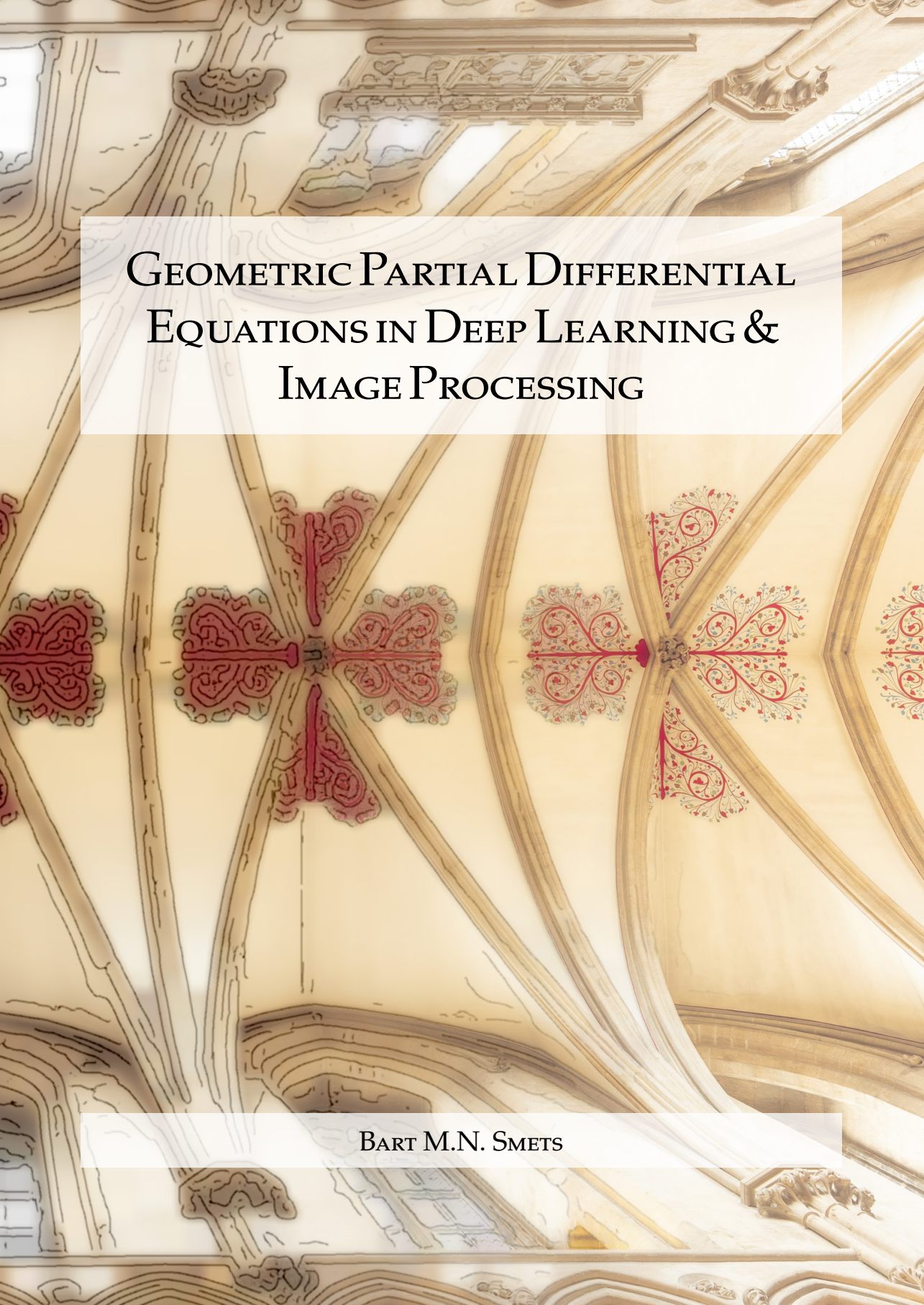


Geometric Partial Differential Equations in Deep Learning & Image Processing

Bart M.N. Smets

GEOMETRIC PARTIAL DIFFERENTIAL EQUATIONS IN DEEP LEARNING & IMAGE PROCESSING

BART M.N. SMETS

September 2024

COLOPHON

This document was typeset using the typographical `classicthesis` template for LaTeX developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. `classicthesis` is available at:

