathbb{R}$. Lastly, we identified $\mathbb{M}_2$ with $\mathbb{R}^2 \times S^1$, where we see $S^1 \subset \mathbb{R}^2$, so that we may also write elements in $SE(2)$ as $(x, y, \boldsymbol{n}(\theta))$ for $\theta \in \mathbb{R}$ and $\boldsymbol{n}(\theta) = (\cos(\theta), \sin(\theta))$. We will use all of these identifications interchangeably, and refer to this as the "fixed coordinate system". These relations are summarized in equation (16) below.

$$\boxed{\mathbb{R}^2 \times (\mathbb{R}/2\pi\mathbb{Z}) \ni (x, y, \theta) \leftrightarrow \underbrace{p = (x, y, \boldsymbol{n}(\theta))}_{\in \mathbb{M}_2 = \mathbb{R}^2 \times S^1} \leftrightarrow g = (x, y, R_\theta) \in SE(2)} \tag{16}$$

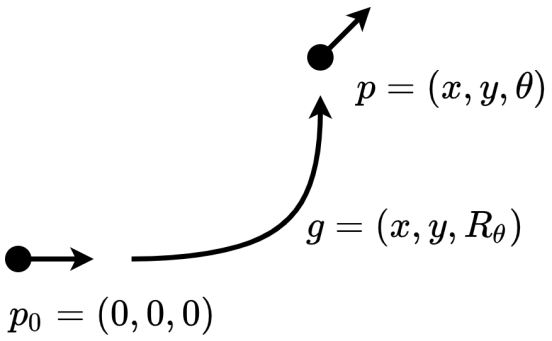


Figure 11: Figure showing the action of $SE(2)$ on the (trivial) homogeneous space $\mathbb{M}_2$. This action is the same as the group product because we can identify $SE(2)$ and $\mathbb{M}_2$.

For $g_1 = (x_1, y_1, \theta_1), g_2 = (x_2, y_2, \theta_2) \in SE(2)$ the group product is explicitly given by

$$(x_1, y_1, \theta_1)(x_2, y_2, \theta_2) = (x_1 + x_2 \cos\theta - y_2 \sin\theta, y_1 + x_2 \sin\theta + y_2 \cos\theta, \theta_1 + \theta_2 \mod 2\pi). \quad (17)$$

The Lie group $SE(2)$ also has an important action on $\mathbb{R}^2$, namely the rotating and translating of elements of $\mathbb{R}^2$. For $g = (\boldsymbol{x}, R_\theta) \in SE(2)$ and $\boldsymbol{v} \in \mathbb{R}^2$ this action is given by

$$g \odot \boldsymbol{v} = R_\theta \boldsymbol{v} + \boldsymbol{x}. \quad (18)$$

Given the fixed coordinate system (16) on $SE(2)$, a natural basis for $T_e SE(2)$ is given by

$$A_1 = \partial_x|_e, \;\; A_2 = \partial_y|_e \;\; A_3 = \partial_\theta|_e. \quad (19)$$

As discussed before, tangent vectors at the identity give rise to left-invariant vector fields through left-multiplication. The corresponding vector fields of the basis above are given by

$$\mathcal{A}_1 = \cos(\theta)\partial_x + \sin(\theta)\partial_y, \; \mathcal{A}_2 = -\sin(\theta)\partial_x + \cos(\theta)\partial_y, \; \mathcal{A}_3 = \partial_\theta. \quad (20)$$

We show how this is computed for $\mathcal{A}_1$. Let $g = (x, y, \theta) \in SE(2)$ be given and f some smooth function on $SE(2)$. We then compute

$$\begin{aligned}
\mathcal{A}_1|_g f &= L_g^* \partial_x|_e f \\
&= \partial_x|_e (f \circ L_g) \\
&= \lim_{t \to 0} \frac{f\big((x, y, \theta)(t, 0, 0)\big) - f\big((x, y, \theta)\big)}{t} \\
&= \lim_{t \to 0} \frac{f\big((x + t\cos(\theta), y + t\sin(\theta), \theta\big) - f\big((x, y, \theta)\big)}{t} \\
&= \cos(\theta)\partial_x f + \sin(\theta)\partial_y f.
\end{aligned}$$

The other two vector fields are done in a similar fashion. Since $SE(2)$ is three dimensional, these three vector fields in fact form a basis for the Lie algebra of $SE(2)$. We will calculate the structure constants next. The only non-zero ones are in fact

$$[\mathcal{A}_1, \mathcal{A}_3] = -\mathcal{A}_2, \quad [\mathcal{A}_2, \mathcal{A}_3] = \mathcal{A}_1.$$

We show the computation of the first bracket, the other ones are done in a similar fashion.

$$\begin{aligned}
[\mathcal{A}_1, \mathcal{A}_3] &= (\cos(\theta)\partial_x + \sin(\theta)\partial_y)\,\partial_\theta - \partial_\theta\,(\cos(\theta)\partial_x + \sin(\theta)\partial_y) \\
&= \cos(\theta)\partial_x\partial_\theta + \sin(\theta)\partial_y\partial_\theta + \sin(\theta)\partial_x - \cos(\theta)\partial_x\partial_\theta - \cos(\theta)\partial_y - \sin(\theta)\partial_y\partial_\theta \\
&= \sin(\theta)\partial_x - \cos(\theta)\partial_y \\
&= -\mathcal{A}_2.
\end{aligned}$$

Next, we want to discuss the exponential map of $SE(2)$, as this will be very important for later on. One way we could compute it would be by identifying $SE(2)$ with a matrix group, where we could use that the Lie group exponential is the same as the matrix exponential. However recall from Lemma 2.16 that convection PDEs are solved by integral curves. Since the exponential curves are defined as integral curves, we can use this fact to compute the exponential map. Note that if we want the integral curve of a left-invariant vector field at some point $g \in G$, all we have to do is left translate the integral curve at the identity e, due to the left-invariance of the vector field. We thus have that

Corollary 2.31. *Let f be a smooth function on G, and let $\boldsymbol{c} = c^i\mathcal{A}_i$ be a left-invariant vector field. The solution to the advection PDE*

$$\begin{cases} \frac{\partial W}{\partial t}(g,t) &= -\boldsymbol{c}W(g,t) \\ W(g,0) &= f(\boldsymbol{p}), \end{cases} \tag{21}$$

is given by transportation along exponential curves,

$$W(g,t) = f(g\exp(-tc^iA_i)).$$

Remark 2.32. *We want to remark that for left-invariant convection we have more structure than in the general case. We will show this without going into much detail Note that $t \mapsto \exp^{tc^iA_i}$ is a one-parameter subgroup, so that we may write*

$$R_{e^{-tc^iA_i}} = \left(R_{e^{-\frac{t}{n}c^iA_i}}\right)^n = \left(I - \frac{t}{n}dR(c^iA_i) + \mathcal{O}\left((\frac{t}{n})^2\right)\right)^n \to e^{-tdR(c^iA_i)}.$$

Here $dR(c^iA_i)$ is the generator of $R_{c^iA_i}$, and can be thought of as a kind of derivative of the operator. This relation can be summarised as follows

$$\boxed{e^{-tdR(A)} = R_{e^{-tA}} = R^{-1}_{e^{tA}}} \tag{22}$$

What we can now do is write out this PDE in fixed coordinates and look for the characteristic curves, which will be exactly the exponential curves we are looking for. For this we use the representation of $\boldsymbol{c}$ via equation (28) to write the convection PDE as

$$\begin{cases} \frac{\partial W}{\partial t}(g,t) &= \left(c^1\cos(\theta) - c^2\sin(\theta)\right)\partial_x W \\ &\quad + \left(c^1\sin(\theta) + c^2\cos(\theta)\right)\partial_y W \\ &\quad + c^3\partial_\theta W, \\ W(\boldsymbol{p},0) &= f(\boldsymbol{p}). \end{cases} \tag{23}$$

Having rewritten it suggestively as this, we can compute the characteristic curves $\gamma(t) = (x(t), y(t), \theta(t))$ quite easily by solving

$$\begin{cases} \dot{x}(t) &= c^1 \cos(\theta(t)) - c^2 \sin(\theta(t)), \\ \dot{y}(t) &= c^1 \sin(\theta(t)) + c^2 \cos(\theta(t)), \\ \dot{\theta}(t) &= c^3. \end{cases} \tag{24}$$

We only solve this for the initial condition $\gamma(t) = (0, 0, 0)$, as we already know that all the other curves can be obtained from left-translations. The solution to this system for $c^3 \neq 0$ is given by

$$\begin{cases} x(t) &= \frac{1}{c^3} \left(c^1 \sin(\theta(t) + c^2 \left(\cos(\theta(t)) - 1\right)\right), \\ y(t) &= \frac{1}{c^3} \left(c^1 \left(1 - \cos(\theta(t))\right) + c^2 \sin(\theta(t))\right), \\ \theta(t) &= c^3 t, \end{cases} \tag{25}$$

and for $c^3 = 0$ it is given by $\gamma(t) = (c^1 t, c^2 t, 0)$, i.e. straight lines in the spatial domain. Using some standard trigonometric identities, we can find a formula that holds for all cases, which is the following expression

$$e^{c^i A_i} = \begin{pmatrix} \left(c^1 \cos(c^3/2) - c^2 \sin(c^3/2)\right) \operatorname{sinc}(c^3/2) \\ \left(c^1 \sin(c^3/2) + c^2 \cos(c^3/2)\right) \operatorname{sinc}(c^3/2) \\ c^3 \end{pmatrix}. \tag{26}$$

We call exponential curves with $c^2 = 0$ horizontal exponential curves, recall this notion from the introduction. Next we will discuss some aspects of Riemannian geometry on Lie groups for $\mathbb{M}_2$. Let ω_i denote the dual elements to $\mathcal{A}_i$. From the relation

$$\begin{pmatrix} \mathcal{A}_1 \\ \mathcal{A}_2 \\ \mathcal{A}_3 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \partial_x \\ \partial_y \\ \partial_\theta \end{pmatrix}, \tag{27}$$

we can easily see that ω^i can be expressed in terms of $dx, dy, d\theta$ as follows

$$\omega^1 = \cos(\theta)dx + \sin(\theta)dy,\ \omega^2 = -\sin(\theta)dx + \cos(\theta)dy,\ \omega^3 = d\theta. \tag{28}$$

We will primarily be interested in metrics that have the following form

$$\mathcal{G} = g_{ii}\omega^i \otimes \omega^i, \tag{29}$$

with $g_{ii} > 0$ constant, i.e. we consider diagonal metrics w.r.t. the covectors ω^i. We will use $\mathcal{G}$ to exclusively denote metrics of this form. This metric is in fact a left-invariant metric, which is due to the fact that $\mathcal{A}_i$ are left-invariant and g_{ii} are constant. We will denote the distance induced by this metric with $d_{\mathcal{G}}$. For $p \in \mathbb{M}_2$, we will use the following short notation: $d_{\mathcal{G}}(p) = d_{\mathcal{G}}(p, p_0) = d_{\mathcal{G}}(p, (0, 0, 0))$. In the introduction we talked about anisotropic Riemannian metrics. We can now define what we mean by this. For metrics $\mathcal{G}$ of this particular form we define the (spatial) anisotropy by

$$\zeta := \frac{\max(g_{11}, g_{22})}{\min(g_{11}, g_{22})}. \tag{30}$$

A highly anisotropic metric thus means $\zeta \gg 1$.

The last thing we need to discuss about $SE(2)$ is an alternative coordinate system, that we will use later on. Since the exponential mapping always is invertible in a neighborhood of the identity element e, it has an inverse on this neighborhood denoted by log. In the case of $SE(2)$, the exponential map is even surjective so that we can use the logarithm to define coordinates on the entire manifold by identifying $T_eSE(2)$ with $\mathbb{R}^3$ through the basis A_i. We denote these coordinates with c, and we thus have that

$$c(g) = (c^1(g), c^2(g), c^3(g)) \quad \text{such that} \quad e^{c^i(g)A_i} = g. \tag{31}$$

We have the explicit formula for these coordinates below.

$$(c^1(g), c^2(g), c^3(g)) \quad = \left(\tfrac{x\cos(\theta/2)+y\sin(\theta/2)}{\operatorname{sinc}(\theta/2)}, \tfrac{-x\sin(\theta/2)+y\cos(\theta/2)}{\operatorname{sinc}(\theta/2)}, \theta\right) \tag{32}$$

We furthermore define the modified logarithmic coordinates, or half-angle coordinates b as

$$(b^1(g), b^2(g), b^3(g)) \quad = (x\cos(\theta/2) + y\sin(\theta/2), -x\sin(\theta/2) + y\cos(\theta/2), \theta)\,. \tag{33}$$

These have the nice property that the Euclidean norm of the spatial part remains the same, i.e.

$$\sqrt{|b^1|^2 + |b^2|^2} = \sqrt{x^2 + y^2}, \tag{34}$$

which is expected, as b^1, b^2 are just rotated x, y coordinates.

3 PDE-G-CNNs

In order to be able to answer the questions we posed in the introduction, we need to provide some more details about PDE-G-CNNs. As we mentioned in the introduction, the PDE-G-CNNs replace the usual activation functions found in CNNs by solutions to a specific evolution PDE, recall Figures 1 and 2. Figure 3 showed an example of the whole architecture of a PDE-G-CNN for a specific line detection in images. This specific use case will be introduced later, for now we want to make some of the ideas mentioned in the introduction more concrete.

In order to do this, let us introduce some notation. We index the depth (the number of layers) of the network with l, and denote the width (the number of channels) at layer l with M_l. In layer l we thus have M_l evolution PDEs, which are parametrized by parameters $\theta_{l,j}$ for $j = 1, ..., M_l$. Moreover we have the linear weights and biases $a_{l,ij}$ and $b_{l,i}$, for $i = 1, ..., M_{l+1}$ and $j = 1, ..., M_l$. The effect of the PDE layer l on inputs $(U_{l,c})_{c=1}^{M_l}$ can then be described by

$$U_{l+1,i} = \sum_{j=1}^{M_l} a_{l,ij} \Phi_{T,\theta_{l,j}}(U_{l,j}) + b_{l,i}. \tag{35}$$

Here we use $\Phi_{T,\theta}(U)$ to denote the solution operator of the PDE parametrized by parameters θ, at time T, with initial condition U. Note that we keep the end time T fixed for all channels and layers. As trainable weights we thus have the affine transformation weights a_{ij} and b_i, as well as the PDE parameters θ_j. Note that PDE-G-CNNs can consist of other layers besides these PDE layers, notably the possible lifting and projecting layer we see in Figure 3.

We will now introduce the definition we will be using for equivariance of the network, briefly discuss the lifting layer, and then discuss the precise evolution PDE that is used. It will be (part of) this PDE and its solution that will be of interest for us in the next section.

3.1 Equivariance

The way we define equivariance constraints for a network is based on homogeneous spaces and actions from Lie groups on them. We introduced homogeneous spaces in Section 2.2.2. Let X, Y be homogeneous spaces of some Lie group G. The input space of a PDE-G-CNN is modeled as a function space on X, say all bounded functions $B(X)$. Similarly, we have that the output space is $B(Y)$. The network, which we denote by $\mathcal{N}$, can then be seen as an operator transforming functions on X to functions on Y, i.e. $\mathcal{N} : B(X) \to B(Y)$.

Recall from Section 2.2.2 that each homogeneous space has some transitive group action of G associated with it. Such an action gives rise to a natural left action of G on $B(X)$. This action, denoted by $\mathcal{L}_g$ for $g \in G$, is called the left-regular representation. For $f \in B(X)$ and $p \in X$ it is given by

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

Of course we also have such an action on $B(Y)$. We now define equivariance of the network in terms of these actions. We require that for all $g \in G$

$$\mathcal{N} \circ \mathcal{L}_g = \mathcal{L}_g \circ \mathcal{N}. \tag{36}$$

We want to note that although we denoted both actions with $\mathcal{L}_g$, they do not need to be the same. Given that $\mathcal{N}$ consists of a composition of l layers, we see that each layer, and in particular each solution operator $\Phi_{T,\theta}$ needs to be equivariant if we want the total network to be equivariant. If the network has a lifting and projection layer, we of course also need these to be equivariant.

What does this equivariance look like for some familiar cases? Let $X = Y = \mathbb{R}^2$. These are of course Lie groups under the usual addition operation, and thus also (trivial) homogeneous spaces of $\mathbb{R}^2$. The left-regular representation then does nothing more than moving the input around, i.e. for $x \in \mathbb{R}^2$ equation (36) tells us that the network needs to satisfy

$$\mathcal{N}(f(\cdot + x)) = (\mathcal{N}f)(\cdot + x),$$

thus shifting inputs shifts the output in the same way.

We are interested in the case of roto-translation equivariance. Again consider $X = Y = \mathbb{R}^2$, but now as homogeneous spaces over $SE(2)$. We have seen that $SE(2)$ acts on $\mathbb{R}^2$ in Section 2.3.1. Again considering equation (36), which tells us that equivariance of the network means that for $g = (\boldsymbol{x}, R_\theta) \in SE(2)$ we need

$$\mathcal{N}(f(R_\theta^{-1}(\cdot - \boldsymbol{x})) = (\mathcal{N}f)(R_\theta^{-1}(\cdot - \boldsymbol{x})),$$

which is exactly the rotation-translation equivariance that we would like.

3.2 Lifting Layer

We want to briefly mention the way images are lifted to $\mathbb{M}_2$. For more details about how to do this, see [25].

Of course, we want this lifting to be done equivariant, similar to the requirement of 36, if the network is required to be equivariant as well. There is a general way of doing this between homogeneous spaces that is trainable. We will detail this for our case of lifting from $\mathbb{R}^2$ to $\mathbb{M}_2$. For this, we need a kernel function $k \in L^1(\mathbb{R}^2)$. Then, given an image $f \in L^2(\mathbb{R}^2)$, we can map f to an element in $\mathbb{M}_2$ by means of the operator $\mathcal{W}_k$ in the following way

$$(\mathcal{W}_k f)(g) := (\mathcal{L}_g k, f)_{L^2(\mathbb{R}^2)} = \int_{\mathbb{R}^2} (\mathcal{L}_g k)(\boldsymbol{x}) f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \tag{37}$$

In the implementation of this lifting, the kernel function is discretized and represented by some array, the values of which are trainable as parameters of the network.

Since most of the time, the output of the network will not be in $\mathbb{M}_2$, we also need some way to project back onto $\mathbb{R}^2$. For this, we can simply use a max-projection over the orientation part. Given some $f : \mathbb{M}_2 \to \mathbb{R}$, we project by

$$\mathcal{P}(f)(\boldsymbol{x}) := \max_{\theta \in [0, 2\pi)} f(\boldsymbol{x}, \theta). \tag{38}$$

This projection is also equivariant, which is again necessary if we want the entire network to be. Lastly, we want to mention that we can of course use (37) in a non-trainable manner by fixing the k. In Figure 12, we can see the effect of lifting an image and projecting back down again. We see there is some loss of information when lifting and projecting, but only minimal.

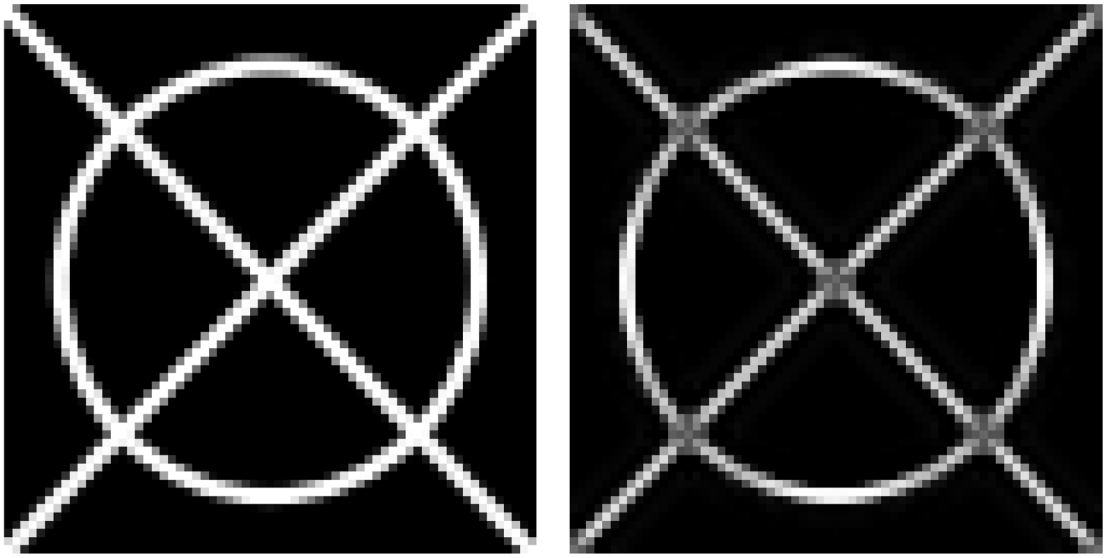


Figure 12: Left is the original image, on the right is the image after being lifted to $\mathbb{M}_2$ using cake wavelets ([25]) as kernel, and max projected down to $\mathbb{R}^2$ again.

3.3 PDE Activation Functions

We will now discuss the evolution PDE, and its (approximate) solution operator in more detail. The evolution equation that is solved can be seen in equation (39) below.

$$\begin{cases} \frac{\partial W}{\partial t}(\boldsymbol{p},t) &= -\boldsymbol{c}W(\boldsymbol{p},t) \\ &\quad - \left(-\Delta_{\mathcal{G}_1}\right)^{\alpha} W(\boldsymbol{p},t) \\ &\quad + ||\nabla_{\mathcal{G}_2^+} W(\boldsymbol{p},t)||^{2\alpha}_{\mathcal{G}_2^+} \qquad \text{for } p \in \mathbb{M}_2,\ t \geq 0, \\ &\quad - ||\nabla_{\mathcal{G}_2^-} W(\boldsymbol{p},t)||^{2\alpha}_{\mathcal{G}_2^-} \\ W(\boldsymbol{p},0) &= f(\boldsymbol{p}). \end{cases} \tag{39}$$

Here $\boldsymbol{c}$ is a left-invariant vector field on $\mathbb{M}_2$, $\mathcal{G}_1, \mathcal{G}_2^+, \mathcal{G}_2^-$ are left-invariant metrics on $\mathbb{M}_2$, and $\alpha \in [\frac{1}{2}, 1]$. The left-invariant vector field and the left-invariant metrics form the trainable parameters θ of the PDE. In particular, we have seen that a left-invariant vector field is determined by three coefficients, and we only consider left-invariant metrics on $\mathbb{M}_2$ as in equation (29), so that each metric also has three trainable parameters. The solution operator Φ of this PDE forms the activation functions of PDE-G-CNNs, as in equation (35). Also note that due to the left-invariance, the solution operator of this PDE will be equivariant, as we required.

The four terms in this PDE each have a distinct effect, recall Figure 2. The first term is a *convection* term, which moves the data over exponential curves. The second term is a (fractional) *diffusion* term, which regularizes data. The third term is a *dilation* term, which corresponds to (soft) max pooling of the data, and the fourth term is an *erosion* term, corresponding to (soft) min pooling. The parameter α controls how strong the diffusion and pooling is. The effect of all of these terms separately can be seen in Figure 13, and the combined effect of convection, dilation and erosion on the test image from Figure 12 can be seen in Figure 14.

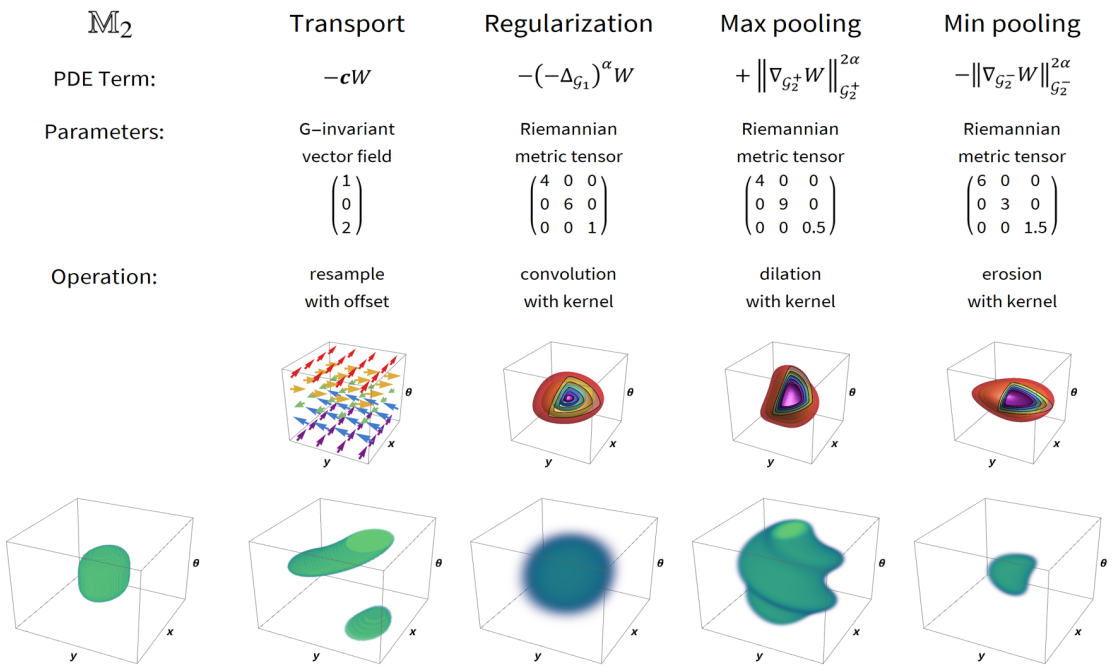


Figure 13: A figure showing the various effects each of the PDE terms have on the cylinder in the bottom left. Taken from [24] with permission.