https://bitbucket.org/amiede/classicthesis/

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-6133-9

GEOMETRIC PARTIAL DIFFERENTIAL EQUATIONS IN DEEP LEARNING & IMAGE PROCESSING

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr. S.K. Lenaerts, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op vrijdag 20 september 2024 om 13:30 uur

door

BART MARTIN NICOLAS SMETS

geboren te Neerpelt, België

Dit proefschrift is goedgekeurd door de promoteren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. E.R. van den Heuvel
promotor:	dr.ir. R. Duits
co-promotor:	dr. J.W. Portegies
leden:	prof.dr. C. Brune (University of Twente)
	prof.dr. G. Citti (Università di Bologna)
	prof.dr. C. Schnörr (Universität Heidelberg)
	dr. A. Di Bucchianico
	prof.dr.ir. B. Koren

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Aan Oem & Bompa.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

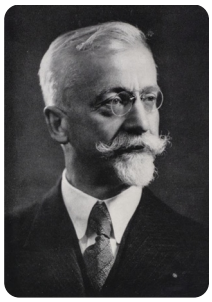
INTRODUCTION

Image analysis plays an important role in many parts of modern society. In fields such as medicine, defense and industrial automation it is critical to have image processing techniques that are efficient and reliable since they form the basis of high-stakes decisions. In this thesis we contribute to the field of image analysis by developing new techniques that synthesize two approaches to image processing: the *geometric partial differential equation* approach and the *deep learning* approach.

While in recent times, deep learning has gotten most of the spotlight, each approach has certain strengths and weaknesses. Our aim is to develop a hybrid approach that captures the strengths of both approaches for those applications where either pure approach is not suitable or does not provide the right tradeoffs.

Gottfried Wilhelm Leibniz 1646–1716.

Sophus Lie 1842–1899.



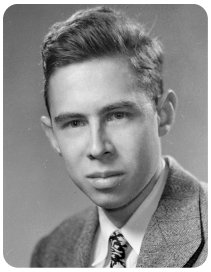
Élie Cartan 1869–1951.

GEOMETRIC PARTIAL DIFFERENTIAL EQUATIONS. The study of *partial differential equations* (PDEs) constitutes one of the largest fields in mathematics, going back as far as *Leibniz* (CAJORI, 1928). In modern times, this body of knowledge has also been leveraged for the development of image processing techniques. While a digital image consist of a grid of discrete pixels, the methodology of using a continuous model with PDEs has been a source of many novel image processing methods (MARAGOS, 2005). The use of continous PDE models is motivated by more intuitive mathematical modeling with a connection to physics. It also does away with the resolution and sampling dependence of the discrete methods, thus staying more faithful to the Euclidean geometry that underlies the problem.

A trend in this field of continuous models is the incorporation of an increasing amount of *differential geometry* (SAPIRO, 2006). In particular the theory of *Lie groups*—named for *Sophus Lie*—has enjoyed a lot of attention since it provides a framework to formalize the symmetries that are natural in imaging. Thanks to the work pioneered by *Élie Cartan*, it is possible to make coordinate-free for-

mulations of PDEs on Lie groups and their homogeneous spaces, giving rise to *geometric partial differential equations* as natural objects of study in image processing.

Going even further, this type of mathematical theory also underpins the field of *neurogeometry* (Citti and Sarti, 2014; Petitot, 2017) where it is used to model the functional architecture of the visual areas of our brain. In this manner geometric PDEs can be said to provide a bridge between image processing as our brain performs it on the one hand and image processing as we engineer it on the other.