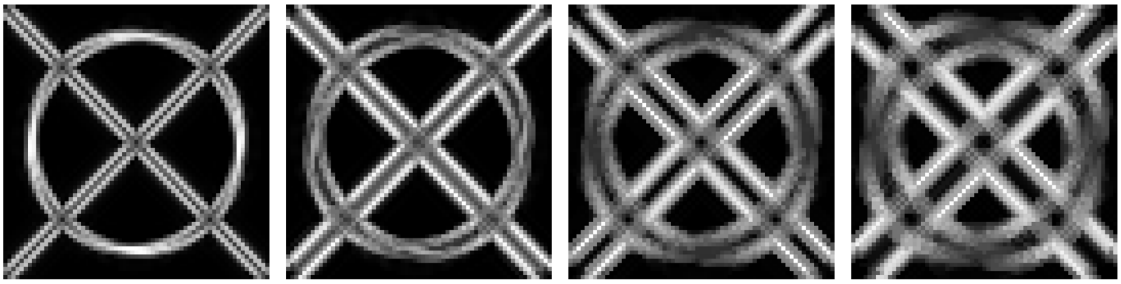


Figure 14: A figure showing the effects of convection, dilation and erosion on the lifted image from Figure 12, with increasingly stronger convection, dilation and erosion. The lifted images were max-projected down to $\mathbb{R}^2$ again. The convection, dilation and erosion applied is the same as is done separately in Figures 15, 17 and 16.

The way that this PDE is solved in the PDE-G-CNNs is via a principle called *operator splitting*, which involves composing the solution operator for the four terms separately for a small time step Δt, until we have reached the desired final time T. Note that operator splitting only produces an approximation, and in general will not yield the exact solution. We will not further concern ourselves with the error of this approximation.

In order to be able to apply operator splitting, we need to be able to solve the four separate evolution PDEs. We will discuss each of these next, with a particular focus on the erosion/dilation PDE.

3.3.1 Convection

The convection part of the PDE looks like

$$\begin{cases} \frac{\partial W}{\partial t}(\boldsymbol{p},t) & = -\boldsymbol{c}W(\boldsymbol{p},t) \\ W(\boldsymbol{p},0) & = f(\boldsymbol{p}). \end{cases} \tag{40}$$

We already encountered this PDE in 2.31, and have seen that the solution is given by transportation along exponential curves, which we already explicitly calculated for $\mathbb{M}_2$. Figure 15 shows the effect of convection on the lifted image from Figure 12. As mentioned, the left-invariant vector field $\boldsymbol{c}$ is completely determined by a single vector in the tangent space at the identity, so that we interchangeably use $\boldsymbol{c}$ to denote $\boldsymbol{c} = c^i A_i$. The coefficients c^i are the trainable parameters for this PDE.

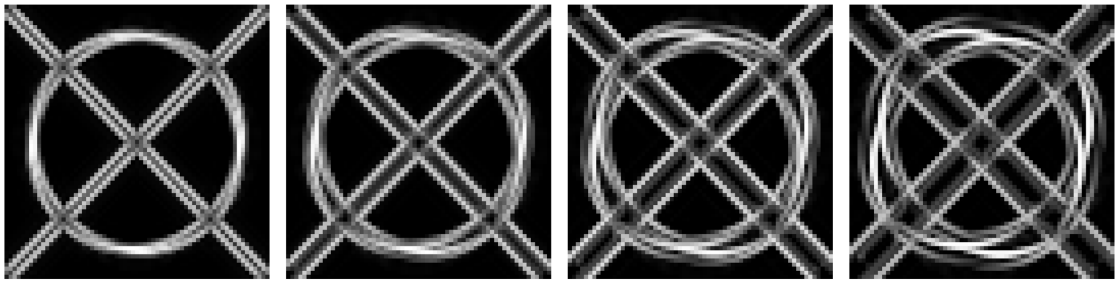


Figure 15: The effect of numerical convection on the lifted image from Figure 12, max projected afterwards. Denoting the vector field used in the convection by (c^1, c^2, c^3), representing $c^i \mathcal{A}_i$, we used in order $(1,1,2), (2,2,2), (3,3,2), (4,4,2)$.

3.3.2 Diffusion

The diffusion part of the PDE looks like

$$\begin{cases} \frac{\partial W}{\partial t}(\boldsymbol{p},t) & = -\ (-\Delta_{\mathcal{G}})^{\alpha}\, W(\boldsymbol{p},t) \\ W(\boldsymbol{p},0) & = f(\boldsymbol{p}). \end{cases} \tag{41}$$

As is the case with the Euclidean variant, the solution of this is given by a convolution with specific kernels. This convolution is now a convolution in the homogeneous space, which is defined below.

Definition 3.1. Let $f \in L^2(\mathbb{M}_2)$ and $k \in L^1(\mathbb{M}_2)$. The group convolution of f with k is then given by the integral

$$(k *_{\mathbb{M}_2} f)(p) := \int_{\mathbb{R}^2} \int_0^{2\pi} k(g^{-1}p) f(g) d\boldsymbol{x} d\theta,$$

where $g = (x, y, \theta)$ in the integrand.

The solution to 41 is given by the group convolution of f with some kernel K_t^{α} [58]. However, the analytic expressions for these kernels are too difficult to use in practice. In [24] approximations are proposed that are more practical. However, previously done experiments showed that diffusion seemed to have a negative effect on the performance, so in the experiments in [24] no diffusion is included. We will also not further include diffusion in our experiments.

3.3.3 Dilation and Erosion

The dilation/erosion part of the PDE looks like

$$\begin{cases} \frac{\partial W}{\partial t}(\boldsymbol{p},t) & = \pm||\nabla_{\mathcal{G}^\pm} W(\boldsymbol{p},t)||^{2\alpha}_{\mathcal{G}^\pm} \\ W(\boldsymbol{p},0) & = f(\boldsymbol{p}). \end{cases} \tag{42}$$

The results of these PDEs will be so called *morphological scale spaces* [59]. We will for now only consider the $-$ variant, corresponding to erosion. We will show later how the solution of the erosion PDE corresponds to the dilation variant.

We first introduce some notation. Define the Hamiltonian function by

$$\mathcal{H}_\alpha : T^*(\mathbb{M}_2) \to \mathbb{R}, \quad (p,\hat{p}) \mapsto ||\hat{p}||^{2\alpha}_{\mathcal{G}^*}.$$

Here, $||\cdot||_{\mathcal{G}^*}$ denotes the norm on the cotangent bundle, induced by the norm $\mathcal{G}$. Note that $||dW||_{\mathcal{G}^*} = ||\nabla_{\mathcal{G}} W||_{\mathcal{G}}$, by construction. We can thus write the erosion PDE as

$$\begin{cases} \frac{\partial W}{\partial t}(\boldsymbol{p},t) & = -\mathcal{H}_\alpha(dW) \\ W(\boldsymbol{p},0) & = f(\boldsymbol{p}). \end{cases} \tag{43}$$

The solution to such a problem depends on the *Lagrangian* of the Hamiltonian, which is obtained by taking the Fenchel transform. The Lagrangian $\mathcal{L}_\alpha$ in our case is given by

$$\begin{aligned} &\mathcal{L}_\alpha : TM \to \mathbb{R} \\ &\mathcal{L}_\alpha(p,\bar{p}) = \sup_{\hat{p}\in T^*_P(M)} (\hat{p},\bar{p}) - H_\alpha(p,\hat{p}) \\ &\qquad = \frac{2\alpha-1}{(2\alpha)^{2\alpha/(2\alpha-1)}} ||\bar{p}||^{\frac{2\alpha}{2\alpha-1}}_{\mathcal{G}^*} \\ &\qquad = \nu_\alpha ||\bar{p}||^{\frac{2\alpha}{2\alpha-1}}_{\mathcal{G}^*} \\ &\qquad = \mathcal{L}^{1D}_\alpha(||\bar{p}||_{\mathcal{G}^*}), \end{aligned}$$

for $\alpha \in (\frac{1}{2}, 1]$. For $\alpha = \frac{1}{2}$ we will see below what happens. The last thing we need to characterize the solution to the erosion PDE is the concept of an *infimal convolution* or *morphological convolution* between two functions on $\mathbb{M}_2$, defined as follows

Definition 3.2. Let $f, h : \mathbb{M}_2 \to \mathbb{R}$ be given. Then their infimal convolution $f \,\square\, h : \mathbb{M}_2 \to \mathbb{R}$ is defined as

$$(f \,\square\, h)(p) = \inf_{g\in SE(2)} \left\{ f(g^{-1}p) + h(gp_0) \right\}. \tag{44}$$

The following proposition then tells us what the solution of (40) is, in terms of an infimal convolution.

Proposition 3.3. *Let $\alpha \in [\frac{1}{2}, 1]$. The viscosity solutions W_α of the exact morphological erosion scale space PDE* (43) *are given by*

$$W_\alpha(p,t) = \inf_{q\in\mathbb{M}_2, \gamma\in\Gamma_t(p,q)} f(q) + \int_0^t \mathcal{L}_\alpha(\gamma(s),\dot{\gamma}(s))\,\mathrm{d}s \tag{45}$$

$$= \inf_{q\in\mathbb{M}_2} f(q) + t\mathcal{L}^{1D}_\alpha(d_{\mathcal{G}}(p,q)/t) \tag{46}$$

$$= (k^\alpha_t \,\square\, f)(p),, \tag{47}$$

where the morphological kernel k_t^α *is defined as*

$$k_t^\alpha(p) = t\mathcal{L}_\alpha^{1D}(d_{\mathcal{G}}(p,p_0)/t) = \nu_\alpha t|d_{\mathcal{G}}(p,p_0)/t|^{\frac{2\alpha}{2\alpha-1}}, \tag{48}$$

if $\alpha \in (\frac{1}{2}, 1]$. *For* $\alpha = \frac{1}{2}$ *we have the kernel*

$$k_t^{\frac{1}{2}}(p) = \begin{cases} 0 & \text{if } |d_{\mathcal{G}}(p,p_0)/t| \leq t \\ \infty & \text{else.} \end{cases} \tag{49}$$

Proof.
See [24, Theorem 5.21], or the reference given therein [60]. □

Remark 3.4. *The solution to the dilation variant of* (42) *is solved almost the same as the erosion variant. We have that* $W_\alpha(p,t) = -(k_t^\alpha \,\square\, -f)(p)$.

So in order to implement this in the PDE-G-CNNs, we would need to be able to calculate the Riemannian distance map $d_{\mathcal{G}}$. However, numerical solvers for this distance map cannot be efficiently parallelized for running on GPUs. So, this is not practical for in a machine learning setting. Instead, an approximation to $d_{\mathcal{G}}$ is proposed, which is easily computable, and is instead used to calculate the kernel. This is the so called *logarithmic metric estimate* ρ, which can be defined on general homogeneous spaces. On $\mathbb{M}_2$ we denote this approximation by ρ_c, due to its relation with the logarithmic coordinates c that were introduced in equation (31). It is given by the following formula

$$\rho_c(\cdot) := \sqrt{g_{11}|c^1(\cdot)|^2 + g_{22}|c^2(\cdot)|^2 + g_{33}|c^3(\cdot)|^2}. \tag{50}$$

The numerical morphological convolution in the PDE-G-CNNs is done with the kernel $k_{t,c}^\alpha$, given by

$$k_{t,c}^\alpha(p) = t\mathcal{L}_\alpha^{1D}(\rho_c(p)/t). \tag{51}$$

In the next section, we will have a look at a slightly different approximation for $d_{\mathcal{G}}$, based on the half-angle coordinates b, defined in equation (33). We denote this approximation by ρ_b, and it is given by

$$\rho_b(\cdot) := \sqrt{g_{11}|b^1(\cdot)|^2 + g_{22}|b^2(\cdot)|^2 + g_{33}|b^3(\cdot)|^2}. \tag{52}$$

We will denote the approximate kernel based on this distance by $k_{t,b}^\alpha$, which, similar to (51), we get by replacing $d_{\mathcal{G}}$ with ρ_b in (49). The main topic of the next section will be to find out how good an approximation ρ_b actually is to the true distance $d_{\mathcal{G}}$. In [24], they already show that locally ρ_c and $d_{\mathcal{G}}$ behave the same, and we will try to find more precise statements for ρ_b and $d_{\mathcal{G}}$. We will discuss why we look at ρ_b and not at the original approximation ρ_c below as well.

In Figures 16 and 17 we see some examples of the effect of dilation and erosion on an image.

The last thing we want to mention is the supervisors of this project have proven the fact that all of these distances, $d_{\mathcal{G}}, \rho_b$ and ρ_c satisfy certain symmetries. In particular, they all have the eight symmetries ε_i, corresponding to flipping signs in the half-angle or logarithmic coordinate system, as shown in Table 1. For ρ_b and ρ_c this is clear, however for $d_{\mathcal{G}}$ this is not immediately clear. Because ρ_b and ρ_c are meant to approximate $d_{\mathcal{G}}$, it is nice to know that they are symmetric in the same way.

	ε_0	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ε_7
b^1, c^1	$+$	$+$	$-$	$-$	$-$	$-$	$+$	$+$
b^2, c^2	$+$	$-$	$+$	$-$	$+$	$-$	$+$	$-$
b^3, c^3	$+$	$+$	$+$	$+$	$-$	$-$	$-$	$-$

Table 1: The symmetries that ρ_b, ρ_c and $d_{\mathcal{G}}$ have, that is they are invariant under composition with these functions.

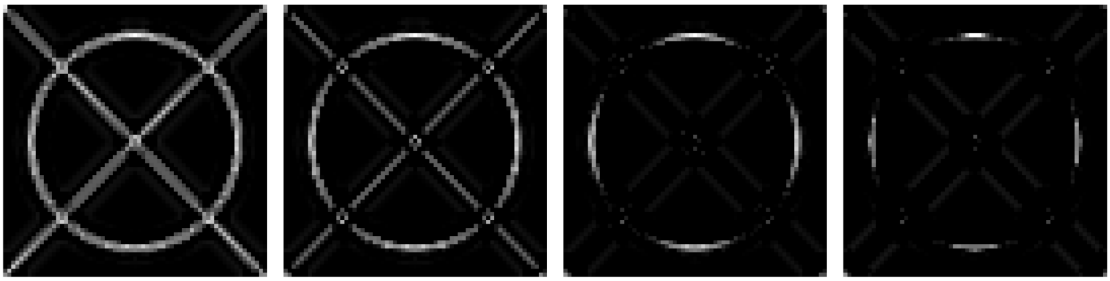


Figure 16: The effect of numerical erosion on the lifted image from Figure 12, max projected afterwards. Denoting the metrics we used by (g_{11}, g_{22}, g_{33}), we used in order $(1,1,1), (1,0.5,1), (1,0.25,1), (1,0.125,1)$.

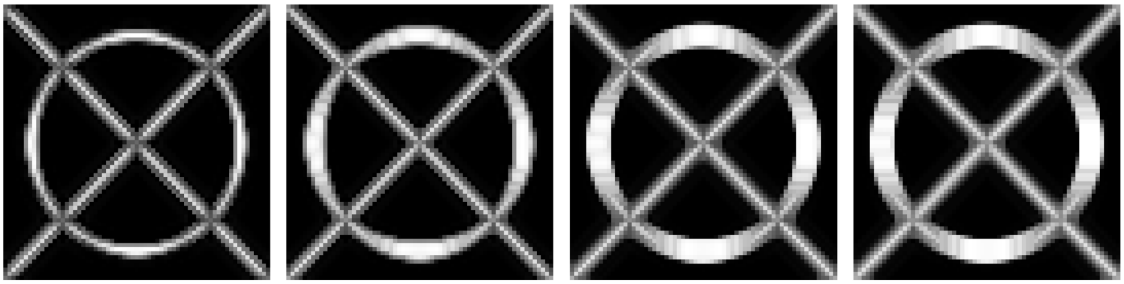


Figure 17: The effect of numerical dilation on the lifted image from Figure 12, max projected afterwards. Denoting the metrics we used by (g_{11}, g_{22}, g_{33}), we used in order $(1,1,1), (0.5,1,1), (0.25,1,1), (0.125,1,1)$.

4 Asymptotics

We have seen that a crucial aspect of the PDE-G-CNNs is the activation function, which solves PDEs of the form (39). By proposition 3.3, we have seen that the solution of the dilation and erosion part of this PDE is given by morphological convolution with a specific kernel. This kernel is completely determined by the Riemannian distance, which is learned by the network. The approximate kernel, that is actually used in the implementations, is obtained by replacing the Riemannian distance with the logarithmic estimate (equation (50)). Therefore, in order to study how well we are approximating the solution to the PDE, it suffices to look at how good the logarithmic approximation approximates the Riemannian distance. In [24] the following lemma was proven, relating this approximation and the true distance

Lemma 4.1 ([24] Lemma 5.7)**.** *There exists a $C \geq 1$ such that*

$$d_{\mathcal{G}}(p) \leq \rho_c(p) \leq C d_{\mathcal{G}},$$

for all $p \in \mathbb{M}_2$ in a compact neighborhood around p_0, away from the cut-locus.

This lemma tells us that locally around p_0 the distances are equivalent. However, we do not know anything about the constant C. From numerical experiments, we expect that this constant C heavily depends on the geometry of $\mathbb{M}_2$, i.e. on the metric parameters g_{ii}. The behaviour that was observed was that if g_{11} is close to g_{22}, the approximation is close to $d_{\mathcal{G}}$, while if g_{11} differs much from g_{22} the approximation becomes worse.

Looking at ρ_b for a second, we can see that this observation makes sense, as from equation (88) in Appendix A we see that if $g_{11} = g_{22}$, we actually have that $\rho_b = d_{\mathcal{G}}$. We also know that ρ_b and ρ_c are similar, at least locally. The remainder of this section will try to make Lemma 4.1 more precise. We will however consider the approximation ρ_b, not ρ_c. The reason for this is exactly the behaviour that $d_{\mathcal{G}} = \rho_b$ in the spatially isotropic case, and the fact that ρ_b is easier to work with.

Recall that we defined the anisotropy of the metric $\mathcal{G}$ as

$$\zeta = \frac{\max(g_{11}, g_{22})}{\min(g_{11}, g_{22})}. \tag{53}$$

It seem reasonable to suspect that the quality of ρ_b as an approximation to $d_{\mathcal{G}}$ will depend on ζ, as if $\zeta = 1$ they are equal.

We will make use of order behaviour in this section, and in order to be as precise as possible we want to provide the definition of this here.

Definition 4.2. If we write $f = g + \mathcal{O}(h)$ what we mean is that

$$\exists_{U, C>0} \forall_{x \in U} \quad |f(x) - g(x)| \leq C h(x),$$

where U is some set containing 0. so as we get closer to 0, the difference between f and g is bounded by some constant times h.

Where there can be confusion about manipulations with this order symbol we will fall back on this definition to make sure all statements are well-backed. In particular, for inequalities we have to be a little careful. In particular we will use a couple of times that if $f = g + \mathcal{O}(h)$ we have that

$$g(x) - Ch(x) \leq f(x) \leq g(x) + Ch(x),$$

for some $C > 0$ and for x in some neighborhood around 0. This is of course equivalent to the definition given above.

The remainder of this section looks as follows. We first provide some global relations relating $d_{\mathcal{G}}$ and ρ_b. Next, we give a motivating example of how we might approach finding such a bound as in Lemma 4.1, by looking at general Riemannian manifolds using normal coordinates. Next we present the first way we constructed a new bound using Taylor expansion of the metric coefficients. We then present a second bound, based on the Eikonal PDEs the distances satisfy. Lastly, we compare these two new bounds.

4.1 Simple Global Bounds

We start by relating ρ_b and ρ_c, so that we can see how similar they are. For this we have the following lemma.

Lemma 4.3. *Let $p \in \mathbb{M}_2$. We then have that*

$$\rho_b(p) \leq \rho_c(p) \leq \frac{\rho_b(p)}{\operatorname{sinc}(\theta/2)}.$$

Proof.
Note that $|\operatorname{sinc}| \leq 1$, so that we find that

$$\begin{aligned}
\rho_b(p)^2 &= g_{11}|b^1(p)|^2 + g_{22}|b^2(p)|^2 + g_{33}|b^3(p)|^2 \\
&= g_{11}|c^1(p)\operatorname{sinc}(\theta/2)|^2 + g_{22}|c^2(p)\operatorname{sinc}(\theta/2)|^2 + g_{33}|c^3(p)|^2 \\
&\leq g_{11}|c^1(p)|^2 + g_{22}|c^2(p)|^2 + g_{33}|c^3(p)|^2 \\
&= \rho_c(p)^2.
\end{aligned}$$

For the other inequality we calculate

$$\begin{aligned}
\rho_c(p)^2 &= g_{11}|c^1(p)|^2 + g_{22}|c^2(p)|^2 + g_{33}|c^3(p)|^2 \\
&= \frac{g_{11}|c^1(p)\operatorname{sinc}(\theta/2)|^2 + g_{22}|c^2(p)\operatorname{sinc}(\theta/2)|^2 + g_{33}|c^3(p)\operatorname{sinc}(\theta/2)|^2}{\operatorname{sinc}(\theta/2)^2} \\
&= \frac{g_{11}|b^1(p)|^2 + g_{22}|b^2(p)|^2 + g_{33}|b^3(p)\operatorname{sinc}(\theta/2)|^2}{\operatorname{sinc}(\theta/2)^2} \\
&\leq \frac{g_{11}|b^1(p)|^2 + g_{22}|b^2(p)|^2 + g_{33}|b^3(p)|^2}{\operatorname{sinc}(\theta/2)^2} \\
&= \frac{\rho_b(p)^2}{\operatorname{sinc}(\theta/2)^2}.
\end{aligned}$$

□

Next we present some global bounds and relations for the approximate distance which will help us later on. We start with a simple global bound for the exact distance $d_{\mathcal{G}}$, based on the observation that $d_{\mathcal{G}}$ and ρ_b are the same if $g_{11} = g_{22}$.

Lemma 4.4. *Let $g_m = \min(g_{11}, g_{22}), g_M = \max(g_{11}, g_{22})$ and $p = (x, y, \theta) \in \mathbb{M}_2$. We then have that*

$$l(p) := \sqrt{g_m x^2 + g_m y^2 + g_{33}\theta^2} \leq d_G(p) \leq \sqrt{g_M x^2 + g_M y^2 + g_{33}\theta^2} =: u_1(p)$$

Proof.
Consider the metric $\mathcal{G}_m = g_m\omega^1 \otimes \omega^1 + g_m\omega^2 \otimes \omega^2 + g_{33}\omega^3 \otimes \omega^3$. By definition, for any point $g \in SE(2)$ and any tangent vector $X \in T_gSE(2)$ we have that $||X||_{\mathcal{G}_m} \leq ||X||_{\mathcal{G}}$. This clearly implies that the length of any curve will be less when measured with $\mathcal{G}_m$ compared to $\mathcal{G}$. From this it follows that $d_{\mathcal{G}_m}(p) \leq d_{\mathcal{G}}(p)$.

Finally, it follows from equation (87) in the Appendix that the induced distance $d_{\mathcal{G}_m}$ can be written as

$$d_{\mathcal{G}_m}(p) = \sqrt{g_m x^2 + g_m y^2 = g_{33}\theta^2}.$$

A similar argument shows the other inequality for u. □

Using this, we can show the following global equivalence result between ρ_b and $d_{\mathcal{G}}$.

Lemma 4.5. *We have that ρ_b and $d_{\mathcal{G}}$ are equivalent with constant $\sqrt{\zeta}$, i.e.*

$$\frac{1}{\sqrt{\zeta}} d_{\mathcal{G}} \leq \rho_b \leq \sqrt{\zeta} d_{\mathcal{G}}.$$

Proof.
We are going to show this by relating l and u_1 to ρ_b.

$$\begin{aligned} g_m\rho_b(p)^2 &= g_m g_{11} b^1(p)^2 + g_m g_{22} b^2(p)^2 + g_m g_{33} b^3(p)^2 \\ &\leq g_m g_M b^1(p)^2 + g_m g_M b^2(p)^2 + g_M g_{33} b^3(p)^2 \\ &= g_m g_M x^2 + g_m g_M y^2 + g_M g_{33} b^3(p)^2 \\ &= g_M l(p)^2, \end{aligned}$$

which shows $\rho_b(p) \leq \sqrt{\zeta} l(p)$. Using similar manipulations we can get the inequality $\frac{1}{\sqrt{\zeta}} u_1(p) \leq \rho_b(p)$. From these inequalities, together with Lemma 4.4 we get the chain of inequalities

$$\frac{1}{\sqrt{\zeta}} d_{\mathcal{G}} \leq \frac{1}{\sqrt{\zeta}} u_1(p) \leq \rho_b \leq \sqrt{\zeta} l(p) \leq \sqrt{\zeta} d_{\mathcal{G}}(p).$$

□

4.2 Motivating Example

Recall from the preliminaries section the notion of normal coordinates. This is a coordinate system, centered around some fixed chosen point, such that the metric around this point resembles an Euclidean metric with a second order correction term given in terms of the curvature of the manifold. In this coordinate system we moreover have the following result, which tells us that the distance from the base point to any other point is actually given by the Euclidean norm in this coordinate system. We give a short proof of this fact below, as well as a way to show this only up to fourth order. The idea in this last proof will also be used to find a bound between ρ_b and $d_{\mathcal{G}}$ in the next subsection.

Lemma 4.6. *Let $(M, \mathcal{G})$ be a Riemannian manifold. Let $p_0 \in M$ be given, and let (U, y) be a normal neighborhood of p_0, i.e. $y = (y^1, \dots y^n)$ are normal coordinates in an open neighborhood U of p_0. For $p \in U \setminus cut(p_0)$ we have that*

$$d_{\mathcal{G}}(p, p_0)^2 = \|y(p)\|^2_{\mathcal{G}} \tag{54}$$

Proof.
Let $\gamma : [0,1] \to M$ denote the arc-length parametrized geodesic between p and p_0. The geodesic equation [52, Lemma 4.1.1] gives $\nabla_{\dot\gamma}\dot\gamma = 0$ w.r.t. the Levi-Civita connection ∇, i.e. geodesics are auto-parallel w.r.t. this connection. The fact that this connection is metric compatible gives us that

$$\frac{d}{dt}\|\dot\gamma(t)\|^2_{\gamma(t)} = 2\,\mathcal{G}_{\gamma(t)}(\dot\gamma(t), \nabla_{\dot\gamma(t)}\dot\gamma(t)) = 0,$$

so the length of the minimizing geodesic from $\gamma(0) = p_0$ to p is $\|\dot\gamma(t)\|_{\gamma(t)} = \|\dot\gamma(0)\|_p = \|y(p)\|$. □

Proof.
Next we show this behaviour, but only up to fourth order. This serves as a motivating example for the upcoming subsection. We are going to use the Taylor expansion of the metric coefficients expressed in normal coordinates. The distance squared is given by

$$d^2(p,p_0) = \int_0^1 \left\| \frac{d}{dt}\gamma(t) \right\|^2_{\mathcal{G}} dt,$$

with $\gamma : [0,1] \to M$ the constant speed geodesic from p_0 to p. If we express the curve γ in the normal coordinates y^i, we have by definition of those coordinates that

$$\gamma^i(t) = t\; y^i(p) \text{ and } \dot\gamma^i(t) = y^i(p). \tag{55}$$

Using these coordinates we can express the distance as:

$$d^2(p,p_0) = \int_0^1 \sum_{i,j=1}^n g_{ij}|_\gamma\, \dot\gamma^i\dot\gamma^j \;\mathrm{d}t = \int_0^1 \sum_{i,j=1}^n g_{ij}(\gamma(t))\, y^i(p)\, y^j(p)\;\mathrm{d}t. \tag{56}$$

In this expression we can make use of Lemma 2.15, giving us

$$= \int_0^1 \left(\delta_{ij} - \frac{1}{3}R_{ikjl}\gamma^k(t)\gamma^l(t) + \mathcal{O}(||y||^4)\right) y^i(p)\, y^j(p)\;\mathrm{d}t,$$

and making use of equation (55) we find

$$= \int_0^1 \left(\delta_{ij} - \frac{t^2}{3}R_{ikjl}\, y^k(p) y^l(p) + \mathcal{O}(||y||^4)\right) y^i(p)\, y^j(p)\;\mathrm{d}t.$$

Now the dependence on t is explicit and it is trivial to evaluate the integral

$$= ||y(p)||^2 - \frac{1}{9}R_{ikjl}\, y^k(p)\, y^l(p)\, y^i(p)\, y^j(p) + \mathcal{O}(||y||^4).$$

However, since the Riemannian curvature tensor is anti-symmetric in its first two and last two components, the summation in the second term here will be equal to 0, so that this is actually equal to

$$= ||y(p)||^2 + \mathcal{O}(||y||^4).$$

Since the expansion here only takes into consideration the first few terms, we cannot conclude the exact relation we found above. However, one can continue the Taylor expansion of the metric coefficients as was done in Lemma 2.15. The next terms will also depend, in more complicated ways, on the curvature tensor, and so it is likely that they will also not contribute due to the anti-symmetry. □

4.3 Taylor Expansion Bound

Inspired by the previous section, we want to do a Taylor expansion of the metric on $\mathbb{M}_2$ to relate it to the approximate norm ρ_b. However, we cannot actually use the normal coordinate system, so that the previous results are not directly applicable. Instead, we will do a Taylor expansion of the metric in the b coordinate system, as these coordinates relate directly to ρ_b.

First off we have a small lemma about the difference between geodesics and straight lines in the b coordinate system, which we will need in the actual theorem.

Lemma 4.7. *Let $p_0 \in \mathbb{M}_2$ be our fixed reference element. Let U be any compact neighborhood of p_0. Let $p \in U$ be arbitrary, let $\gamma(p,\cdot) : [0,1] \to M$ denote the geodesic curve from p_0 to p and let $\eta(p,\cdot) : [0,1] \to M$ denote the straight line in b coordinates between p_0 and p. We denote the b coordinates of the curves by γ^i and η^i. We have that, for $t \in [0,1]$,*

$$\begin{aligned} \ddot{\gamma}^k(p,t) &= \ddot{\eta}^k(p,t) + \mathcal{O}(\rho_b^2(p)), \\ \dot{\gamma}^k(p,t) &= \dot{\eta}^k(p,t) + \mathcal{O}(\rho_b^2(p)), \\ \gamma^k(p,t) &= \eta^k(p,t) + \mathcal{O}(\rho_b^2(p)). \end{aligned} \tag{57}$$

In particular, what we mean by this is that

$$\exists_{r>0, C>0} \forall_{p \in U, t \in [0,1]} : |\ddot{\gamma}^k(p,t) - \ddot{\eta}^k(p,t)| \leq C\rho^2(p),$$

and similarly for the other two statements.

Proof.

We start of by showing that $\dot{\gamma}^k(p,t) = \mathcal{O}(d_{\mathcal{G}}(p))$. We can see this as follows. First of all, we always parametrize the geodesic by constant velocity from $[0,1]$, so that $||\dot{\gamma}(t)||_{\mathcal{G}} = d_{\mathcal{G}}(p)$ for all $t \in [0,1]$. Fix some $t \in [0,1]$. We then know that in $T_{\gamma(t)}SE(2)$, $\mathcal{G}|_{\gamma(t)}$ is a coercive bi-linear form, hence we can bound $||\dot{\gamma}(t)||_{\mathcal{G}} \geq \xi(t)|\dot{\gamma}(t)|$, the Euclidean norm in the coordinates. From this we get that

$$|\gamma^i(t)|^2 \leq |\dot{\gamma}(t)|^2 \leq \frac{1}{\xi(t)}||\dot{\gamma}(t)||_{\mathcal{G}}^2 \leq \xi^* d_{\mathcal{G}}^2(p),$$

with $\xi^* = \max_{t\in[0,1]} \frac{1}{\xi(t)} < \infty$. So, we have $\dot{\gamma}^k = \mathcal{O}(d_{\mathcal{G}})$. From our choice of U and Lemma 4.5 we then also know that $\dot{\gamma}^k = \mathcal{O}(\rho)$, and hence $\dot{\gamma}^k\dot{\gamma}^j = \mathcal{O}(\rho^2)$.

Since γ is a geodesic, it satisfies the geodesic equations, given by

$$\ddot{\gamma}^i + \Gamma^i_{jk}(\gamma(t))\dot{\gamma}^k\dot{\gamma}^j = 0.$$

Note that the Christoffel symbols are smooth functions, as they are defined in terms of the metric coefficients, and so in particular they are bounded on the compact set U i.e., we have $\Gamma^i_{jk} = \mathcal{O}(1)$. Now from the geodesic equation, together with the fact that $\dot{\gamma}^k = \mathcal{O}(d_{\mathcal{G}})$ and $\Gamma^i_{jk}(\gamma(t)) = \mathcal{O}(1)$ we thus have that $\ddot{\gamma}^i = \mathcal{O}(\rho^2(p))$. Now since η is a straight line in this coordinate system, it has zero second derivative, so we conclude that $\ddot{\gamma}^i - \ddot{\eta}^i = \mathcal{O}(\rho^2(p))$. In particular this thus means that there exists some constant $C > 0$ such that for all p close to p_0, we have that

$$|\ddot{\gamma}^i(p,t) - \ddot{\eta}^i(p,t)| \leq C\rho^2(p).$$

Notice that, since γ^i is a smooth curve connecting $p_0^i = 0$ with p^i from $t = 0$ to 1, by the mean value theorem there exists some t^* such that $\dot{\gamma}^i(t^*) = p^i$. Moreover, since $\dot{\eta}^i(t) = p^i$ for all

$t \in [0,1]$, the function $\dot{\gamma}^i - \dot{\eta}^i$ thus has (at least) one zero in the interval $[0,1]$, namely at t^*. Since its derivative is bounded, as was shown above, and we are working in a compact interval, $\dot{\gamma}^i - \dot{\eta}^i$ is Lipschitz (in t) with constant $C\rho^2(p)$, and so we can conclude that

$$\begin{aligned} |\dot{\gamma}^i(t) - \dot{\eta}^i(t)| &= |\dot{\gamma}^i(t) - \dot{\eta}^i(t) - \left(\dot{\gamma}^i(t^*) - \dot{\eta}^i(t^*)\right)| \\ &\leq C\rho^2(p)|t - t^*| \\ &\leq C\rho^2(p), \end{aligned}$$

meaning that $\dot{\gamma}^i - \dot{\eta}^i = \mathcal{O}(\rho^2(p))$. Since the function $\gamma^i - \eta^i$ also has a zero in $[0,1]$ (at least two), we can apply the same argument as above to find that

$$|\gamma^i - \eta^i| \leq C\rho^2(p),$$

or in other words $\gamma^i = \eta^i + \mathcal{O}(\rho^2)$.

□

Next we present the main theorem of this subsection. It gives a bound on the Riemannian distance on $\mathbb{M}_2$, in terms of ρ_b. The introduced anisotropy ζ of the metric plays an important role in the quality of this bound.

Theorem 4.8. *Let $U \subset \mathbb{M}_2$ be any compact neighborhood of p_0. For all $p \in U$ we have that*

$$\boxed{\rho_b^2(p)\left(1 - \frac{\rho_b(p)^2}{6}\left(\frac{4\zeta - 3}{g_{33}}\right) - C\rho_b(p)^3\right) \leq d_{\mathcal{G}}(p)^2,}$$

for some $C > 0$.

Proof.

We start by fixing some $p \in U$. We denote the b coordinates of the arc-parametrized geodesic $\gamma : [0,1] \to \mathbb{M}_2$ between p_0 and p by γ^i, and similarly for the straight line in the b coordinates, η^i. We start out by Taylor expanding the metric components of $\mathcal{G}$, expressed in the b coordinate system. We will denote these by g_{ij}. In view of Remark 2.14, we have to choose a connection in order to be able to define the Hessian. The connection we choose is the Levi-Cevita connection of the metric that induces ρ_b, that is the constant diagonal metric

$$g_{ii} db^i \otimes db^i,$$

where g_{ii} denote the same coefficients as in equation (29). In practice this results in us being able to just compute the Hessian as usual, as the Christoffel symbols vanish because we have constant metric coefficients in the b coordinate system for this metric.

We refer to Appendix A for the details and exact formulas for this Taylor expansion. We will use them with a reference to the Appendix below. So we end up with the following expansion for our metric components

$$g_{ij}(p) = \tilde{g}_{ij} + g_{ij,k}(p_0)b^k(p) + \frac{1}{2}H_{ijkl}(p_0)b^k(p)b^l(p) + \mathcal{O}(\rho_b(p)^3), \tag{58}$$

where $\tilde{g}_{ij}$ is the constant diagonal matrix with g_{ii} on its diagonal. Now we are going to plug this into the definition of the squared distance, as we did in the second proof of Lemma 4.6. First recall that the squared distance is given by

$$d^2_{\mathcal{G}}(p, p_0) = \int_0^1 g_{ij}(\gamma(t))\dot{\gamma}^i(t)\dot{\gamma}^j(t)dt,$$

where γ is the geodesic from p_0 to p. Plugging the expansion into this we find

$$= \int_0^1 \left(\tilde{g}_{ij} + g_{ij,k}(p_0)\gamma^k(t) + \frac{1}{2}H_{ijkl}(p_0)\gamma^k(t)\gamma^l(t) + \mathcal{O}(\rho_b(p)^3) \right) \dot{\gamma}^i(t)\dot{\gamma}^j(t)dt. \tag{59}$$

We now consider each of the three terms of the Taylor expansion separately.

- The first term, containing $\tilde{g}_{ij}$, we are going to find a lower bound for, using a straight line in the coordinate space. In particular, we have that

  $$\int_0^1 \tilde{g}_{ij}\dot{\gamma}^i(t)\dot{\gamma}^j(t)dt \geq \inf_{\alpha} \int_0^1 \tilde{g}_{ij}\dot{\alpha}^i(t)\dot{\alpha}^j(t)dt.$$

  This infimum is attained by a straight line, i.e. $\alpha^i(t) = tb^i(p)$. Hence we get that $\dot{\alpha}^i = b^i(p)$ and so we find

  $$\begin{aligned} \int_0^1 \tilde{g}_{ij}\dot{\gamma}^i(t)\dot{\gamma}^j(t)dt &\geq \int_0^1 \tilde{g}_{ij}b^i(p)b^j(p)dt \\ &= \rho_b(p)^2. \end{aligned}$$

- The term $\int_0^1 g_{ij,k}(p_0)\gamma^k(t)\dot{\gamma}^i(t)\dot{\gamma}^j(t)dt$ is equal to 0. This is because of the fact that $d_{\mathcal{G}}$ is inversion invariant i.e., $d_{\mathcal{G}}(p) = d_{\mathcal{G}}(p^{-1})$, due to the left-invariance of the distance. Because inversion corresponds to replacing b^i with $-b^i$ in the current coordinate system, $d_{\mathcal{G}}$ is an even function, and so we conclude that all odd terms of the expansion have to be equal to 0.
- Lastly we take a look at the Hessian term in the expansion. We are going to bound this term in the expansion by

  $$\begin{aligned} \int_0^1 H_{ijkl}(p_0)\gamma^k(t)\gamma^l(t)\dot{\gamma}^i(t)\dot{\gamma}^j(t)dt &\geq -\left| \int_0^1 H_{ijkl}(p_0)\gamma^k(t)\gamma^l(t)\dot{\gamma}^i(t)\dot{\gamma}^j(t)dt \right| \\ &\geq -\int_0^1 \left| H_{ijkl}(p_0)\gamma^k(t)\gamma^l(t)\dot{\gamma}^i(t)\dot{\gamma}^j(t) \right| dt. \end{aligned}$$

  What we would now like to do is replace the geodesic coordinates in this expression by the straight line coordinates, as for these we can actually calculate the integral. This is is similar to what happened in the second proof of Lemma 4.6, only there the geodesics were already straight lines. In view of Lemma 4.7 we are going to make some error that we can control.

  We will need Lemma 4.7 and the fact that $\dot{\gamma}^i = \mathcal{O}(\rho_b)$ and . We first show that $\gamma^k(t)\dot{\gamma}^i(t) = \eta^k(t)\dot{\eta}^i(t) + \mathcal{O}(\rho_b^3)$. We do this as follows

$$\begin{aligned}|\eta^k(t)\dot{\eta}^i(t) - \gamma^k(t)\dot{\gamma}^i(t)| &= |\eta^k(t)\dot{\eta}^i(t) - \eta^k(t)\dot{\gamma}^k(t) + \eta^k(t)\dot{\gamma}^k(t) - \gamma^k(t)\dot{\gamma}^i(t)| \\ &\leq |\eta^k(t)\dot{\eta}^i(t) - \eta^k(t)\dot{\gamma}^k(t)| + |\eta^k(t)\dot{\gamma}^i(t) - \gamma^k(t)\dot{\gamma}^i(t)| \\ &\leq |\eta^k(t)|\,|\dot{\eta}^i(t) - \dot{\gamma}^k(t)| + |\dot{\gamma}^i(t)|\,|\eta^k(t) - \gamma^k(t)| \\ &\leq C_1\rho_b(p)C_2\rho_b(p)^2 + C_3\rho_b(p)C_4\rho_b(p)^2 \\ &\leq C_5\rho_b(p)^3,\end{aligned}$$

where $C_i > 0$ are some constants, and where on the second to last line we used $\eta^k(t) = tb^k(p) = \mathcal{O}(\rho_b), \dot{\gamma}^i = \mathcal{O}(\rho_b)$ and Lemma 4.7. This shows us that we indeed have that $\gamma^k(t)\dot{\gamma}^i(t) = \eta^k(t)\dot{\eta}^i(t) + \mathcal{O}(\rho_b^3)$. We can apply the same trick again twice to conclude that

$$\left|\gamma^k(t)\gamma^l(t)\dot{\gamma}^i(t)\dot{\gamma}^j(t)\right| = |\eta^k\eta^l\dot{\eta}^i\dot{\eta}^j| + \mathcal{O}(\rho^5).$$

We thus have that

$$-\int_0^1 \left||H_{ijkl}(p_0)|\gamma^k(t)\gamma^l(t)\dot{\gamma}^i(t)\dot{\gamma}^j(t)\right| dt = -\int_0^1 ||H_{ijkl}(p_0)|\eta^k\eta^l\dot{\eta}^i\dot{\eta}^j|\,dt + \mathcal{O}(\rho_b(p)^5).$$

From this can then conclude that the Hessian term can be lower bounded, for some $C > 0$, by the expression

$$\begin{aligned}\int_0^1 H_{ijkl}(p_0)\gamma^k(t)\gamma^l(t)\dot{\gamma}^i(t)\dot{\gamma}^j(t)dt &\geq -\left|\int_0^1 H_{ijkl}(p_0)\gamma^k(t)\gamma^l(t)\dot{\gamma}^i(t)\dot{\gamma}^j(t)dt\right| \\ &\geq -\int_0^1 |H_{ijkl}(p_0)\gamma^k(t)\gamma^l(t)\dot{\gamma}^i(t)\dot{\gamma}^j(t)|dt \\ &\geq -\int_0^1 \sum_{i,j,k,l} |H_{ijkl}(p_0)\eta^k\eta^l\dot{\eta}^i\dot{\eta}^j|\,dt - C\rho^5(p).\end{aligned}$$

Now we use the special form that the exponential curves have in these exponential coordinates, to conclude that

$$\begin{aligned}\int_0^1 \sum_{i,j,k,l} |H_{ijkl}(p_0)\eta^k\eta^l\dot{\eta}^i\dot{\eta}^j|\,dt &= \int_0^1 t^2 \sum_{i,j,k,l} |H_{ijkl}(p_0)b^k(p)b^l(p)b^i(p)b^j(p)|dt \\ &= \frac{1}{3}\sum_{i,j,k,l} |H_{ijkl}(p_0)b^k(p)b^l(p)b^i(p)b^j(p)|.\end{aligned}$$

Now recall equation (59). Plugging in all the previous results allows us to conclude that

$$d^2(p,p_0) \geq \rho_b(p)^2 - \frac{1}{6}\sum_{i,j,k,l} |H_{ijkl}(p_0)b^k(p)b^l(p)b^i(p)b^j(p)| - C\rho^5(p).$$

Note that the error term of equation (58) becomes $\mathcal{O}(\rho_b(p)^5)$ as we multiply it with $\dot{\gamma}^i(t)\dot{\gamma}^j(t)$ in equation (59), and so we can combine it with the error term from the Hessian. Now we want to

explicitly write out the contraction with the Hessian. In Appendix A we explicitly calculate these Hessian terms and the sum. This is given by

$$|H_{ijkl}(p_0)b^k(p)b^l(p)b^i(p)b^j(p)| = \frac{1}{2}b^3(p)^2\big(g_{11}b^2(p)^2 + 3|g_{11}-g_{22}|\left(b^2(p)^2 + b^1(p)^2\right) + g_{22}b^1(p)^2\big).$$

We see that this is a quadratic form in the squared b-coordinates of the point p. We thus write this as

$$|H_{ijkl}(p_0)b^k(p)b^l(p)b^i(p)b^j(p)| = \frac{1}{4}(\boldsymbol{b}^2)^T M(\boldsymbol{b}^2), \tag{60}$$

with $\boldsymbol{b} = \boldsymbol{b}(p) = (b^1(p)^2, b^2(p)^2, b^3(p)^2)$ and the matrix M is given by

$$M = \begin{pmatrix} 0 & 0 & g_{11}+3|g_{11}-g_{22}| \\ 0 & 0 & g_{22}+3|g_{11}-g_{22}| \\ g_{11}+3|g_{11}-g_{22}| & g_{22}+3|g_{11}-g_{22}| & 0 \end{pmatrix}.$$

Combining all the things we found above we thus conclude that we have the following bound on $d_{\mathcal{G}}$:

$$d^2_{\mathcal{G}}(p) \geq \rho^2(p) - \frac{1}{24}(\boldsymbol{b}^2)^T M(\boldsymbol{b}^2) - C\rho^5(p). \tag{61}$$

We see that despite the fact that we know that if $\zeta = 1$ the Riemannian distance and ρ_b should be the same, this bound does not tell us that. We suspect that this is because of the way we lower-bounded the first term. That bound will not be tight if $g_{11} = g_{22}$, as exponential curves are always different from straight lines, which are determined by the metric.

We may equivalently write this bound as

$$d_{\mathcal{G}}(p)^2 \geq \rho_b(p)^2\left(1 - \varepsilon(p)\right), \tag{62}$$

where the error term is given by

$$0 < \varepsilon(p) = \frac{\rho^2}{24}\left(\frac{\boldsymbol{b}^2}{\rho_b(p)^2}\right)^T M\left(\frac{\boldsymbol{b}^2}{\rho_b(p)^2}\right) + C\rho^3(p). \tag{63}$$

Notice that $\frac{b^i(p)^2}{\rho_b(p)^2} \leq \frac{1}{g_{ii}}$, so that we can bound each component of the vector in the equation above. Let $g_m = \min(g_{11}, g_{22})$ and $g_M = \max(g_{11}, g_{22})$. We then have that

$$\begin{aligned}
\varepsilon(p) &\leq \frac{\rho^2}{12}\left(\frac{g_{11}+3|g_{11}-g_{22}|}{g_{22}g_{33}} + \frac{g_{22}+3|g_{11}-g_{22}|}{g_{11}g_{33}}\right) + C\rho^3(p) \\
&= \frac{\rho^2}{12}\left(\frac{g_{11}^2+3g_{11}|g_{11}-g_{22}|}{g_{11}g_{22}g_{33}} + \frac{g_{22}^2+3g_{22}|g_{11}-g_{22}|}{g_{11}g_{22}g_{33}}\right) + C\rho^3(p) \\
&\leq \frac{\rho^2}{12}\left(\frac{2g_M^2+6g_M|g_{11}-g_{22}|}{g_{11}g_{22}g_{33}}\right) + C\rho^3(p) \\
&= \frac{\rho^2}{6}\left(\frac{g_M^2+3g_M(g_M-g_m)}{g_m g_M g_{33}}\right) + C\rho^3(p) \\
&= \frac{\rho^2}{6}\left(\frac{4\zeta-3}{g_{33}}\right) + C\rho^3(p)
\end{aligned}$$

□

Recall that we were interested in these distances because they are used for solving certain parts in the PDE activation function (39). In particular, the distance is used to calculate kernels with which we take morphological convolutions. It is now easy to see how the error we found for the distance translates to the error in the kernel, as the following corollary shows.