Frank Rosenblatt 1928–1971, inventor of the perceptron, arguably making him the father of deep learning.

DEEP LEARNING. Using artificial neural networks for image processing is a comparatively young idea that takes a very different approach (Egmont-Petersen, de Ridder, and Handels, 2002). It leverages modern compute power and the availability of vast datasets to train very general and unstructured models to perform the task at hand. Nonetheless, researchers in this field have also taken an interest in geometry (Bronstein, Bruna, LeCun, Szlam, and Vandergheynst, 2017), particularly Lie group theory, just like in the geometric PDE approach. In the nascent field of *geometric deep learning* the mathematical theory is applied to construct network architectures with geometric symmetries that fit the application being considered.

Yann LeCun b. 1960, pioneer in using deep learning for machine vision.

OUR THESIS. Our main goal is to provide a synthesis between *geometric partial differential equations* and *neural networks*. Doing so, we also provide a stronger link between neural networks and neurogeometry. This work is already partially done in the sense that geometric deep learning has incorporated many of the geometric tools we need. But it has not incorporated *partial differential equations*. This then will be the main contribution of this thesis: the development of a hybrid neural network architecture that we refer to as a *PDE-based group equivariant convolutional neural network*, or *PDE-G-CNN* for short.

We intend to take linear and morphological PDEs from classic image processing and introduce them as operators into neural networks. These PDEs have parameters which would have to be chosen by hand in classic usage, but in a neural network they become the *trainable weights*. This way, the PDEs will provide a

strong geometric prior but the data will determine exactly how the PDEs will behave. As the morphological PDEs we propose are not linear we can use them as a network's nonlinearities and remove the need for traditional activation functions, taking away some of the arbitrariness and mystification in neural network design.

The practical motivation for synthesizing geometric PDEs and neural networks is that we hope to balance the main benefits of both.

- *Performance.* The popularity of neural networks has mostly been driven by their superior performance when compared with legacy methods. We aim to retain this high level of performance for image processing applications.
- *Interpretability.* What a trained neural networks does to accomplish its task is notoriously hard to interpret. Constructing them in terms of PDEs gives a concrete geometric interpretation for its operations.
- *Strong prior/weak data.* PDEs add a strong additional prior on top of only using geometric symmetries. Assuming this prior aligns with the application, we can expect to get good results even if the available training data is weak, i.e. has few and/or noisy samples.
- *Manual tuning.* Classic use of geometric PDEs requires manual fine-tuning of its parameters which can easily lead to poor generalization beyond the few samples that were manually examined during development. In a neural network all the parameters will be systematically trained in way that has been observed to generalize well.

To summarize, we can say that the PDE-G-CNNs we propose automate the geometric image processing on Lie groups (Citti, Franceschiello, Sanguinetti, and Sarti, 2016; Duits and Burgeth, 2007; Franken and Duits, 2009) and conversely it structures neural networks using geometric PDEs to obtain better performance, interpretability and data-efficiency (Bellaard, Bon, Pai, Smets, and Duits, 2023; Pai, Bellaard, Smets, and Duits, 2023; Smets, Portegies, Bekkers, and Duits, 2022).

LIETORCH. We implemented our PDE-G-CNN algorithms as an extension to the widely used *PyTorch* deep learning framework (PASZKE et al., 2019). Our *LieTorch* package is available at www.gitlab.com/bsmetsjr/lietorch and contains both CPU and CUDA implementations of our algorithms that are exposed as standard PyTorch modules.

ADDITIONAL RESEARCH. In addition to the main line of research we outlined above we will explore two additional topics.

The first is *semiring activation*, which follows up on the fact that in PDE-based networks we could dispense with traditional activation functions since we had morphological PDEs to serve as nonlinearities. We can still have *trainable nonlinearities* with some of the same structure of a morphological PDE but without actually being described by a PDE. This line of thinking introduces a systematic approach for building trainable nonlinear operators based on the algebraic structure of a semiring and experiments with them in neural networks.