Theorem 4.9. *Let $U \subset \mathbb{M}_2$ be a compact neighborhood of p_0. For $p \in U$ let $\varepsilon(p)$ denote a correction term such that*

$$\rho_b^2(p)\,(1-\varepsilon(p)) \leq d_{\mathcal{G}}^2(p).$$

We then have the following bound on the morphological kernels

$$\boxed{k_{t,b}^{\alpha}(p)\left(1-\frac{\alpha}{2\alpha-1}\varepsilon(p)+\mathcal{O}(\varepsilon(p)^2)\right) \leq k_t^{\alpha}(p).}$$

Proof.
Writing out the definition of the kernels we see that

$$\begin{aligned}
k_t^{\alpha}(p) &= t\nu_{\alpha}\left|\frac{d^2(p)}{t^2}\right|^{\frac{\alpha}{2\alpha-1}} \\
&\geq t\nu_{\alpha}\left|\frac{\rho^2(p)\,(1-\varepsilon(p))}{t^2}\right|^{\frac{\alpha}{2\alpha-1}} \\
&= \bar{k}_t^{\alpha}(p)\,(1-|\varepsilon(p)|)^{\frac{\alpha}{2\alpha-1}} \\
&= \bar{k}_t^{\alpha}(p)\left(1-\frac{\alpha}{2\alpha-1}\varepsilon(p)+\mathcal{O}(\varepsilon(p)^2)\right).
\end{aligned}$$

□

We thus see that our error gets multiplied with a factor $\frac{\alpha}{2\alpha-1}$. The behaviour that this error grows without bound as $\alpha \to \frac{1}{2}$ is expected, as for $\alpha \to \frac{1}{2}$ the kernels k_t^{α} also explode to ∞ in the region where $d_{\mathcal{G}} > t$.

From a practical point of view, Theorem 4.8 and Theorem 4.9 tell us that we require g_{33} to be large, and $\zeta \approx 1$ if we want the approximations to be reasonable. This is in line with what numerical experiments showed. These limitations of the Riemannian approximations were not addressed in [24, App.A]. It is however unfortunate that this bound does not reflect the behaviour that at $\zeta = 1$ the approximate distance is exact. Next, we will see a bound that does show this behaviour.

4.4 Dual Norm Bound

Next, we will present another method to find a bound of the same form as in Theorem 4.12, but now based on a different technique. It will turn out that this bound does tell us that ρ_b is equal to $d_{\mathcal{G}}$ if $\zeta = 1$. It is based on the following observation.

Lemma 4.10. *Let $p \in \mathbb{M}_2$ be given, and let $\gamma : [0,1] \to \mathbb{M}_2$ be the geodesic from p_0 to p. We then have that*

$$\rho_b(p) \leq d_{\mathcal{G}}(p) \max_{t\in[0,1]} ||d\rho_b|_{\gamma(t)}||_{G^*}$$

Proof.
To see this, consider the function $\rho \circ \gamma : [0,1] \to \mathbb{R}$. We have that

$$\int_0^1 (\rho \circ \gamma)' dt = \rho(p) - \rho(p_0) = \rho(p),$$

and also that

$$|(\rho \circ \gamma)'(t)| = |\langle d\rho|_{\gamma(t)}, \dot{\gamma}(t)\rangle| \leq ||d\rho|_{\gamma(t)|}||_{\mathcal{G}^*} \, ||\dot{\gamma}(t)||_{\mathcal{G}} \leq \max_{t^*\in[0,1]} ||d\rho|_{\gamma(t^*)}||_{\mathcal{G}^*} \, ||\dot{\gamma}(t)||_{\mathcal{G}}.$$

So we can conclude that

$$\rho(p) = \int_0^1 (\rho \circ \gamma)' \, dt \leq \int_0^1 ||\dot{\gamma}(t)||_{\mathcal{G}} \, dt \max_{t^*\in[0,1]} ||d\rho|_{\gamma(t^*)}||_{\mathcal{G}^*} = d_{\mathcal{G}}(p) \max_{t^*\in[0,1]} ||d\rho|_{\gamma(t^*)}||_{\mathcal{G}^*}.$$

□

We want to calculate and bound the quantity $\max_{t\in[0,1]} ||d\rho_b|_{\gamma(t)}||_{G^*}$ in such a way that we can get a useful bound out of this Lemma 4.10. The way we are going to do this is by first calculating $||d\rho|_p||_{\mathcal{G}^*}$ for arbitrary $p \in \mathbb{M}_2$, and using the fact that it satisfies an Eikonal PDE we are going to bound this expression. Lastly we are going to bound the maximum over the geodesic. First of, we are going to express the dual norm in half-angle coordinates.

4.4.1 Dual Norm in b Coordinates

Recall that the metric $\mathcal{G}$ is defined as a diagonal metric in terms of the frame $\mathcal{A}_i$, equation (29), so that we may easily calculate $||dV||^2_{G^*}$, for some function V, in this frame as

$$||dV||^2_{G^*} = g^{11}|\mathcal{A}_1 V|^2 + g^{22}|\mathcal{A}_2 V|^2 + g^{33}|\mathcal{A}_3 V|^2, \tag{64}$$

where $g^{ii} = g_{ii}^{-1}$. We can make this more explicit by writing it as

$$g^{11}|(\cos(\theta)\partial_x + \sin(\theta)\partial_y)V|^2 + g^{22}|(-\sin(\theta)\partial_x + \cos(\theta)\partial_y)V|^2 + g^{33}|\partial_\theta V|^2. \tag{65}$$

We can now rewrite the partial derivatives w.r.t. the fixed coordinates to partial derivatives w.r.t. the b coordinates. Starting with the first term, we get that

$$\begin{aligned}\cos(\theta)\partial_x V &= \cos(\theta)\left(\partial_{b^1} V \frac{\partial b^1}{\partial x} + \partial_{b^2} V \frac{\partial b_2}{\partial x}\right) \\ &= \cos(\theta)\left(\partial_{b^1} V \cos(\theta/2) - \partial_{b^2} V \sin(\theta/2)\right).\end{aligned}$$

$$\begin{aligned}\sin(\theta)\partial_y V &= \sin(\theta)\left(\partial_{b^1} V\frac{\partial b^1}{\partial y} + \partial_{b^2} V\frac{\partial b_2}{\partial y}\right)\\ &= \sin(\theta)\left(\partial_{b^1} V\sin(\theta/2) - \partial_{b^2} V\cos(\theta/2)\right).\end{aligned}$$

Using some trigonometric rules we deduce that the first term is equal to

$$\begin{aligned}g^{11}|\mathcal{A}_1 V|^2 &= g^{11}|\cos(\theta)\big(\partial_{b^1} V\cos(\theta/2) - \partial_{b^2} V\sin(\theta/2)\big) + \sin(\theta)\big(\partial_{b^1} V\sin(\theta/2) - \partial_{b^2} V\cos(\theta/2)\big)|^2\\ &= g^{11}|\partial_{b^1} V\left(\cos(\theta)\cos(\theta/2) + \sin(\theta)\sin(\theta/2)\right) + \partial_{b^2} V\left(-\cos(\theta)\sin(\theta/2) + \sin(\theta)\cos(\theta/2)\right)|^2\\ &= g^{11}|\cos(\theta/2)\partial_{b^1} V + \sin(\theta/2)\partial_{b^2} V|^2.\end{aligned}$$

Following the same procedure for the second term results in

$$g^{22}|\mathcal{A}_2 V|^2 = g^{22}|-\sin(\theta/2)\partial_{b^1} V + \cos(\theta/2)\partial_{b^2} V|^2. \tag{66}$$

The last term is slightly different, and we will again do this in detail. Again from the chain rule, we can rewrite the derivative with respect to θ in terms of derivatives with respect to b^i. We get

$$\begin{aligned}\partial_\theta V &= \partial_{b^3} V + \partial_{b^2} V\frac{\partial b^2}{\partial\theta} + \partial_{b^1} V\frac{\partial b_1}{\partial\theta}\\ &= \partial_{b^3} V + \frac{1}{2}\partial_{b^2} V(-x\cos(\theta/2) - y\sin(\theta/2)) + \frac{1}{2}\partial_{b^1} V(-x\sin(\theta/2) + y\cos(\theta/2))\\ &= \partial_{b^3} V - \frac{1}{2}b^1\partial_{b^2} V + \frac{1}{2}b^2\partial_{b^1} V\\ &= \partial_{b^3} V + \frac{1}{2}\partial_\psi V.\end{aligned}$$

Here we introduced the short-hand notation $\partial_\psi V := b^2\partial_{b^1} V - b^1\partial_{b^2} V$. It follows from these computations that we have that

$$||dV||^2_{G^*} = g^{11}|\cos(\theta/2)\partial_{b^1} V + \sin(\theta/2)\partial_{b^2} V|^2 + g^{22}|-\sin(\theta/2)\partial_{b^1} V + \cos(\theta/2)\partial_{b^2} V|^2 + g^{33}|\partial_{b^3} V + \frac{1}{2}\partial_\psi V|^2. \tag{67}$$

We rewrite this slightly to the following form

$$= g^{22}\left(\left|\frac{\partial V}{\partial b^1}\right|^2 + \left|\frac{\partial V}{\partial b^2}\right|^2\right) + (g^{11} - g^{22})\left|\cos(\frac{b^3}{2})\frac{\partial V}{\partial b^1} + \sin(\frac{b^3}{2})\frac{\partial V}{\partial b^2}\right|^2 + g^{33}\left|\frac{1}{2}\frac{\partial V}{\partial\psi} + \frac{\partial V}{\partial b^3}\right|^2. \tag{68}$$

4.4.2 Comparison between ρ_b and $d_{\mathcal{G}}$

The next thing we want to do, is to calculate the norm of $d\rho_b$. For this, we are going to use the following fact. The approximate distance ρ_b satisfies the Eikonal PDE

$$\begin{cases} g^{11}|\partial_{b^1}\rho_b|^2 + g^{22}|\partial_{b^2}\rho_b|^2 + g^{33}|\partial_{b^3}\rho_b|^2 &= 1,\\ \rho_b(0,0,0) &= 0.\end{cases} \tag{69}$$

This is something general distance maps satisfy [61, Theorem 6.24]. In our case, a quick verification of this fact can be done because we have an exact expression for ρ_b. We now only need one last intermediate result before presenting the main theorem of this subsection.

Lemma 4.11. *Let $p \in \mathbb{M}_2$ be given. Around p_0 we can bound the expression $||d\rho_b||^2_{\mathcal{G}^*}$ as follows*

$$||d\rho_b||^2_{\mathcal{G}^*} \leq 1 + \frac{\zeta}{g_{33}} \frac{\rho_b^2}{2} + \mathcal{O}(\theta^3).$$

Proof.
For convenience we denote ρ_b with ρ. We now first calculate the quantity $||d\rho|_{\gamma(t)}||^2_{G^*}$ for arbitrary p, which we can do using the dual norm that we rewrote in the previous section. In particular we have that

$$||d\rho|_p||^2_{\mathcal{G}^*} = g^{22}\left(\left|\frac{\partial\rho}{\partial b^1}\right|^2 + \left|\frac{\partial\rho}{\partial b^2}\right|^2\right) + (g^{11} - g^{22})\left|\cos(\frac{b^3}{2})\frac{\partial\rho}{\partial b^1} + \sin(\frac{b^3}{2})\frac{\partial\rho}{\partial b^2}\right|^2 + g^{33}\left|\frac{1}{2}\frac{\partial\rho}{\partial\psi} + \frac{\partial\rho}{\partial b^3}\right|^2. \quad (70)$$

What we will now do is first expand the square in the second term in (70), and then Taylor expand the various sine and cosine terms, so that we can rewrite the whole expression into a form where we can use that ρ_b satisfies (69). For this we need some standard Taylor series, presented here for convenience

$$\begin{aligned}
\cos^2(\theta/2) &= 1 - \frac{\theta^2}{4} + \frac{\theta^4}{48} + \mathcal{O}(\theta^6),\\
\sin^2(\theta/2) &= \frac{\theta^2}{4} - \frac{\theta^4}{48} + \mathcal{O}(\theta^6),\\
\sin(\theta) &= \theta - \frac{\theta^3}{6} + \frac{\theta^5}{120} + \mathcal{O}(\theta^6).
\end{aligned}$$

Expanding the square we find

$$\left|\cos(\frac{b^3}{2})\frac{\partial\rho}{\partial b^1} + \sin(\frac{b^3}{2})\frac{\partial\rho}{\partial b^2}\right|^2 = \cos^2(b^3/2)|\partial_{b^1}\rho|^2 + \sin(b^3)\partial_{b^1}\rho\partial_{b^2}\rho + \sin^2(b^3/2)|\partial_{b^2}\rho|^2. \quad (71)$$

From here we can plug in the Taylor expansions of the cosine and sine terms, to conclude that this term, up to $\mathcal{O}(\theta^6)$ is

$$\begin{aligned}
\left|\cos(\frac{b^3}{2})\frac{\partial\rho}{\partial b^1} + \sin(\frac{b^3}{2})\frac{\partial\rho}{\partial b^2}\right|^2 = {} & |\partial_{b^1}\rho|^2\\
& + \theta\left(\partial_{b^1}\rho\,\partial_{b^2}\rho\right)\\
& + \frac{\theta^2}{4}\left(|\partial_{b^2}\rho|^2 - |\partial_{b^1}\rho|^2\right)\\
& + \frac{\theta^3}{4}\left(-\partial_{b^1}\rho\,\partial_{b^2}\rho\right)\\
& + \frac{\theta^4}{48}\left(|\partial_{b^1}\rho|^2 - |\partial_{b^2}\rho|^2\right)\\
& + \frac{\theta^5}{120}\left(\partial_{b^1}\rho\,\partial_{b^2}\rho\right)\\
& + \mathcal{O}(\theta^6).
\end{aligned}$$

We can of course calculate the derivatives of ρ_b, so that we may also write this term as (adding the metric coefficient back in front that we left out)

$$(g^{11}-g^{22})\left|\cos(\frac{b^3}{2})\frac{\partial\rho}{\partial b^1}+\sin(\frac{b^3}{2})\frac{\partial\rho}{\partial b^2}\right|^2 = (g^{11}-g^{22})|\partial_{b^1}\rho|^2 + (g_{22}-g_{11})\left(\frac{b^1b^2b^3}{\rho^2}\right) + (g^{11}-g^{22})\frac{1}{4}\left(\frac{g_{22}^2(b^2)^2(b^3)^2-g_{11}^2(b^1)^2(b^3)^2}{\rho^2}\right) + \mathcal{O}(\theta^3).$$

We will leave the first term in (70) as is and now take a look at the third term. We first calculate the ∂_ψ term

$$\begin{aligned}\partial_\psi\rho(p) &= b^2(p)\partial_{b^1}\rho(p) - b^1(p)\partial_{b^2}\rho(p)\\ &= b^2(p)\frac{g_{11}b^1(p)}{\rho(p)} - b^1(p)\frac{g_{22}b^2(p)}{\rho(p)}\\ &= (g_{11}-g_{22})\frac{b^1(p)b^2(p)}{\rho(p)}.\end{aligned}$$

Similarly we find that $\partial_{b^3}\rho(p) = g_{33}b^3(p)/\rho(p)$. This would mean that the third term becomes

$$g^{33}\left|\frac{1}{2}\frac{\partial\rho}{\partial\psi}+\frac{\partial\rho}{\partial b^3}\right|^2 = \frac{(g_{11}-g_{22})^2}{4g_{33}}\frac{\left(b^1(p)b^2(p)\right)^2}{\rho(p)^2} + (g_{11}-g_{22})\frac{b^1(p)b^2(p)b^3(p)}{\rho(p)^2} + g^{33}\left|\frac{\partial\rho}{\partial b^3}\right|^2 \quad (72)$$

Collecting all results we calculated above, and making use of (69), we get that the dual norm of $d\rho_b$ is given by

$$||d\rho|_p||^2_{\mathcal{G}^*} = g^{11}|\partial_{b^1}\rho|^2 + g^{22}|\partial_{b^2}\rho|^2 + g^{33}|\partial_{b^3}\rho|^2 + \varepsilon = 1+\varepsilon, \quad (73)$$

where ε represents the error term that arises from the Taylor expansions done in the second term and the additional terms in the third term. More specifically, it is given by

$$\varepsilon = (g^{11}-g^{22})\frac{1}{4}\left(\frac{g_{22}^2(b^2)^2(b^3)^2-g_{11}^2(b^1)^2(b^3)^2}{\rho^2}\right) + \mathcal{O}(\theta^3) + \frac{(g_{11}-g_{22})^2}{4g_{33}}\frac{\left(b^1(p)b^2(p)\right)^2}{\rho(p)^2}.$$

Here the first line comes from the Taylor expansion, and the second line comes from the third term in (70). Notice that some terms have canceled in this error expression, specifically the "first order" terms. We can rewrite ε a little bit into

$$\varepsilon = \frac{g_{11}-g_{22}}{4g_{11}\,g_{22}\,g_{33}\,\rho^2}\left(g_{11}^2g_{33}(b^1b^3)^2 - g_{22}^2g_{33}(b^2b^3)^2 + g_{11}g_{22}(g_{11}-g_{22})(b^1b^2)^2\right) + \mathcal{O}(\theta^3). \quad (74)$$

In order to make this expression a bit more manageable, we introduce the normalized coordinates

$$\tilde{b}^i := \sqrt{g_{ii}}\frac{b^i}{\rho_b}. \tag{75}$$

This error term can then be written as

$$\varepsilon = \frac{\rho^2}{4}\frac{g_{11}-g_{22}}{g_{11}\,g_{22}\,g_{33}}\left(g_{11}(\tilde{b}^1\tilde{b}^3)^2 - g_{22}(\tilde{b}^2\tilde{b}^3)^2 + (g_{11}-g_{22})(\tilde{b}^1\tilde{b}^2)^2\right) + O(\theta^3). \tag{76}$$

We notice that $|\tilde{b}^i| \leq 1$ by construction, so that the above expression can be bounded by

$$\varepsilon \leq \frac{\rho^2}{4}\frac{|g_{11}-g_{22}|}{g_{11}\,g_{22}\,g_{33}}\left(g_{11}+g_{22}+|g_{11}-g_{22}|\right) + O(\theta^3). \tag{77}$$

Lastly we perform some algebraic manipulations on this expression. As before, let $g_m = \min(g_{11}, g_{22})$ and $g_M = (g_{11}, g_{22})$. Note that $g_{11} - g_{22} + |g_{11} - g_{22}| = 2g_M$, so that we may write this bound equivalently as

$$\begin{aligned}
\varepsilon &\leq \frac{\rho^2}{4}\frac{|g_{11}-g_{22}|}{g_{11}\,g_{22}\,g_{33}}2g_M + O(\theta^3) \\
&= \frac{\rho^2}{2}\frac{g_M - g_m}{g_M\,g_m\,g_{33}}g_M + O(\theta^3) \\
&= \frac{\rho^2}{2}\frac{\zeta-1}{g_{33}} + O(\theta^3),
\end{aligned}$$

proving the lemma.

□

We are now ready to present the main theorem, providing us with a similar result as Theorem 4.8, but providing sharper behaviour around $\zeta = 1$.

Theorem 4.12. *Let $U \subset \mathbb{M}_2$ be a compact neighborhood of p_0. Then for all $p \in U$ we have that*

$$\boxed{\rho^2(p)\left(1 - \frac{\rho_b^2}{2}\frac{\zeta^2(\zeta-1)}{g_{33}} - C\rho_b^3\right) \leq d_{\mathcal{G}}^2(p),}$$

for some $C > 0$.

Proof.

Let $p \in U$, and let $\gamma : [0,1] \to \mathbb{M}_2$ be the geodesic from p_0 to p. Notice that for the Riemannian distance we have that

$$d_{\mathcal{G}}(\gamma(s)) \leq d_{\mathcal{G}}(\gamma(t)), \text{ for } s \leq t. \tag{78}$$

Making use of Lemma 4.5 we can find a similar statement for ρ_b, namely that

$$\rho_b(\gamma(s)) \leq \zeta\rho_b(\gamma(t)), \text{ for } s \leq t. \tag{79}$$

From this we get that

$$\max_{t\in[0,1]} \rho_b(\gamma(t)) \leq \zeta\rho_b(\gamma(1)) = \zeta\rho_b(p). \tag{80}$$

Combining this with Lemma 4.10 and 4.11 we get the following bound

$$\rho_b^2(p) \leq d_{\mathcal{G}}^2(p)\left(1 + \frac{\rho_b^2}{2}\frac{\zeta^2(\zeta-1)}{g_{33}} + C\rho_b(p)^3\right), \tag{81}$$

for some $C > 0$, coming from the order term in Lemma 4.10. We can transform to the desired inequality by means of the geometric series.

□

Note that this bound tells us that when $\zeta = 1$, we indeed have that (up to third order) the two distances coincide, unlike the bound from Theorem 4.8. Also, just like the previous bound, it tells us that g_{33} needs to be large to have a good approximation. However, we do expect that this bound is less sharp for higher ζ, as it is a third order polynomial instead of the linear behaviour in ζ the other bound had. Also note that this bound is applicable to Theorem 4.9.

4.5 Comparison between the Bounds

The last thing we want to do in this section is to quickly compare the bounds from Theorem 4.8 and 4.12. We do this by plotting the expression we found for the (multiplicative) error term we found. Since $\frac{1}{g_{33}}$ is a multiplicative factor in both of these errors, we fix $g_{33} = 1$, and consider these errors as functions of ζ. Looking at the form of them, we see that the dual error is a third degree polynomial in ζ, while the Taylor error is linear in ζ, so we already know that at some point the Taylor error is going to be sharper. Both of the errors are plotted in Figure 18.

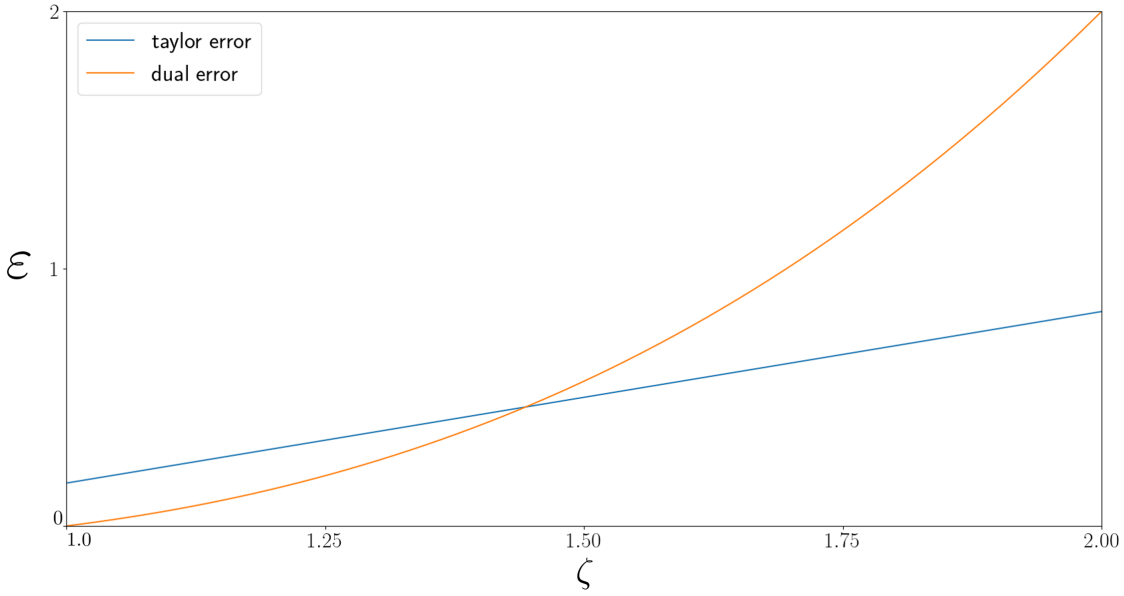


Figure 18: The errors from Theorem 4.8 ("Taylor error") and from Theorem 4.12 ("dual error"). As we expected, for small ζ the dual error is more accurate, while for larger ζ the Taylor error becomes better.

For practical purposes, what we will do is combine the two approximations by taking the minimum of the two. This ensures that the bound we use is as good as possible, both for ζ close to 1 and

large ζ. However, note that we do not know if these bounds are sharp. We want to end with the following remark about how we are going to use the bounds that we found in the next section for checking the numerical behaviour in the network.

Remark 4.13. *In order to check if in numerical experiments the approximations are reasonable, we do the following. We ignore the higher order terms in the error expressions in Theorem 4.8 and Theorem 4.12, and fix some a priori tolerance level for the multiplicative error term $|\varepsilon(p)| < \varepsilon_{tol}$. Define*

$$B = \min\left(\frac{\zeta^2(\zeta-1)}{g_{33}}, \frac{3}{24}\left(\frac{\zeta-1}{g_{33}} + \frac{\zeta}{3g_{33}}\right)\right).$$

We then consider all $p \in \mathbb{M}_2$ such that

$$\boxed{\rho_b^2(p) \leq \frac{\varepsilon_{tol}}{B},} \tag{82}$$

and compare this to the region where ρ_b is numerically sampled.

5 Visualization and Interpretation

Now that we have improved on Lemma 4.1 for the specific case of $\mathbb{M}_2$, an interesting thing to find out is if in trained PDE-G-CNNs, the learned geometry is actually suitable for the approximation ρ_b (or ρ_c). We will assess this by means of Remark 4.13, choosing some reasonable tolerance level.

Besides the obvious reasons this is interesting to check, i.e. to see if we are actually accurately solving the PDE we think we are, there is another reason why we might want to investigate the anisotropy of the networks. Recall the discussion from the introduction, where we related highly anisotropic metrics with association fields. Do PDE-G-CNNs need the same highly anisotropic metrics as the biological model? We have now seen that this might be a problem, as we have seen that we cannot guarantee that the approximation is good in this case. This would motivate the need for using a more suitable approximation in those cases.

We will attempt to answer these questions and some of the other questions posed in the introduction in this section. First of all, we introduce a new data set that we will use as a benchmark for all experiments. Next we investigate how the metric coefficients are configured in fully trained networks. Furthermore, we try to visualize the network with the hope of better understanding what is happening. Lastly, we perform some experiments to see if different approximations for $d_{\mathcal{G}}$ yield better performances.

5.1 Lines Data Set

All of the experiments will be done on a new data set. The task is to find a continuous line given only line segments of this line, as well as randomly placed other line segments. In Figure 19 we can see two examples of the "easy" variant of this data set, and in Figure 20 we see two examples of the "medium" variant. We will refer to this data set as the "Lines" data set.

The first reason we consider this data set is because we expect that PDE-G-CNNs will benefit from their architecture. In particular, in this data set we combine the need for local orientations (the line segments need to be aligned if they are part of the continuous curve) as well as morphological operations (stretching of the line segments in a particular direction, recall Figure 7). If the network behaves similar to the model of our visual system, we expect to see that the geometries learned for dilation and erosion are orthogonal to each other, which again relates to the idea of association fields and having a preference to align with a certain direction. This extending of lines is called "excitation" in neurogeometry, whereas the sharpening in the orthogonal direction is called "inhibition".

The second reason is a practical one, that is that these images are relatively easy to generate and interpret. We will have a look at intermediate output of the network a little later on, and the fact that these images are relatively simple still allows us to make some sense of them. It also allows us to work with relatively low resolution images, which from a computational viewpoint is nice.

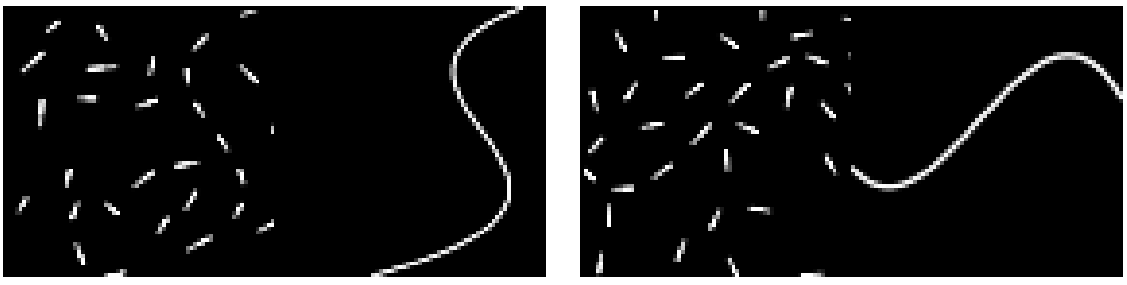


Figure 19: Two examples from the "easy" difficulty line data set. The left images is the input, the right image is the ground truth.

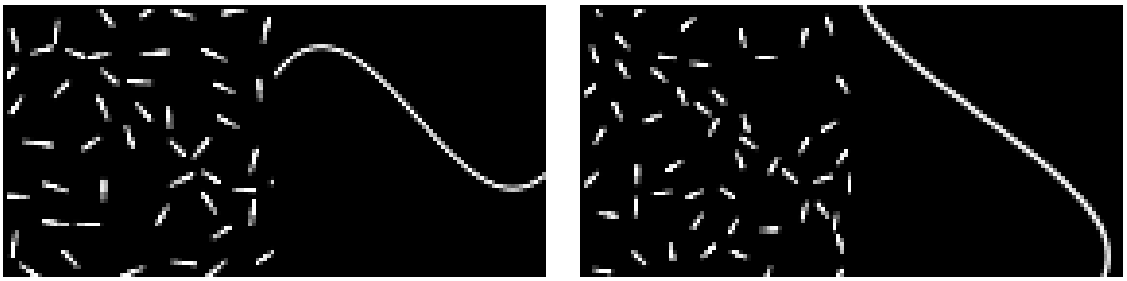


Figure 20: Two examples from the "medium" difficulty line data set. The left images is the input, the right image is the ground truth.

The ground truth is created by randomly sampling some points between chosen endpoints, which are on opposite sides of the unit square. We then fit a B-spline through these points to get a curve we can sample from. It is also very convenient that we can sample from the derivative of the curve, as this is what we use to give the correct orientation to the line segments drawn in the images. Lastly, we plot random line segments in the image, making sure that they do not overlap too much with the line segments that are already placed.

The difference between the easy and medium data set is that in the medium data set, more randomly placed line segments are present in the image, that are closer to each other and the ground truth line. The ground truth lines are generated in the same way for the easy and medium data set. For training we used 3000 images, and for testing 300 images.

5.2 Performance on Easy Data Set

We start off by training both a normal CNN and a PDE-G-CNN on the easy variant of the data set, Figure 19, to compare the performance of the two networks types. For this we use the `Lietorch` package for `pytorch`. This packages was introduced in [24], and is available at `https://gitlab.com/bsmetsjr/lietorch`. The exact configurations for all experiments can be found in Appendix B, and are based on the architectures used in [24].

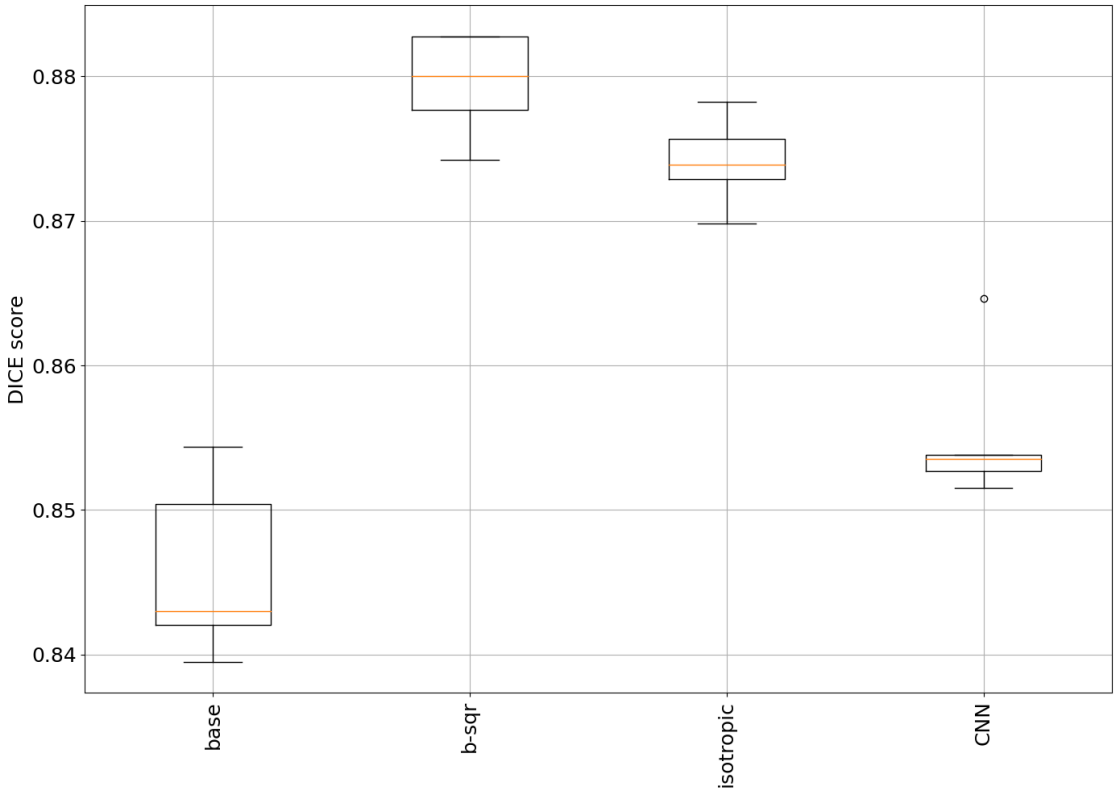


Figure 21: Performance metric for the "easy" line data set. For each network architecture 5 models were trained with the same configuration of hyper parameters, described in Appendix B.

In Figure 21 we can see the performance of the PDE-G-CNNs and the CNN trained on the easy line data set. The performance of the PDE-G-CNN can be seen in the "base" column, and the performance of the CNN in the "CNN" column. Both models perform well, reaching DICE scores above 0.84. However, overall the CNN still outperforms the PDE-G-CNN, contrary to what we might have hoped. That said, the amount of parameters needed in the PDE-G-CNNs is much less than what CNNs need: for this example we have 3424 total parameters for the PDE-G-CNN versus 10600 for the CNN.

Before addressing this performance gap, let us first take a look at the questions we asked ourselves in the beginning of this section: how does Remark 4.13 hold in the trained PDE-G-CNNs? For now we fix the tolerance $\varepsilon_{tol} = 0.5$. Recall that this is a multiplicative factor, and not directly determines the maximum distance between the approximation and true distance.

However, we run into some trouble if we want to calculate this bound, as the trained network parameters are passed through a ReLU function before being used to calculate ρ_c. This means that there are certain channels where $\min(g_{11}, g_{22}) = 0$, and we thus have an infinite anisotropy. We disregard these cases for a second, as there are relatively few. Now we have to check if the region where the PDE-G-CNNs sample the approximation is contained in the region we get from Remark 4.13. The situation is schematically depicted in Figure 22.

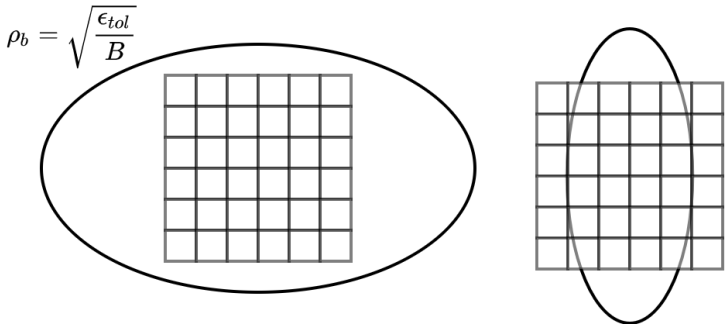


Figure 22: Figure depicting the two situations that can occur. Left is a situation in which the sampling, depicted by the grid, is completely inside of this region, while on the right we sample outside of this situation.

As it turns out, quite a substantial number of channels, around 85%, sample outside of the region determined by the bounds. We do have to remember that these PDE-G-CNNs use the ρ_c approximation instead of the ρ_b, but in view of Lemma 4.3 we do not expect that the actual scenario is much better.

Based on these observations we propose a couple of minor tweaks to the implementation of the PDE-G-CNNs. First of all, we suspect that performance is hindered by the fact that the ReLU function is forcing metric coefficients to 0 and staying at 0, so we replace the ReLU function by squaring the metric coefficients. This prevents them from being stuck at 0 if they happen to become negative during training. Also, we start using ρ_b instead of ρ_c, as we have seen that this might be a better candidate approximation, and it is easier to relate to the results that we have derived in this report. We will keep these changes for the remainder of this report.

In the "b-sqr" column of Figure 21, the performance after making these changes can be seen. We see that the performance has increased quite a bit, now also being higher then the CNNs performance. However, applying the same analysis as before, we again get that around 85% of the channels have configuration with errors (possibly) larger then ε_{tol}.

These observations seem to imply that anisotropy is needed for a fully trained network, so that it seems PDE-G-CNNs do need (approximately) the same geometry as the biological visual systems. However, in the "isotropic" column of Figure 21 we trained PDE-G-CNNs where we constrained each metric to $g_{11} = g_{22}$, the spatially isotropic case. We know that in this case, the Riemannian distance $d_{\mathcal{G}}$ and ρ_b coincide. Surprisingly, this network still performs really well, outperforming both the "base" model and the CNN.

Lastly, we want to show some of the outputs of the network to get an idea of how good these look. We are also going to look at some of the intermediate values of the network, since we hope that with PDE-G-CNNs, the network's behaviour can be explained better than for CNNs, which act as a "black box". While we do not claim that we can fully explain the network behaviour, these intermediate results offer a nice glimpse into what is happening. Figures 23 and 24 show these visualizations for the "base" model, using c-coordinates and ReLUs, and 25 shows this same visualization for a CNN trained on the easy line data set. Note that the intermediate outputs of the PDE-G-CNNs are projected to $\mathbb{R}^2$ using a max projection, as these of course are in reality living in $\mathbb{M}_2$. We also want to note that the scale of each intermediate output, i.e. the value corresponding to what is drawn as black and what as white, is different for each image, and so these colors cannot be compared between different images.

Looking at the intermediate output, we can see it appears to be the case that the PDE-G-CNN really is trying to create some sort of global line filter. In particular, looking at the last columns of Figures 23 and 24, we see that we get exactly something like we discussed in Figure 8 in the

introduction. We also see that this "filter" is only really present in the deeper layers, suggesting that it takes some time to combine the convection, dilation and erosion in something like this. We can compare this to the CNNs' intermediate output. The line now seems to suddenly appear at a certain depth, and the effect of creating a filter like the PDE-G-CNNs have is not present.

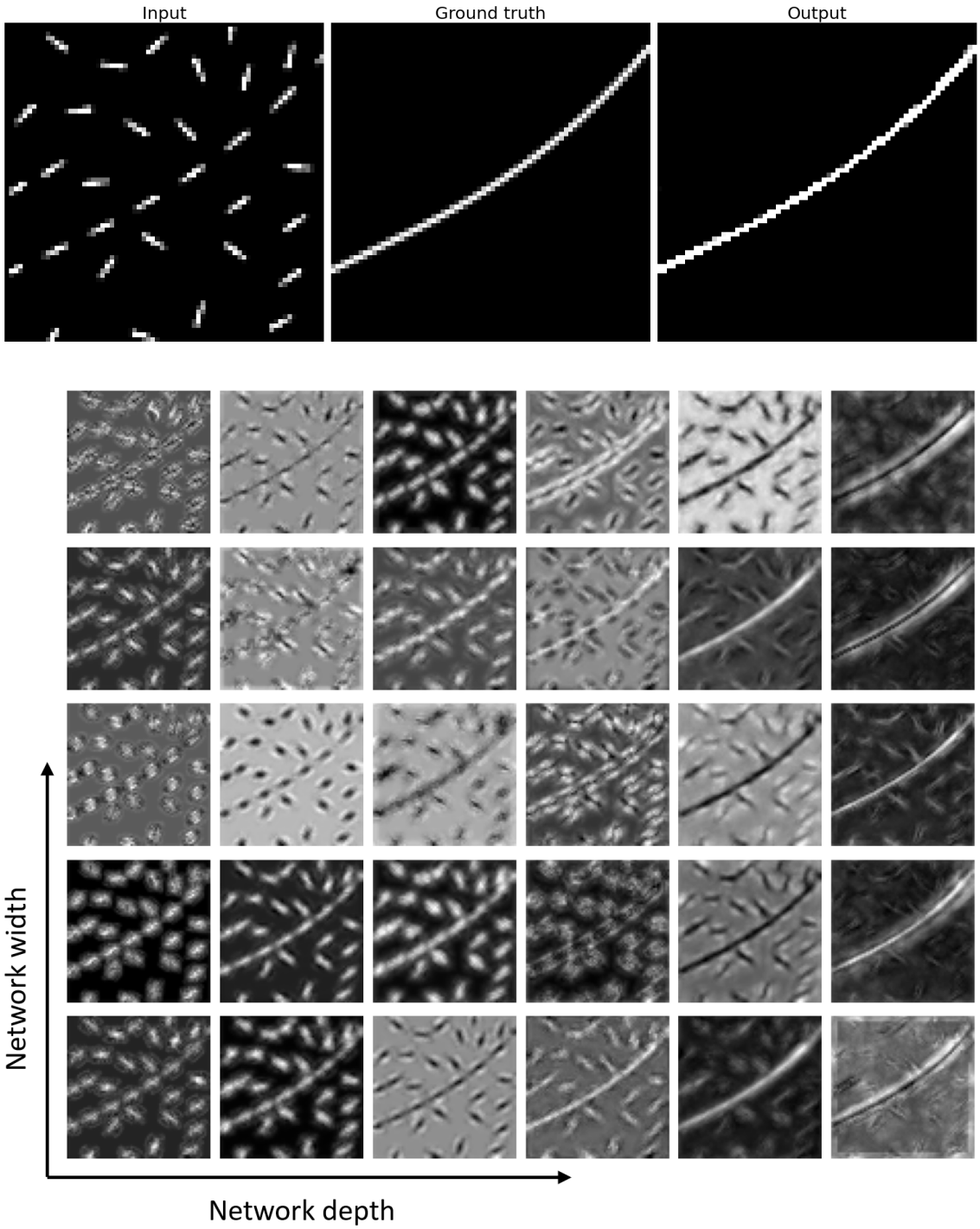


Figure 23: The input, ground truth and output of one test example. Below are some of the intermediate results from the network, columns correspond to layers and rows correspond to channels. All of these intermediate results are max-projected to $\mathbb{R}^2$ in order to visualize them.

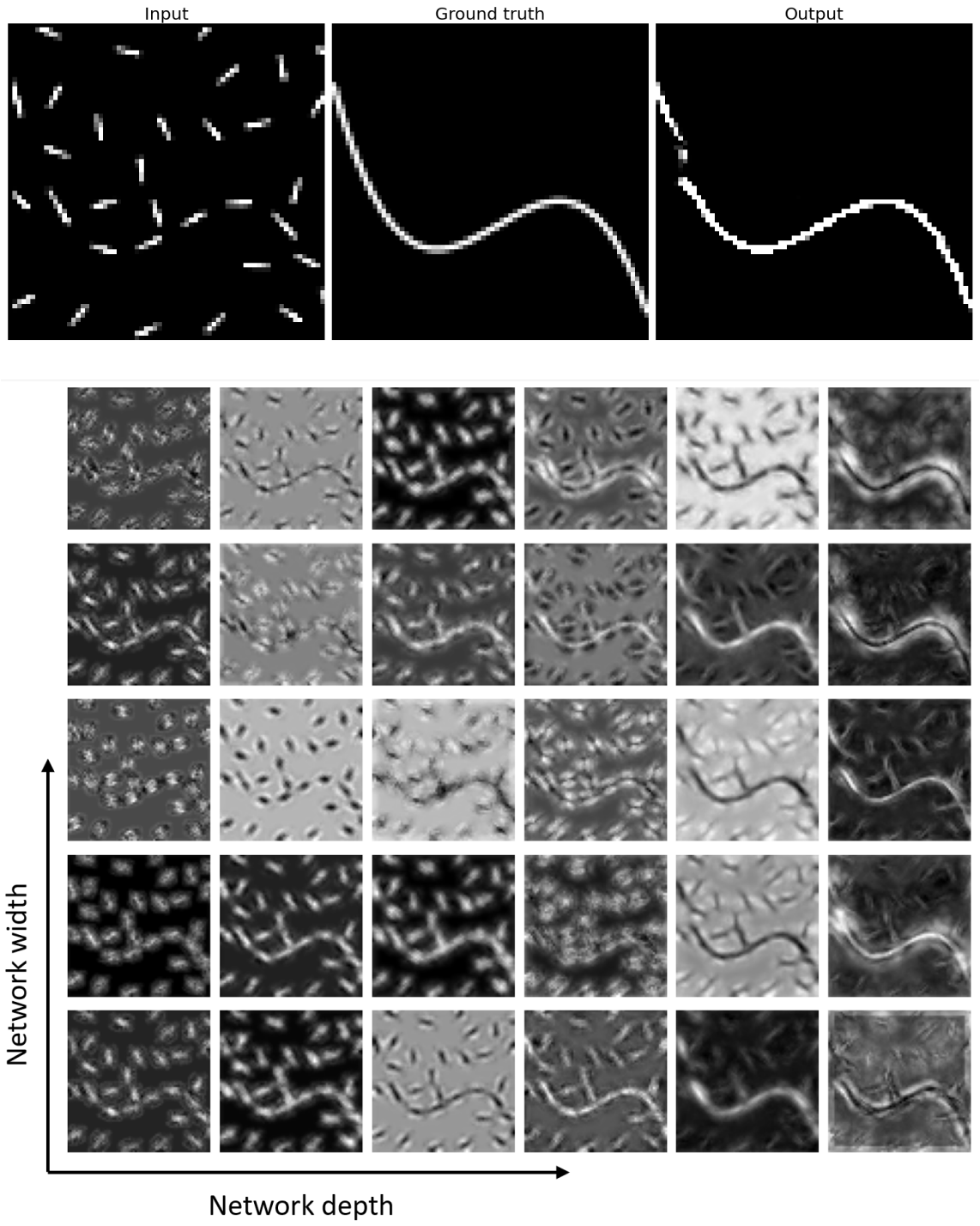


Figure 24: Another example that has some fault in the output. We see that the network could not quite line up the correct line segments, resulting in the disconnect in the left part of the curve.

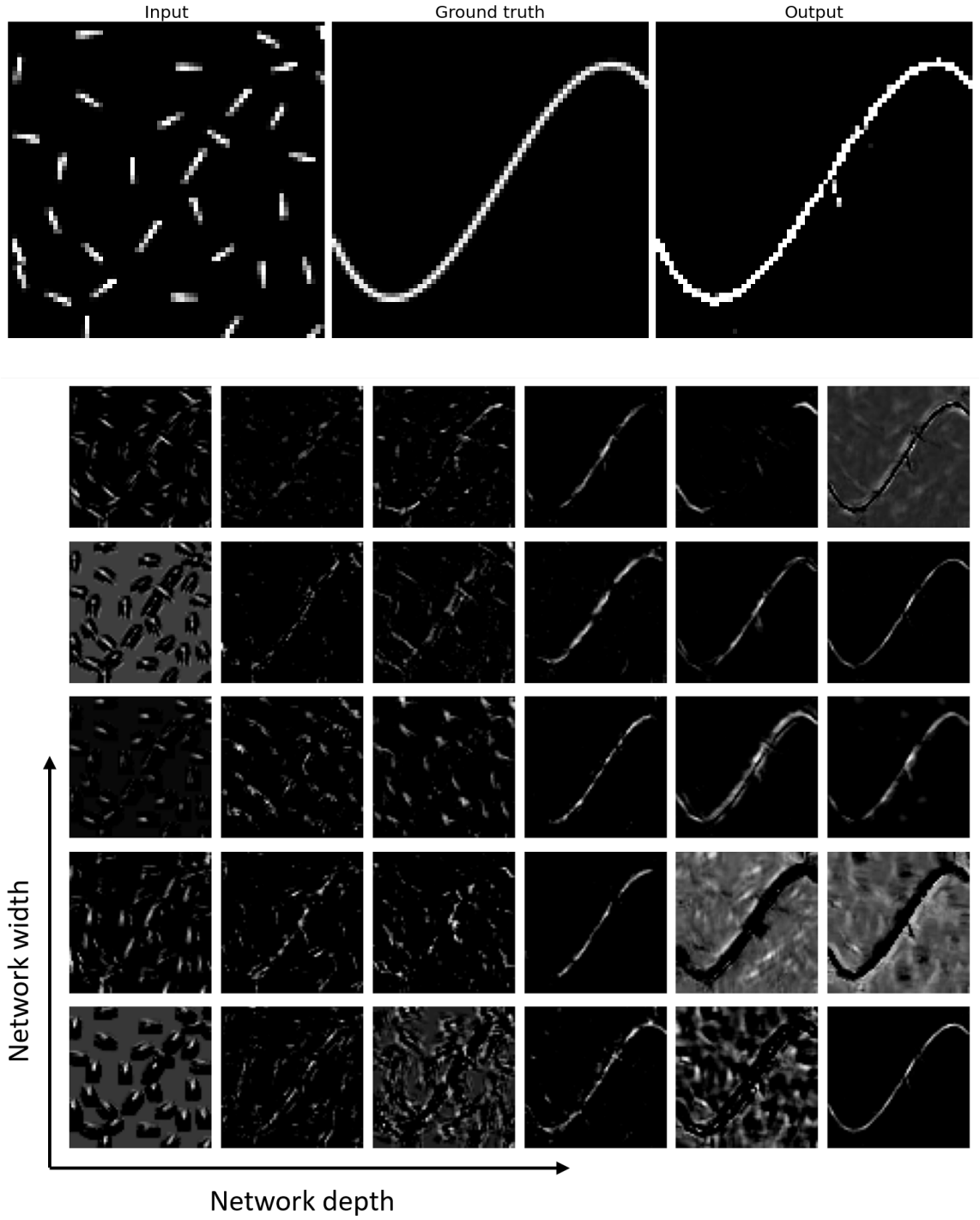


Figure 25: The same plot as above, for the CNN trained on the easy dataset.