The second topic is *total variation and mean curvature denoising in* SE(2). Total variation and mean curvature PDEs (VOGEL and OMAN, 1996; ZHU and CHAN, 2012) have previously been leveraged as a basis for image denoising algorithms. Our contribution consists of formulating these PDEs on the Lie group SE(2) and using them to denoise the *orientation score* (DUITS, DUITS, VAN ALMSICK, and TER HAAR ROMENY, 2007) of an image rather than the image itself. This produces a denoising method that deals very well with edges and crossings in an image and produces very high signal-to-noise ratios. We also extend this method to deal with volumetric data such as *diffusion-weighted magnetic resonance imaging* data.

OUTTAKES. We will not cover any works for which I was not a primary author. Here we give an overview of these works.

- Geometric flows in image processing via a Cartan connection in DUITS, SMETS, WEMMENHOVE, PORTEGIES, and BEKKERS (2021).
- Follow-up research on various aspects of PDE-G-CNNs:
  - Sub-Riemannian versions of PDE-G-CNNs and their analysis in BELLAARD, BON, PAI, SMETS, and DUITS (2023).

  - U-Net variant of PDE-G-CNNs and its data-efficiency benefits in Pai, Bellaard, Smets, and Duits (2023).
  - An axiomatic basis for PDE-G-CNNs when restricted to $\mathbb{R}^2$ in Bellaard, Sakata, Smets, and Duits (2024).

- Geodesic tracking of vascular trees:
  - Tracking vascular trees with data-driven Cartan connections in van den Berg, Smets, Pai, Mirebeau, and Duits (2024).
  - Tracking vascular trees with optical and TVF enhancement in van den Berg, Zhang, Smets, Berendschot, and Duits (2023).

While I contributed to these works, the large majority of the credit goes to their primary authors.

OUTLINE. This thesis consists of three parts.

- In Part I we cover the necessary theoretical preliminaries. These cover a subset of differential geometry that underpins the later parts. Some of the material in this part is adapted from our lecture notes for the course *Mathematics of Neural Networks* (Smets, 2023), though most of it is newly written for this thesis.

- In Part II we cover deep learning.
  - Chapter 5 is an introduction to *group equivariant convolutional neural networks* and a survey of the research around them. It was newly written for this thesis.
  - Chapter 6 covers *PDE-G-CNNs* and is an expanded version of the theoretical parts of Duits, Smets, Bekkers, and Portegies (2021) and Smets, Portegies, Bekkers, and Duits (2022).
  - Chapter 7 covers the experimental results of Smets, Portegies, Bekkers, and Duits (2022).
  - Chapter 8 covers *semiring activation* and is an expanded version of Smets, Donker, Portegies, and Duits (2024).

- In Part III, which only contains Chapter 9, we cover our work on *image denoising*, it is based on Duits, St-Onge, Portegies, and Smets (2019) and Smets, Portegies, St-Onge, and Duits (2021).

Part I

THEORETICAL PRELIMINARIES

This first part of the thesis is a summary of the geometric theory used in our research. Parts II and III, while covering different subjects, share common underpinnings that we factored out here. We cover that subset of differential geometry which is relevant to our research so that a reader less familiar with differential geometry may also use it as a reference while reading the later parts. We specifically want to cover the tools required to formulate partial differential equations on homogeneous spaces and be able to talk about equivariances in that setting. We also establish notation for the rest of the work.

Most results discussed are standard results referencing established works, with the exception of Section 3.3 which contains some minor novelties. This part is partially based on the lecture notes for the course *Mathematics of Neural Networks* that we set up and taught from 2021 to 2023 at the Eindhoven University of Technology.