5.3 Visualization of Trained PDE-G-CNNs

Seeing how in the previous subsection we found that both anisotropic and isotropic networks perform well, we want to get a better understanding of what is happening. With the hope of achieving this, we will visualize (parts of) the PDE-G-CNNs. We will only talk about the PDE layers in the network here. In Appendix C we briefly talk about the lifting layer.

5.3.1 PDE layer

In order to visualize the PDE layer, we need to visualize one activation function, as in equation (39). Since we solve this PDE using operator splitting, we will first discuss the visualizations for each of the terms separately, before combining them in one visualization.

We can visualize the convection by plotting exponential curves, as the convection is taking place over these curves, as we have seen before. We will show some examples of these curves, both plotted in 3d and projected down to $\mathbb{R}^2$, in Figure 26. Although the effect of convection is not very straightforward, recall Figure 15 from Section 3, these plots do tell us something about the magnitude and direction of the convection happening.

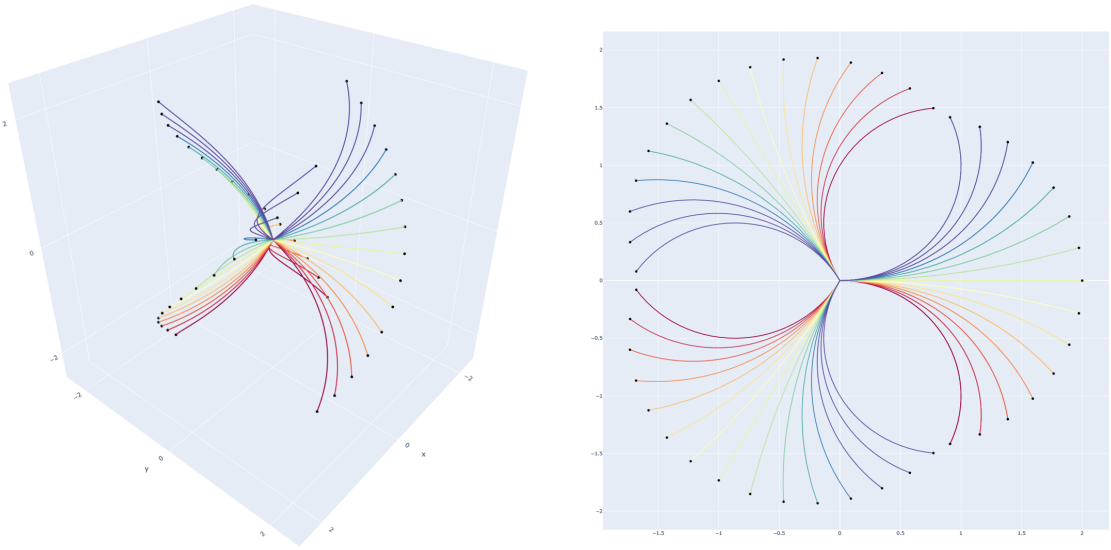


Figure 26: Some exponential curves for various convection vectors $\boldsymbol{c} = c^i A_i$, both in $\mathbb{M}_2$ and projected to $\mathbb{R}^2$.

Next, we turn our attention to visualizing the dilation/erosion part. Since the solution to either of these operations is a morphological convolution with a specific kernel, it makes sense to find a way to visualize this kernel. In our case, this kernel is completely determined by the distance ρ_b, so we opt to visualize this function instead of the kernel. One reasonable way to visualize these would be to plot iso-surfaces of the distance. Figure 27 shows some examples of this. Note that the level sets of the actual kernels we take the infimal convolution with have the same shape as the plotted level sets here.

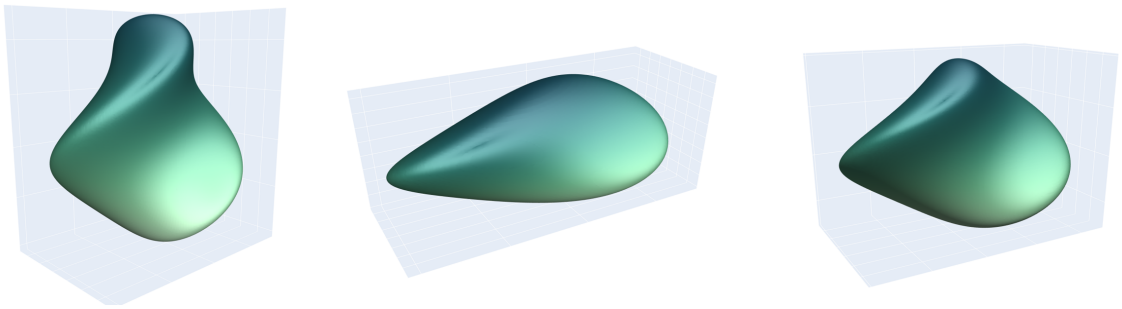


Figure 27: Level sets of ρ_b for various configurations of the metric coefficients.

However, we would like to stay away from 3D visualizations if possible, as they can be confusing and un-intuitive. What we will do instead, is project the level sets down to 2D. The way we do this is by plotting level sets of the function $\rho_{2d}(x, y) = \min_{\theta \in [0,2\pi]} \rho_b(x, y, \theta)$. Figure 28 shows an example, together with the 3D level set.

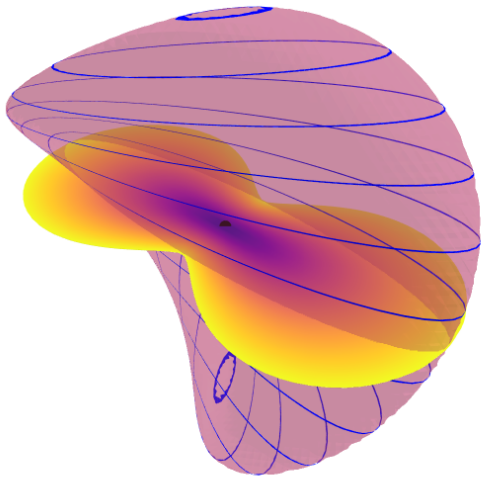


Figure 28: Illustration of how the 3D level set relates to the min-projected level set. The blue lines are iso lines on fixed values of θ.

The last thing we have to do now is to combine the convection and dilation/erosion visualizations into one plot. In the *lietorch* code, first convection and then the morphological operations are applied. Let $f : \mathbb{M}^2 \to \mathbb{R}$ denote our input image. Applying convection is applying $\mathcal{R}_g$ for some $g \in G$. Next, taking the morphological convolution with kernel k of this function results in

$$
\begin{aligned}
(k \,\square\, \mathcal{R}_g f)(p) &= \inf_{h \in G} k(h^{-1}p) + (\mathcal{R}_g f)(hp_0) \\
&= \inf_{h \in G} k(h^{-1}p) + f(hgp_0) \\
&= \inf_{h' \in G} k(g(h')^{-1}p) + f(h'p_0) \\
&= \inf_{h \in G} \hat{k}(h^{-1}p) + f(hp_0) \\
&= (\hat{k} \,\square\, f)(p),
\end{aligned}
$$

where we define $\hat{k} = \mathcal{L}_{g^{-1}} k$, i.e. $\hat{k}(p) = k(gp)$. So we see that applying convection and then a morphological convolution results in a convolution with a translated kernel.

So, what we will do to visualize the convection, dilation and erosion effect of one PDE channel, is to plot the min-projected level sets of the translated distance maps that are trained. Figure 29 below shows some examples of the visualization, with a (non-zero) convection, dilation and erosion. Since dilation dilates the orientation score, i.e. increases responses in certain directions, we color it green. For a similar reason we color the erosion distance map red. Lastly, we need to address the arrows drawn inside of the level set. These represent the association fields that we talked about in the introduction, determined by the specific metric of the distance. We have chosen to represent these by horizontal curves, and not geodesics, for the sole reason that they are much easier to compute, and are just there to convey the idea that these plots are meant to represent the fact that we are training association fields.

The particular choices of metrics that were chosen for the example in Figure 29 were deliberate. We see that the dilation and erosion happens in orthogonal directions, i.e. we "excite" in one direction and "inhibit" in the other. We would expect to find such behaviour in our network as well when training for the line data set. It would make sense that we dilate in one direction, focusing the local orientation of the image, and sharpen in the orthogonal direction, to prevent misalignment of the local orientations.

With all of these things now discussed, we can turn to visualizing the complete PDE part of a PDE-G-CNN. The way we do this is straightforward. We go through the layers one by one, and for each channel in a layer we make a visualization like in Figure 29, and plot these in one column. We have opted to not plot the linear part that is between each layer, as this might be un-intuitive, and distract from the activation function plots.

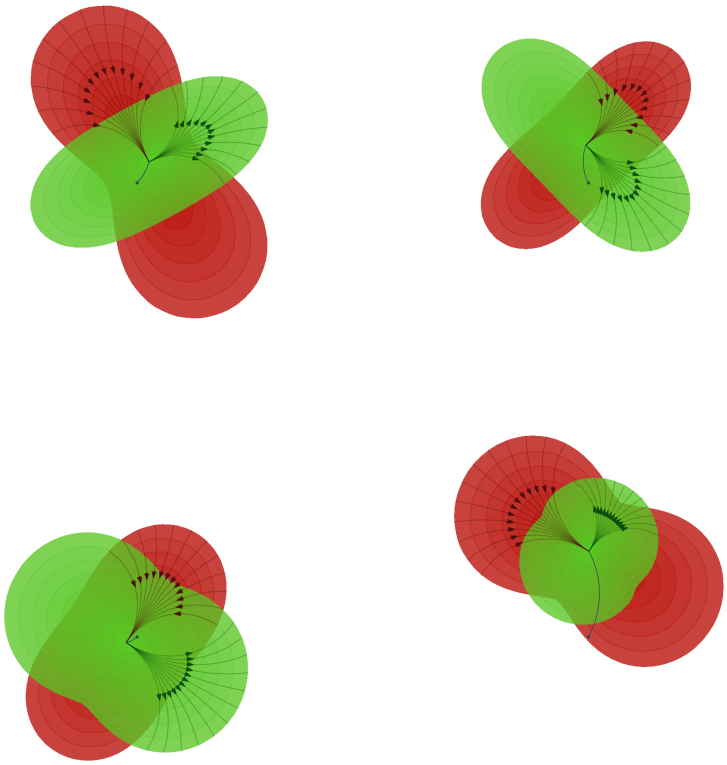


Figure 29: Figure showing the way we visualize the activation function. The blue curve is a projected exponential curve, the green surface is the min-projected level set of the dilation distance, and the red surface of the erosion distance. The association fields we draw here are determined by these distances, and are horizontal exponential curves ending at the boundary of the level sets.

In Figure 30 we can see such a visualization for the three different PDE-G-CNNs we trained in the previous subsection. Note that each level set is drawn for the same value, so we get the effect that some of the level set are much larger. This is also the reason for the cut-off for some of the smaller pictures; the level set is very large in this case. If we were to have drawn these fully, the others would not have been visible.

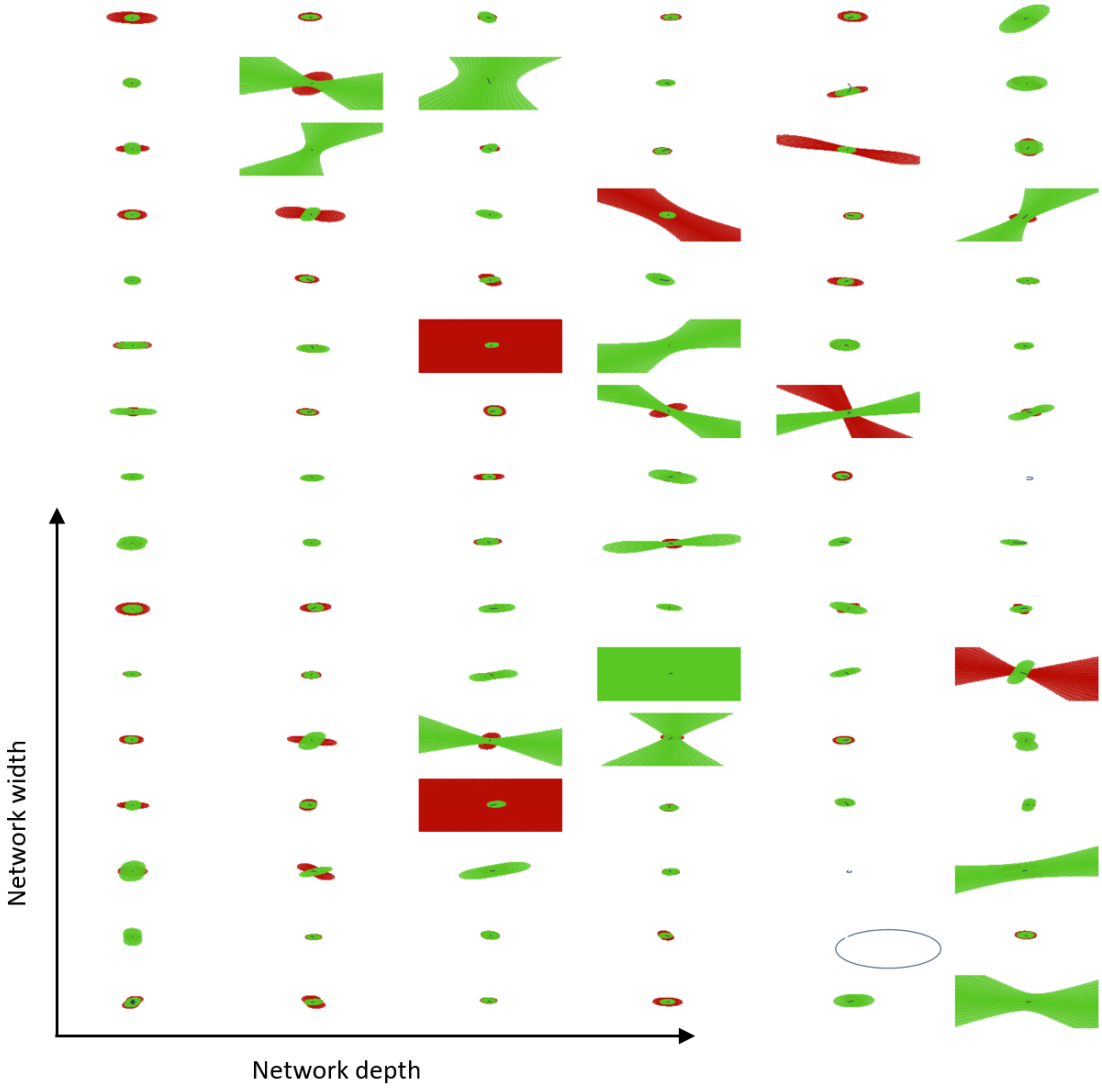
There are some things we want to discuss. First of all, we can clearly see some cases of very high anisotropy in the first two pictures, which we of course already noticed earlier when checking for the validity of the approximation we use. This anisotropy presents itself as the level sets being very large in one direction and small in another.

On the other hand, we see quite a lot of very small level sets in the picture. These correspond to relatively large values for the metric parameters. In practice, this means that the kernel with which the numerical morphological convolution is taken increases quickly outwards from the center, so that these channels do not change their input very much (at least not through the dilation and erosion). If they also have small convection vectors, these channels can be seen as a sort of "skip connection", merely passing the input through to the next layer.

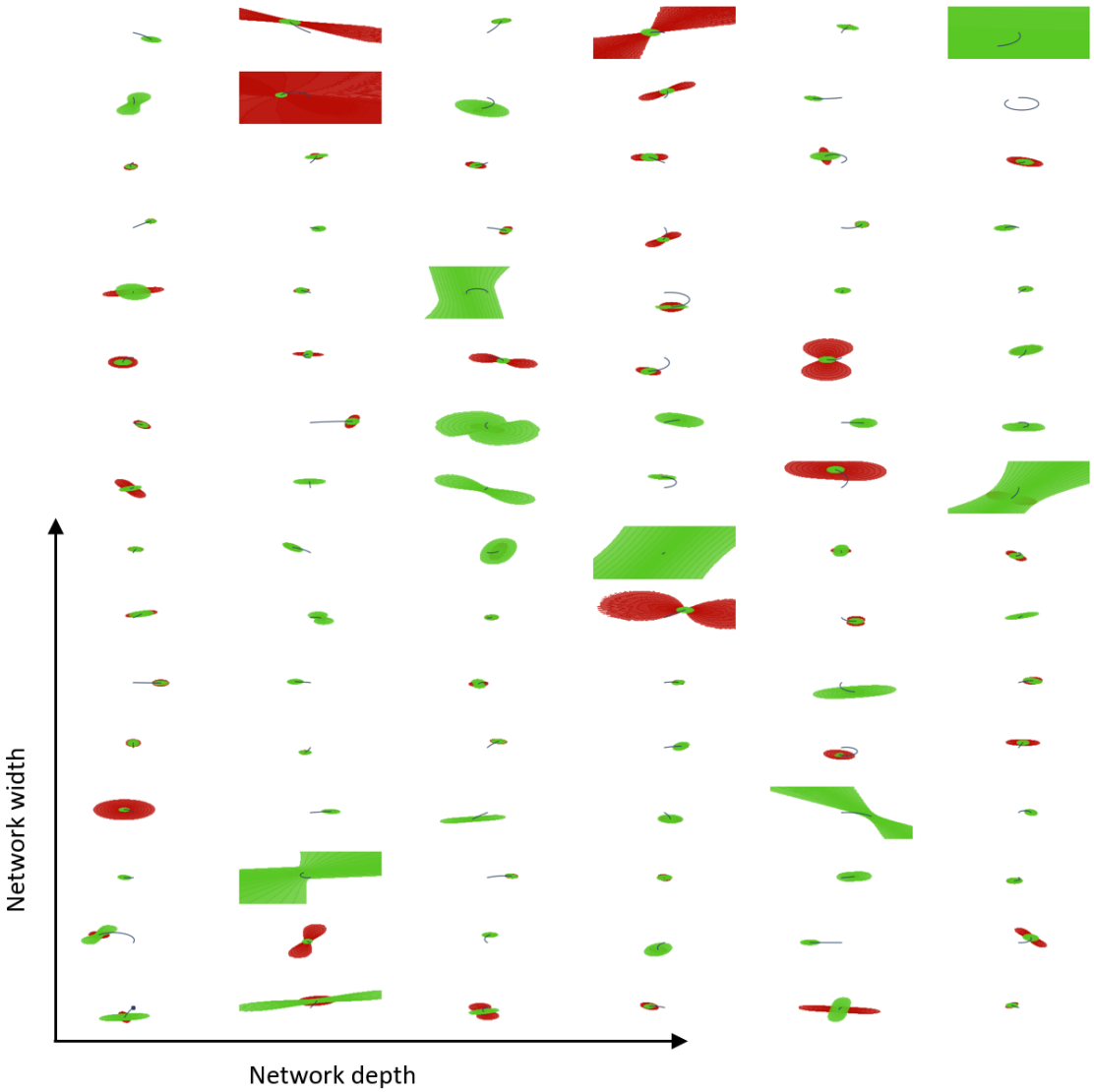
Lastly, we can see in this first two sub-figures of Figure 29 that we indeed have that several of the larger level sets, those that would actually perform a meaningful amount of dilation and erosion,

have kernels that are orthogonal to each other, or one is much smaller then the other. It does not seem like the network needs to have this to function, which of course the performance of the anisotropic network already disproves, but if it is available it does seem to make use of these configurations somewhat.

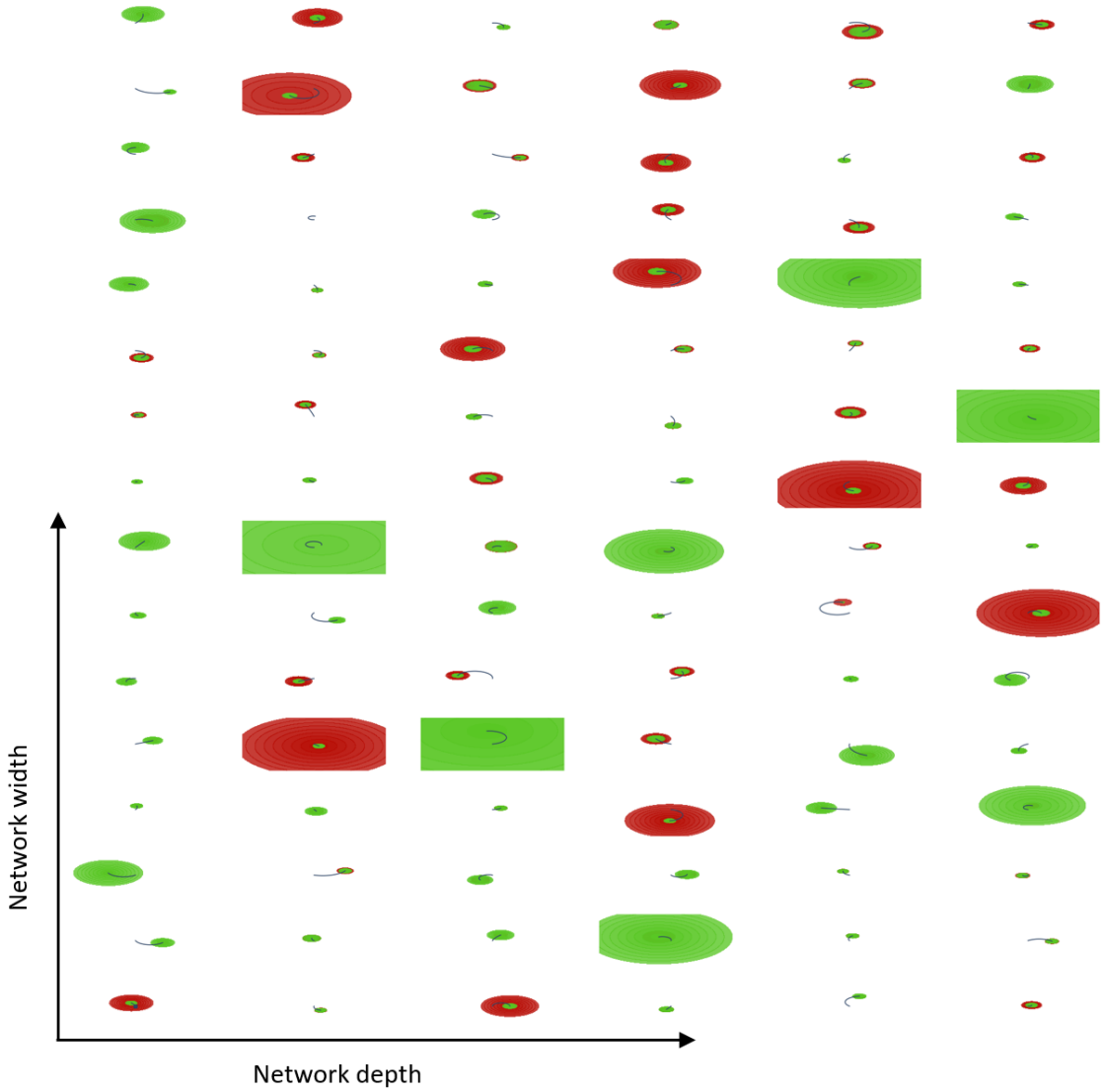
If we draw these figures for each epoch we train we can get an idea of the evolution of the network. What is visible when doing this, is that not a lot of the metric parameters in the network change. Many of the channels with small level sets remain more or less the same over training, suggesting that they indeed function as some sort of skip connection as we discussed above. The channels with large level sets in the final network appear to mostly have started already with large level sets. This could be an incentive to look at other, maybe better, initialization methods for the metric parameters.



(a) Visualized PDE layers of a PDE-G-CNN trained on the "easy" Line data set. This network used the old approximation ρ_c and applied the ReLU function to the trained metric coefficients. Note that for the larger level sets we often either have orthogonality between the two kernels, or one is much smaller then the other.



(b) Visualized PDE layers of a PDE-G-CNN trained on the "easy" Line data set. This network was trained with squared metric coefficients and the ρ_b approximation. The smaller level sets seem smaller then in (a). We also again see some example of orthogonality between the kernels in some places.



(c) Visualized PDE layers of a PDE-G-CNN trained on the "easy" Line data set. This network was restricted to only working with isotropic metrics, as we can see from the exclusively elliptic shape of the level sets. It again used squared metric coefficients and the ρ_b approximation.

Figure 30: Various full network visualizations. For clarity we omitted the association fields and linear combinations between the layers. Other than that, the individual pictures are made in the same way as in Figure 29.

5.4 Further Experiments

The last thing we want to investigate is whether a more suited approximation for highly anisotropic metrics is needed. As we already established, it occurs relatively often that the network gets to a situation where the ρ_b approximation becomes rough, so we wonder if a better approximation will increase the performance. We will consider two different approximations. The first one comes from [62], and is given by

$$\rho_{b,sr} := \begin{cases} \sqrt{\sqrt{\alpha g_{11} g_{33}}|b^2| + g_{11}|b^1|^2 + g_{33}|b^3|^2} & \text{if } g_{11} \leq g_{22} \\ \sqrt{\sqrt{\alpha g_{22} g_{33}}|b^1| + g_{22}|b^2|^2 + g_{33}|b^3|^2} & \text{else.} \end{cases} \tag{83}$$

Here, $\alpha = 44$ is chosen [63]. This is specifically an approximation for the sub-Riemannian distance we would get if either g_{11} or g_{22} would approach infinity. The second approximation combines the global bound we found in Section 4 with ρ_b and the above approximation, and is given by

$$\rho_{b,com} := \max(l, \min(\rho_{b,sr}\ , \rho_b)), \tag{84}$$

where l is the lower bound from Lemma 4.4.

All of these experiments will be done on the "medium" difficulty lines data set. This is because we want to get a clear picture of what has influence on the performance and what does not, and since on the easy data set the basic architectures already solved the task well, we thought this would present a more accurate picture. Recall that the difference between the medium and easy data set can be seen in Figures 19, 20.

For comparison's sake, we trained a base model (base now refers to a model with the b-coordinates and squared metric components) and an isotropic model on this data set as well. We visualize the output for the base model in Figures 32 and 33 (in the same way as for the easy data set), and the performance results can be found in Figure 31. As is to be expected, the performance measures are all significantly lower than with the easy data set, however as the output visualization shows, visually the results still look good.

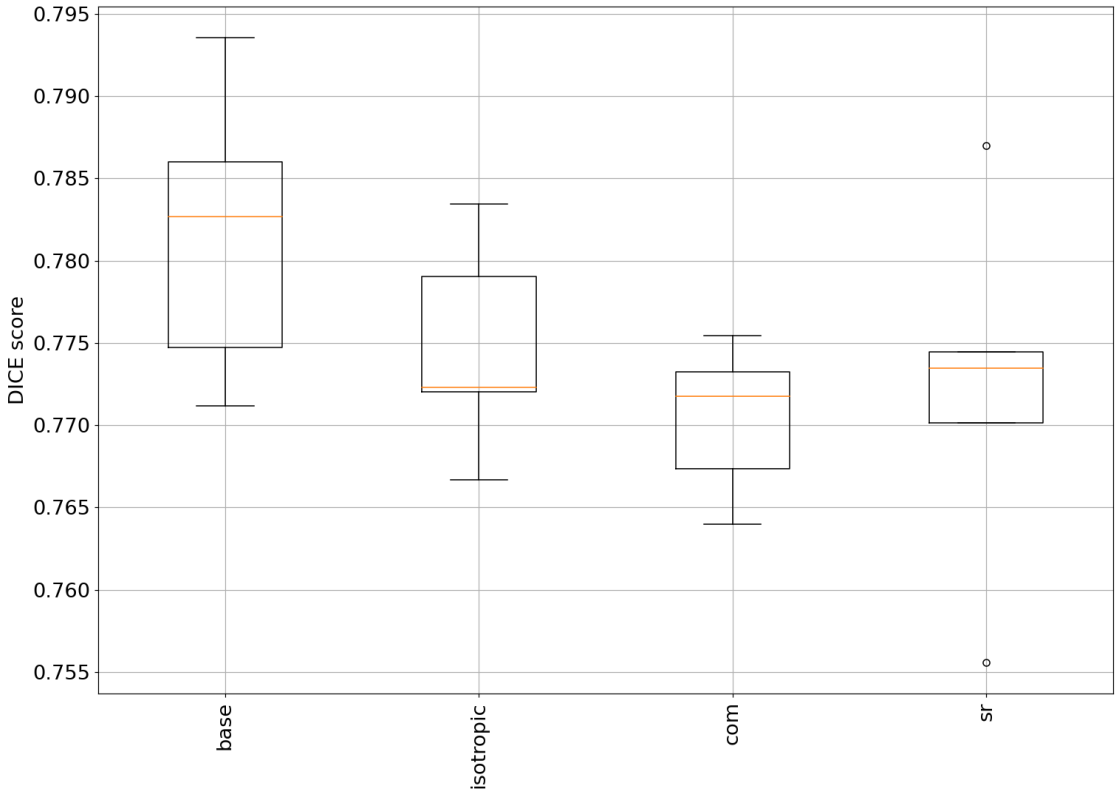


Figure 31: The performance metric for the "medium" Line data set. Again, for each network architecture 5 models were trained with the same configuration of hyper parameters, as described in Appendix B.

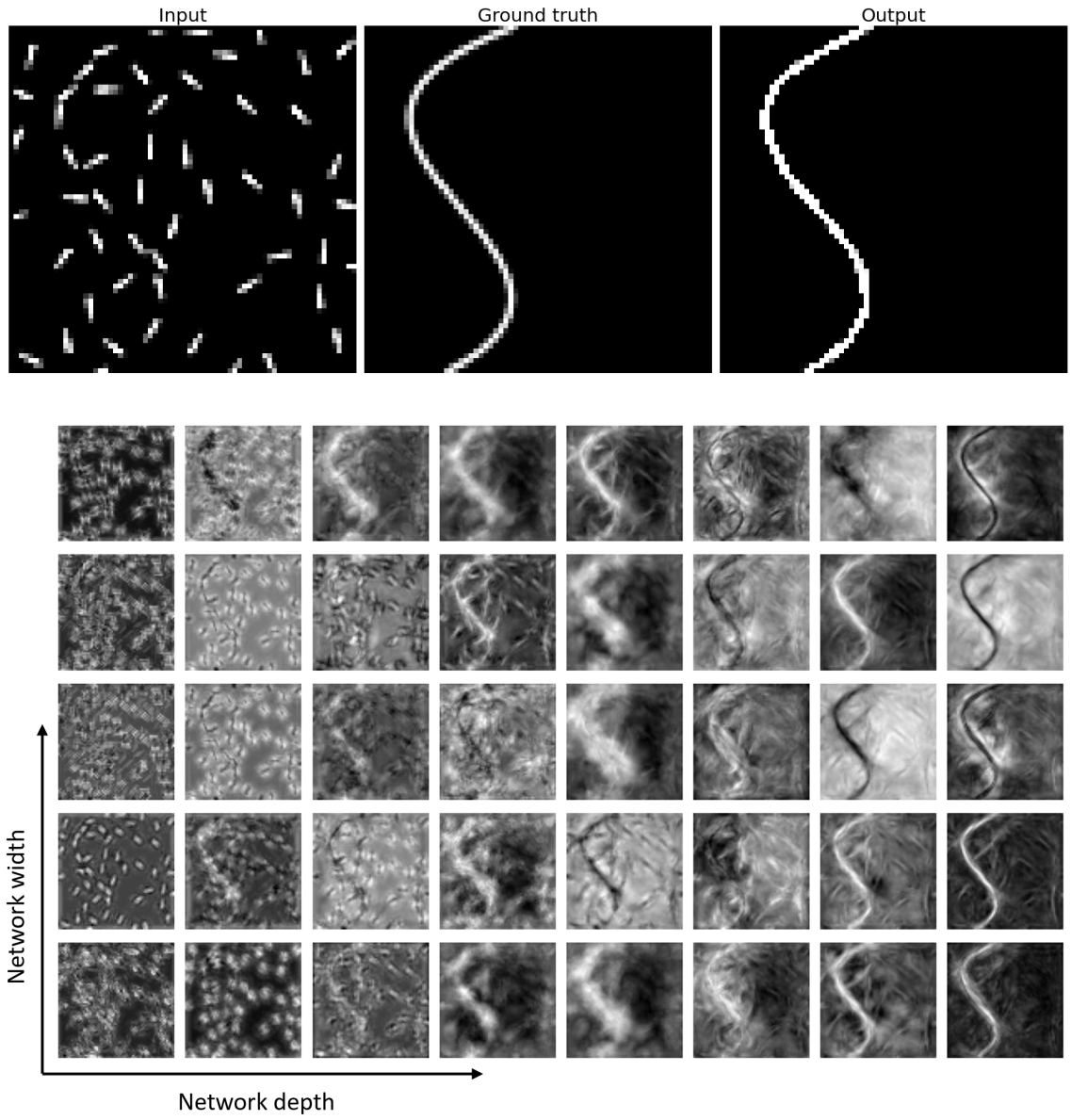


Figure 32: The input, ground truth and output of the network, for a PDE-G-CNN trained on the medium data set.

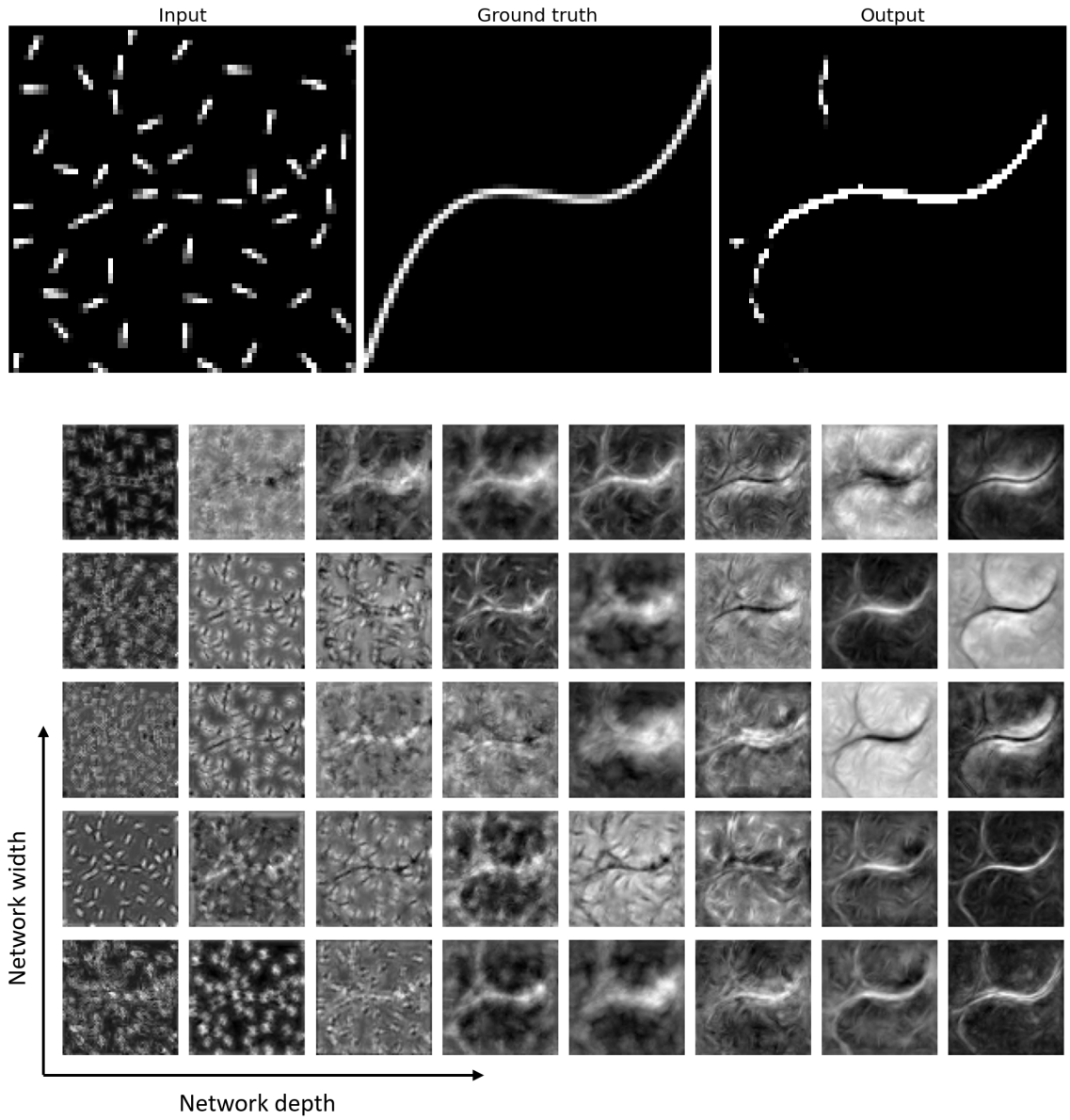


Figure 33: The input, ground truth and output of the network, for a PDE-G-CNN trained on the medium data set. We see that the output is not correct, however this test case was difficult, as the randomly placed lines happened to line up in the bottom left corner.

Let us discuss the results. We can see the performance of the $\rho_{b,sr}$ approximation in the third column in Figure 31, and the performance of $\rho_{b,com}$ in the fourth column. Clearly, the performance of both is slightly lower than our base model, the first column. So it seems like the network does not benefit from more accurate kernels in anisotropic cases. Perhaps this is due to the fact that because we already are sampling on course grids in order to make the computation time reasonable, the additional accuracy that we have in theory does not matter in practice. Moreover, the more complicated formulas might make gradient descent harder.

It could of course also be the case that the regions we found using 4.13 are too restrictive, and that in reality ρ_b is accurate on a larger region. We would like to have some threshold to tell us when we should switch between the different approximations, but we leave this to further research.

6 Conclusion

In this report we have analyzed and visualized PDE-G-CNNs designed to work on image data lifted to $\mathbb{M}_2$. In particular, we derived two new bounds relating the true Riemannian distance $d_{\mathcal{G}}$ with the half-angle coordinate approximation ρ_b. We also related this error to the kernels which are used to solve dilation and erosion evolution equations, which form part of the activation function in PDE-G-CNNs.

Next, we trained PDE-G-CNNs on a new data set, called the "Line" data set, and analyzed the performance. We looked at how well the dilation and erosion evolution equations are solved in a concrete case, using the newly found bounds. We presented a way to visualize the network in terms of its geometric parameters, and made several observations from these visualizations. Lastly, we looked at the (projected) feature maps of the PDE-G-CNNs in an attempt to better understand what the network is doing to reach its output.

In the introduction we posed several questions, which we answered throughout the report. The first thing was whether we could improve the assessment of the quality of the approximation used in the PDE-G-CNNs. This was done with Theorem 4.8 and 4.12, that confirmed the suspicion we had that the quality of the performance depends on ζ and g_{33}. Using this we could calculate regions in which we could still guarantee a certain degree of accuracy; this was done in Remark 4.13. Lastly, we checked in Section 5.2 whether the trained PDE-G-CNNs are actually sampling in these regions, which turned out to not be the case in many channels. This improved on previous work in [24], where a more general but less sharp bound was found.

Next, we wanted to visualize the PDE-G-CNNs in a meaningful way. This was done in Section 5.3, where we discussed how we could visualize the parameters of an individual PDE activation function, which we used to visualize the entire PDE-layer part of the network. From this we noticed that many channels appear "stuck" in a configuration in which the metric parameters are relatively large, implying that the effect of these channels is not very large. We also saw that certain configurations that we would expect there to be from a modeling point of view were present, in particular the fact that erosion and dilation happen in orthogonal directions.

Lastly, we wondered how geometrically interpretable the functioning of PDE-G-CNNs is. To answer this we first looked at the performance of PDE-G-CNNs, and compared this with regular CNNs and PDE-G-CNNs using other approximations. This was done throughout Section 5. We also talked about the concept of "geons" in the introduction, and wondered if we would see the formation of these in the feature maps of the PDE-G-CNNs. We plotted these feature maps in Figures 24,23, 32, 33, and compared this with feature maps of CNNs in Figure 25. The conclusion was that we do indeed see that gradually over the depth of the network we have that a global line filter seems to emerge from the combination of association fields. In the CNNs' output we do not see such gradual formation of the final output.

Future Directions

The last thing we want to discuss is some future research directions related to this report. For one, it could be interesting to see if the ideas used here to find the bounds between the distances can be generalized to other homogeneous spaces. Both approaches could be applied if we have an explicit expression for the approximate distance.

The visualizations of the entire network could be improved. Now, due to the large difference in size of level sets, several plots are cut off. Maybe there is a different way to visualize the difference between the sizes. One could for example plot the kernels that are actually sampled in the PDE-G-CNNs, so that this difference becomes immediate. Also, we concluded that a lot of channels appear to not apply a lot of dilation/erosion. It would be interesting to investigate how important these channels actually are, and whether the performance could be improved by for example different initialization methods.

More fundamentally, we wonder if we could mathematically quantify the creation of geons. For example, can we say something about the dynamics of when individual lines become connected via dilation/erosion? Could we set up axioms that give rise to the choices of the PDEs that are used in PDE-G-CNNs, similar to [64]? These questions will be answered in future work.

A Details of Computations

This appendix contains a variety of computations which are rather tedious, and would distract from the point if we were to put them in the main text. They are presented here for completeness of the arguments presented.

In particular, we show here how the Taylor expansion was calculated using Mathematica. We will not provide the full steps here, as that would take up too much space, but instead we provide some detail as to how this was calculated in Mathematica and provide some intermediate results.

We first want to compute the metric coefficients of $\mathcal{G}$ in the b-coordinate system. In order to do this we start by expressing it in the fixed coordinate system. Recall that $\mathcal{G}$ is a diagonal metric with respect to ω^i. In light of equation (27) we can express $\mathcal{G}$ in the fixed coordinate system by calculating

$$\begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}^T \begin{pmatrix} g_{11} & 0 & 0 \\ 0 & g_{22} & 0 \\ 0 & 0 & g_{33} \end{pmatrix} \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{85}$$

which equals

$$\begin{pmatrix} g_{22}\sin^2(\theta) + g_{11}\cos^2(\theta) & (g_{11} - g_{22})\sin(\theta)\cos(\theta) & 0 \\ (g_{11} - g_{22})\sin(\theta)\cos(\theta) & g_{11}\sin^2(\theta) + g_{22}\cos^2(\theta) & 0 \\ 0 & 0 & g_{33} \end{pmatrix}. \tag{86}$$

From this form we also immediately see that if $g_{11} = g_{22}$, we have a diagonal metric in the fixed coordinate system. This implies that if $g_{11} = g_{22}$, the exact distance is given by

$$d_{\mathcal{G}}(p) = \sqrt{g_{11}x^2 + g_{11}y^2 + g_{33}\theta^2}. \tag{87}$$

However, recall equation (34). This tells us that

$$d_{\mathcal{G}}(p) = \sqrt{g_{11}x^2 + g_{11}y^2 + g_{33}\theta^2} = \sqrt{g_{11}b^1(p)^2 + g_{11}b^2(p)^2 + g_{33}\theta^2} = \rho_b(p). \tag{88}$$

The next step now is to change the metric from the fixed coordinate system to the b-coordinate system. We already know how to transform between the coordinates from equation (33). So, transforming between the coordinate vector fields of these coordinates is done via the Jacobian, given below

$$(\frac{\partial b^i}{\partial x^j})(x, y, \theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & \sin\left(\frac{\theta}{2}\right) & \frac{1}{2}y\cos\left(\frac{\theta}{2}\right) - \frac{1}{2}x\sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) & -\frac{1}{2}x\cos\left(\frac{\theta}{2}\right) - \frac{1}{2}y\sin\left(\frac{\theta}{2}\right) \\ 0 & 0 & 1 \end{pmatrix}. \tag{89}$$

We then just apply the same idea as in equation (85) to transform the metric coefficients from the fixed to the b coordinate system. Unfortunately the expressions for the metric coefficients become very long and cumbersome, and do not fit on the page. See the Mathematica notebook for the exact expression.

Having calculated this, we can now calculate the Taylor expansion for these metric coefficients. We let $g_{ij}(p)$ be the metric coefficients in the b-coordinate system, and let $g_{ij,k} = \partial_{b^k} g_{ij}, g_{ij,kl} = \partial_{b^l}\partial_{b^k} g_{ij}$. We want to know these in the point $p_0 = e = (0, 0, 0)$. Using Mathematica to find these derivatives, we get that

$$(g_{ij,k}(0,0,0))_{i,j,k} = \begin{pmatrix} (0,0,0) & \left(0,0,\frac{1}{2}\left(g_{11}-g_{22}\right)\right) & \left(0,-\frac{g_{11}}{2},0\right) \\ \left(0,0,\frac{1}{2}\left(g_{11}-g_{22}\right)\right) & (0,0,0) & \left(\frac{g_{22}}{2},0,0\right) \\ \left(0,-\frac{g_{11}}{2},0\right) & \left(\frac{g_{22}}{2},0,0\right) & (0,0,0) \end{pmatrix} \tag{90}$$

and

$$(g_{ij,kl}(0,0,0))_{i,j,k,l} =$$

$$\begin{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}\left(g_{22}-g_{11}\right) \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & \frac{1}{4}\left(g_{11}-g_{22}\right) \\ 0 & 0 & 0 \\ \frac{1}{4}\left(g_{11}-g_{22}\right) & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}\left(g_{11}-g_{22}\right) \end{pmatrix} & \begin{pmatrix} 0 & 0 & \frac{1}{4}\left(g_{11}-g_{22}\right) \\ 0 & 0 & 0 \\ \frac{1}{4}\left(g_{11}-g_{22}\right) & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4}\left(g_{22}-g_{11}\right) \\ 0 & \frac{1}{4}\left(g_{22}-g_{11}\right) & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & \frac{1}{4}\left(g_{11}-g_{22}\right) \\ 0 & 0 & 0 \\ \frac{1}{4}\left(g_{22}-g_{11}\right) & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4}\left(g_{22}-g_{11}\right) \\ 0 & \frac{1}{4}\left(g_{22}-g_{11}\right) & 0 \end{pmatrix} \\ & & \begin{pmatrix} \frac{g_{22}}{2} & 0 & 0 \\ 0 & \frac{g_{11}}{2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{pmatrix} \tag{91}$$

In the proof of lemma 4.8 we end up contracting these arrays over coordinates, i.e. we compute there something of the form $g_{ij,k}(0,0,0)b^ib^jb^k$, with b^i the coordinate functions. We calculate these quantities next. Contracting over the array in equation (90) gives us

$$g_{ij,k}(p_0)b^ib^jb^k = 0. \tag{92}$$

We contract over the absolute values of the array in equation (91), which results in

$$|g_{ij,kl}(p_0)|b^ib^jb^kab^l = \frac{1}{2}b^3(p)^2\left(g_{11}b^2(p)^2 + 3|g_{11}-g_{22}|\left(b^2(p)^2+b^1(p)^2\right)+g_{22}b^1(p)^2\right) \tag{93}$$

We can write this last expression as a quadratic form in the squares of b^i, as follows

$$g_{ij,kl}(0,0,0)b^ib^jb^kb^l = \frac{1}{4}\begin{pmatrix}(b^1)^2\\(b^2)^2\\(b^3)^2\end{pmatrix}^T M \begin{pmatrix}(b^1)^2\\(b^2)^2\\(b^3)^2\end{pmatrix}, \tag{94}$$

with

$$M = \begin{pmatrix} 0 & 0 & g_{11}+3|g_{11}-g_{22}| \\ 0 & 0 & g_{22}+3|g_{11}-g_{22}| \\ g_{11}+3|g_{11}-g_{22}| & g_{22}+3|g_{11}-g_{22}| & 0 \end{pmatrix}.$$

This matrix somewhat resembles the curvature matrix when computing it using the Torsion free Cartan connection, recall equation (15), using the left-invariant frame $\mathcal{A}_i$ on $SE(2)$. This curvature matrix would not have the additional $3|g_{11}-g_{22}|$ terms. We are unsure if there is anything deeper going on here or whether this is coincidence.

B Experiment Details

Here the details of the experiments that were done in Section 5 are collected. For all the results in Figures 21 and 31 we trained the architectures for 5 different runs. All of them were trained on the *continuous dice loss*. The ground-truth images are in the range $[0, 1]$, and our network segmentation is passed through a sigmoid to produce a number in the range $[0, 1]$. To calculate the continuous dice loss we then compute

$$\text{loss}(a, b) = 1 - \frac{2\sum_{i,j} a_{i,j} b_{i,j} + \varepsilon}{\sum_{i,j} a_{i,j} + \sum_{i,j} b_{i,j} + \varepsilon}, \tag{95}$$

where a and b are the (2D) output segmentation map of the network and the ground truth respectively, and ε is there to prevent numerical errors with division. All of the networks had a batch size of 30, and were trained with a learning rate of 0.01 with an exponential decay factor of 0.95 over the epochs. The optimizer used was the ADAM optimizer. Lastly, we put a normalized L^2 loss on the total sum of the parameters, with a weight factor of 0.001.

B.1 Easy Data Set

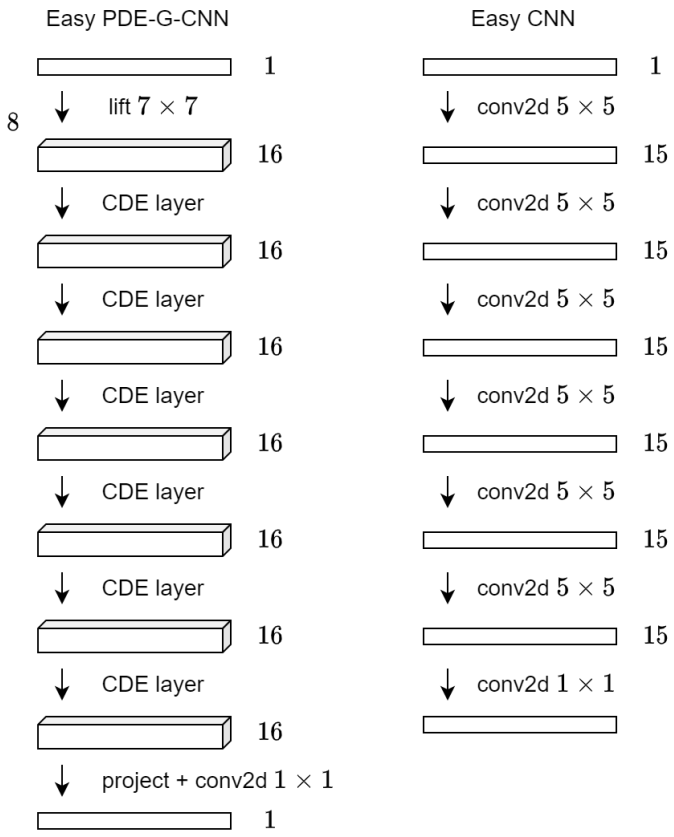


Figure 34: Architectures used for the networks trained on the easy Line data set. Numbers on the right indicate the number of channels, boxes with depth indicate that the data lives on $\mathbb{M}_2$, and the number to the left of the 3D boxes indicates the number of orientations used. Based heavily on the architecture used in [24]. The PDE-G-CNN has 7 PDE layers, with a total of 3424 trainable parameters, and the CNN has 6 convolutional layers, with a total of 10600 trainable parameters.

B.2 Medium Data Set

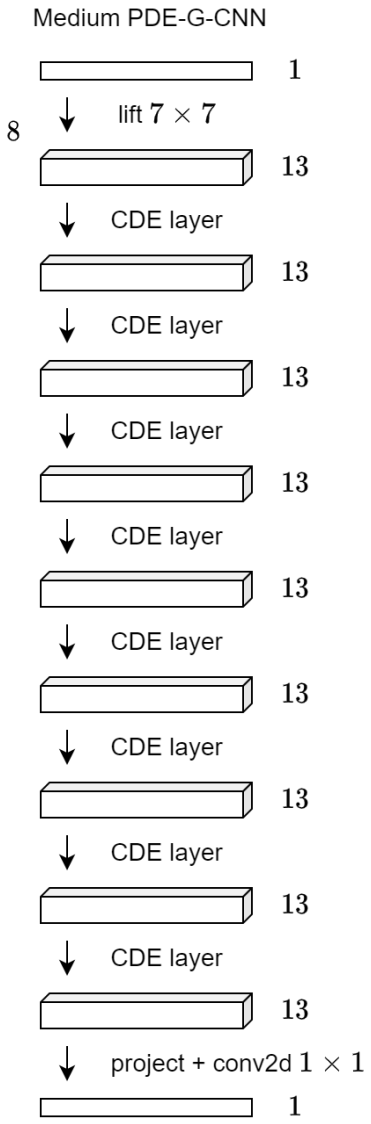


Figure 35: Architecture used for the networks trained on the medium Line data set. Numbers on the right indicate channels, boxes with depth indicate that the data lives on $\mathbb{M}_2$, and the number to the left of the 3D boxes indicates the number of orientations used. Based heavily on the architecture used in [24]. This network has 9 PDE layers, with a total of 3172 trainable parameters.

C Lifting Layer

We already explained the basics of the lifting layer in Section 3.2. Experiments show that the lifting layer is in fact very important for the performance of the network, with the current design of them. Reducing the amount of channels in the lifting layer damages the performance quite a bit. This seems logical, since the entire kernel is trainable, and we use 8 channels of 7×7 kernels, there are quite a lot of parameters here.

We start of by showing some examples of the learned kernels of trained networks, specifically the "base" models of Section 5.2. These can be seen in Figure 36. As we can see, there appears to be some structure in them, but nothing that immediately jumps out.

Figure 36: The lifting layer kernels visualized for three networks. These are used to lift the input image to $\mathbb{M}_2$ in the trainable way described in Section 3.2.

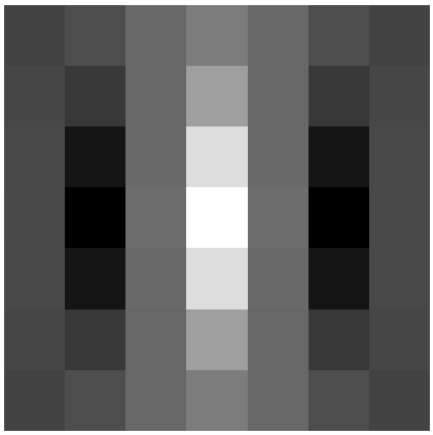


Figure 37: Lifting kernel based on cake wavelets [25], for comparison to the trained lifting kernels. For all the lifting of images with a fixed kernel, this is the one that was used.

Could we, instead of training the kernels, choose a kernel we know is a good choice and just use this instead? Let us investigate this theoretically a little bit. Suppose we fix our lifting kernel $k : \mathbb{R}^2 \to \mathbb{R}$, and consider some image $f : \mathbb{R}^2 \to \mathbb{R}$. What happens if we were to just apply convection and a linear layer to the lifted image $(\mathcal{W}_k f)$? Let us denote the effect of convection by

$\mathcal{R}_{g_i}$, for some collecting of $g_i \in SE(2)$, and let $(\lambda_i)_{i=1}^n$ be weights in $\mathbb{R}$. Applying this to the lifted image gives us

$$\begin{aligned}
\sum_{i=1}^{n} \lambda_i \mathcal{R}_{g_i}(\mathcal{W}_k f)(p) &= \sum_{i=1}^{n} \lambda_i (\mathcal{W}_k f)(pg_i) \\
&= \sum_{i=1}^{n} \lambda_i \int_{\mathbb{R}^2} \mathcal{L}_{pg_i} k(x) f(x) dx \\
&= \sum_{i=1}^{n} \lambda_i \int_{\mathbb{R}^2} \mathcal{L}_p \circ \mathcal{L}_{g_i} k(x) f(x) dx \\
&= \int_{\mathbb{R}^2} \sum_{i=1}^{n} \lambda_i \mathcal{L}_p \circ \mathcal{L}_{g_i} k(x) f(x) dx \\
&= (\mathcal{W}_{\sum_{i=1}^n \lambda_i \mathcal{L}_{g_i} k} f)(p),
\end{aligned}$$

so that applying linear combinations and convections to our lifted image is the same as moving, rotating and taking linear combinations of the kernel that was used to compute the lifted image. So, if we were to lift our image and pass it through a layer with just convection and linear combinations, we might expect that with this fixed kernel we can mimic the effect of training our kernels, see Figure 39. From an interpretability standpoint this would be preferable, as then the lifting is uniform over all use cases, and all learning is solely done on the PDEs in $\mathbb{M}_2$.

Initial experiments showed inconclusive results. On the Line data set, it seems like making this change hurts the performance, see Figure 38 column "fixed-lift". However, on the Rotnist and Drive data set, used in [24], the performance remained the same. We also tried a very simple way of lifting images to $\mathbb{M}_2$, by just stacking the same image on top of each other, the performance of which is visible in Figure 38 column "trivial-lift". More experiments are needed to figure this out.

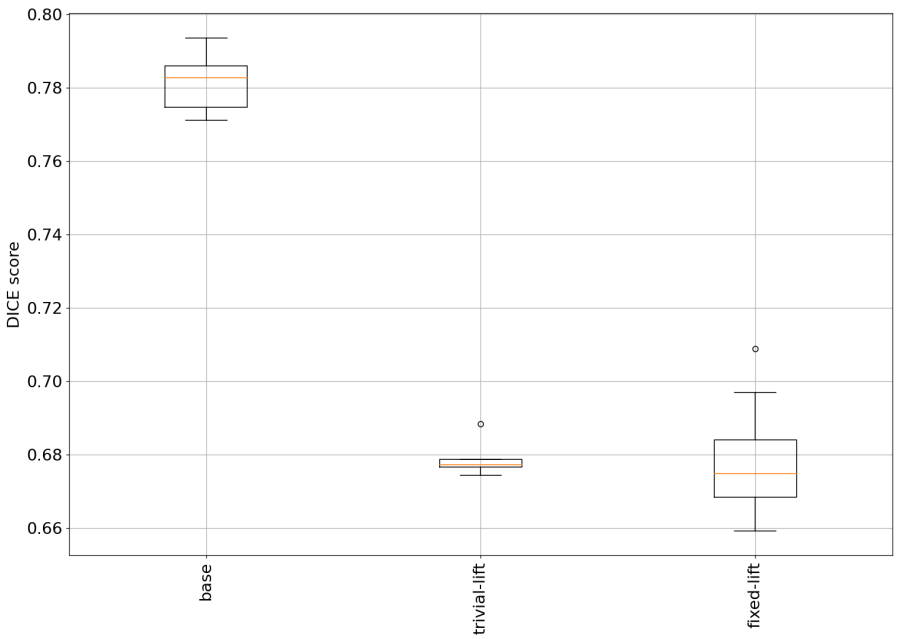


Figure 38: The performance metrics for the different lifting methods, trained on the "medium" Line data set. Hyper parameters are the same ones as reported in Appendix B.2. The base model is the same as the base model from Section 5.4.

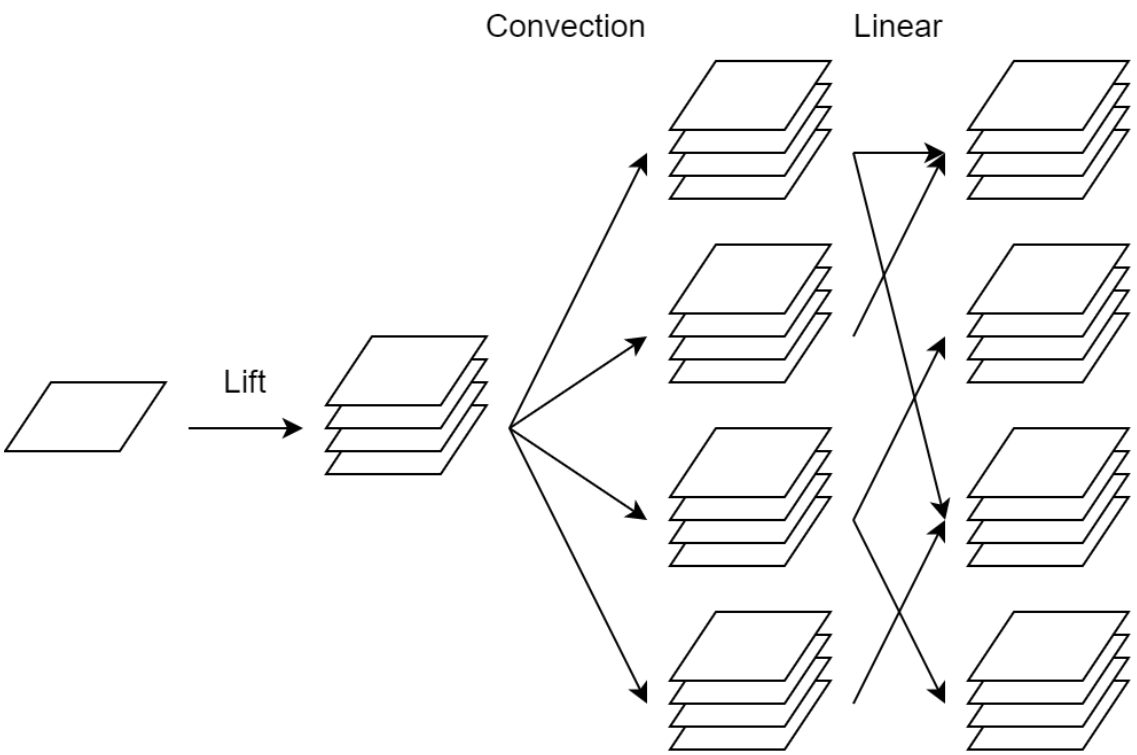


Figure 39: Illustrating the proposed construction to replace the trained kernels in the lifting layer. This architecture would be equivalent to rotating and translating the initial lifting kernel and taking linear combinations. The remainder of the architecture would remain the same.

References

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[2] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR.

[3] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 440–448. Springer, 2018.

[4] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. *arXiv preprint arXiv:1602.02660*, 2016.

[5] Sander Dieleman, Kyle W Willett, and Joni Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450(2):1441–1459, 2015.

[6] Marysia Winkels and Taco S Cohen. 3D G-CNNs for pulmonary nodule detection. *arXiv preprint arXiv:1804.04656*, 2018.

[7] Daniel Worrall and Gabriel Brostow. Cubenet: Equivariance to 3D rotation and translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 567–584, 2018.

[8] Erik J Bekkers, Remco Duits, Alexey Mashtakov, and Gonzalo R Sanguinetti. A PDE approach to data-driven sub-Riemannian geodesics in SE(2). *SIAM Journal on Imaging Sciences*, 8(4):2740–2770, 2015.

[9] Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2865–2873, 2015.

[10] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018.

[11] Erik J Bekkers. B-Spline CNNs on Lie groups. In *International Conference on Learning Representations*, 2019.

[12] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to Lie groups on arbitrary continuous data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3165–3176. PMLR, 13–18 Jul 2020.

[13] Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *Advances in Neural Information Processing Systems*, 32, 2019.

[14] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5028–5037, 2017.

[15] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2747–2755, Stockholmsmässan, Stockholm Sweden, July 2018. PMLR.

[16] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) equivariant representations with spherical CNNs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018.

[17] Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. In *Advances in Neural Information Processing Systems*, pages 14334–14345, 2019.

[18] Mercedes E. Paoletti, Juan M. Haut, Swalpa Kumar Roy, and Eligius M. T. Hendrix. Rotation equivariant convolutional neural networks for hyperspectral image classification. *IEEE Access*, 8:179575–179591, 2020.

[19] Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. *arXiv preprint arXiv:2106.06020*, 2021.

[20] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 09–15 Jun 2019.

[21] Stefanos Zafeiriou, Michael Bronstein, Taco Cohen, Oriol Vinyals, Le Song, Jure Leskovec, Pietro Liò, Joan Bruna, and Marco Gori. Guest editorial: Non-euclidean machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):723–726, 2022.

[22] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

[23] L. Sifre. *Rigid-Motion Scattering for Image Classification*. PhD thesis, Ecole Polytechnique Paris, CMAP, 2014.

[24] B. Smets, J. Portegies, E. Bekkers, and R. Duits. PDE-based group equivariant convolutional neural networks, 2020.

[25] R. Duits. *Perceptual organization in image analysis*. PhD thesis, Eindhoven University of Technology, 2005.

[26] M. H. J. Janssen, A. J. E. M. Janssen, E. J. Bekkers, J. Oliván Bescós, and R. Duits. Design and processing of invertible orientation scores of 3D images. *Journal of Mathematical Imaging and Vision*, 60(9):1427–1458, 2018.

[27] B. Franceschiello, A. Mashtakov, G. Citti, and A. Sarti. Geometrical optical illusion via sub-Riemannian geodesics in the roto-translation group. *Differential Geometry and its Applications*, 65:55 – 77, 2019.

[28] G. Citti and A. Sarti. A cortical based model of perceptual completion in the roto-translation space. *Journal of Mathematical Imaging and Vision*, 24:307–326, 05 2006.

[29] R. Duits and E. M. Franken. Left invariant parabolic evolution equations on SE(2) and contour enhancement via invertible orientation scores, part I: Linear left-invariant diffusion equations on SE(2). *Quarterly of Applied mathematics, AMS*, 68:255–292, June 2010.

[30] R. Duits and E. M. Franken. Left invariant parabolic evolution equations on SE(2) and contour enhancement via invertible orientation scores, part II: Nonlinear left-invariant diffusion equations on invertible orientation scores. *Quarterly of Applied mathematics, AMS*, 68:293–331, June 2010.

[31] J. Zhang, R. Duits, B.M. ter Haar Romeny, and G.R. Sanguinetti. Numerical approaches for linear left-invariant diffusions on SE(2), their comparisons to exact solutions, and their applications in retinal imaging. *Numerical Mathematics: Theory Methods and Applications*, 9(1):1–50, January 2016.

[32] U. Boscain, R. A. Chertovskih, J. P. Gauthier, and A. O. Remizov. Hypoelliptic diffusion and human vision: A semidiscrete new twist. *SIAM Journal on Imaging Sciences*, 7(2):669–695, 2014.

[33] M. Bertalmío, L. Calatroni, V. Franceschi, B. Franceschiello, and D. Prandi. A cortical-inspired model for orientation-dependent contrast perception: A link with Wilson-Cowan equations. In *Scale Space and Variational Methods in Computer Vision*, pages 472–484, Cham, 2019. Springer International Publishing.

[34] R. Duits, H. Fuehr, B.J. Janssen, L.M.J. Florack, and H.A.C. van Assen. Evolution equations on Gabor transforms and their applications. *ACHA*, 35(3):483–526, 2013.

[35] D. Barbieri, G. Citti, G. Cocci, and A. Sarti. A cortical-inspired geometry for contour perception and motion integration. *Journal of Mathematical Imaging and Vision*, 49(3):511–529, 2014.

[36] J. Petitot. The neurogeometry of pinwheels as a sub-riemannian contact structure. *Journal of Physiology-Paris*, 97(2):265–309, 2003. Neurogeometry and visual perception.

[37] M. Felsberg, P-E. Forssen, and H. Scharr. Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 209–222, 2006.

[38] P. Savadjiev, G.J. Strijkers, A.J. Bakermans, E. Piuze, S.W. Zucker, and K. Siddiqi. Heart wall myofibers are arranged in minimal surfaces to optimize organ function. *PNAS*, 109(24):9248–9253, 2012.

[39] Remco Duits, Erik Bekkers, and Alexey Mashtakov. Fourier transform on the homogeneous space of 3D positions and orientations for exact solutions to linear PDEs. *Entropy*, 21(1):38, 2019.

[40] P. Momayyez-Siahkal and K. Siddiqi. 3D stochastic completion fields for fiber tractography. In *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*, pages 178–185, June 2009.

[41] J. August and S.W. Zucker. Sketches with curvature: the curve indicator random field and markov processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):387–400, 2003.

[42] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3):574–591, 1959.

[43] W. H. Bosking, Y. Zhang, B. Schofield, and D. Fitzpatrick. Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex. *Journal of Neuroscience*, 17(6):2112–2127, 1997.

[44] D. J. Field, A. Hayes, and R. F. Hess. Contour integration by the human visual system: Evidence for a local "association field". *Vision Research*, 33(2):173–193, 1993.

[45] R. Duits, U. Boscain, F. Rossi, and Y. Sachkov. Association fields via cuspless sub-riemannian geodesics in se(2). *Journal of Mathematical Imaging and Vision*, 49(2):384–417, 2014.

[46] R. Duits, S. P. L. Meesters, J. Mirebeau, and J. M. Portegies. Optimal paths for variants of the 2d and 3d reeds-shepp car with applications in image analysis, 2016.

[47] G. Sanguinetti, E. Bekkers, R. Duits, M. H. J. Janssen, A. Mashtakov, and J. Mirebeau. Sub-riemannian fast marching in se(2). In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 366–374, Cham, 2015. Springer International Publishing.

[48] R. Duits, A. Ghosh, T. Dela Haije, and Y. Sachkov. *NeuroMathematics of Vision*, chapter Cuspless Sub-Riemannian Geodesics within the Euclidean Motion Group SE(d), pages 173–215. NeuroMathematics of Vision. Springer, 2014.

[49] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer Berlin Heidelberg, 2013.

[50] I. Biederman. Geon theory as an account of shape recognition in mind and brain. *The Irish Journal of Psychology*, 14(3):314–327, 1993.

[51] R. Duits, B. Smets, E. Bekkers, and J. Portegies. Equivariant deep learning via morphological and linear scale space pdes on the space of positions and orientations. In *Scale Space and Variational Methods in Computer Vision*, pages 27–39, Cham, 2021. Springer International Publishing.

[52] J. Jost. *Riemannian Geometry and Geometric Analysis*. Universitext. Springer International Publishing, 2017.

[53] J.M. Lee. *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing, 2019.

[54] B. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Graduate Texts in Mathematics. Springer, 2003.

[55] J.M. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer, 2003.

[56] J. Milnor. Curvatures of left invariant metrics on lie groups. *Advances in Mathematics*, 21(3):293–329, 1976.

[57] R. Duits, B. M. N. Smets, A. J. Wemmenhove, J. W. Portegies, and E. J. Bekkers. *Recent Geometric Flows in Multi-orientation Image Processing via a Cartan Connection*, pages 1–60. Springer International Publishing, Cham, 2021.

[58] R. Duits, E. J. Bekkers, and A. Mashtakov. Fourier transform on the homogeneous space of 3d positions and orientations for exact solutions to linear pdes. *Entropy*, 21(1), 2019.

[59] M. Schmidt and J. Weickert. Morphological counterparts of linear shift-invariant scale-spaces. *J. Math. Imaging Vis.*, 56(2):352–366, oct 2016.

[60] Z. Balogh, A. Engulatov, L. Hunziker, and O. Maasalo. Functional inequalities and hamilton–jacobi equations in geodesic spaces. *Potential Analysis*, 36:317–337, 02 2012.

[61] D. Azagra, J. Ferrera, and F. López-Mesas. Nonsmooth analysis and hamilton–jacobi equations on riemannian manifolds. *Journal of Functional Analysis*, 220(2):304–361, 2005.

[62] A. F. M. ter Elst and D. W. Robinson. Weighted subcoercive operators on Lie groups. *Journal of Functional Analysis*, 157:88–163, 1998.

[63] E. Bekkers, D. Chen, and J. Portegies. Nilpotent approximations of sub-riemannian distances for fast perceptual grouping of blood vessels in 2d and 3d. *Journal of Mathematical Imaging and Vision*, 60, 07 2018.

[64] R. Duits, L. Florack, J. de Graaf, and B. ter Haar Romeny. On the axioms of scale space theory. *Journal of Mathematical Imaging and Vision*, 20:267–298, 05 2004.