1

RIEMANNIAN GEOMETRY

In our research, both in deep learning and image processing, we consider data that naturally lives on manifolds and with which we work using differential geometric tools. In this chapter we recapitulate the basic tools of Riemannian geometry and some of its generalizations that we will use throughout. A thorough reference to this material is found in Lee (2012, 2018).

1.1 DIFFERENTIABLE MANIFOLDS

A *topological manifold* of dimension n is a second-countable Hausdorff space M which is locally homeomorphic to an open subset of $\mathbb{R}^n$. Being locally homomorphic to an open subset of $\mathbb{R}^n$ means that for each $p \in M$ there is a homeomorphism $u : U \to u(U) \subseteq \mathbb{R}^n$ where U is an open neighborhood of p and the image $u(U)$ is an open subset of $\mathbb{R}^n$. We call (U, u) a *chart* on M.

In a Hausdorff space each pair of points have neighborhoods that are disjoint. A second-countable space has a countable basis for its topology.

A family $(U_\alpha, u_\alpha)_{\alpha \in A}$ of charts on M such that $\bigcup_{\alpha \in A} U_\alpha = M$ is called an *atlas*. The mappings $u_{\alpha\beta} := u_\beta \circ u_\alpha^{-1} : u_\alpha(U_{\alpha\beta}) \to u_\beta(U_{\alpha\beta})$ are called the *transition maps* of the atlas, where $U_{\alpha\beta} = U_\alpha \cap U_\beta$. The relationships between the manifold, charts and transition maps is illustrated in Figure 1.1.

An atlas $(U_\alpha, u_\alpha)_{\alpha \in A}$ for a manifold M is called a C^k *atlas* if all transition maps $u_{\alpha\beta}$ are in $C^k\left(u_\beta(U_{\alpha\beta}), u_\alpha(U_{\alpha\beta})\right)$. Two C^k atlases are called C^k *equivalent* if their union is again a C^k atlas. An equivalence class of all such C^k atlases we call a C^k *structure* on M or just a *differentiable structure* on M if the order k is implied. By a C^k *manifold* we will mean a topological manifold equipped with a C^k structure, when we say *differentiable manifold* we mean a C^k manifold where we imply some k. With a chart of a manifold

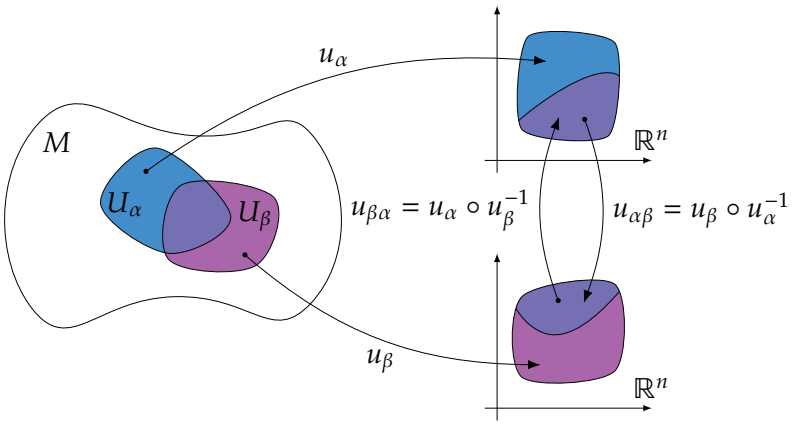


FIGURE 1.1: A manifold M with two charts (U_α, u_α) and (U_β, u_β) and their transitions maps $u_{\alpha\beta}$ and $u_{\beta\alpha}$.

we will then mean a chart belonging to one of the atlases of its differentiable structure.

In this work, when we say *manifold* without further elaboration we mean a C^∞ manifold, and we use the term *smooth* synonymously with C^∞.

We can go even further than smooth manifolds and define such things as real analytic C^ω manifolds, but we will not consider these here.

A subset N of a manifold M is called a *submanifold* of M, if for all $p \in N$ there exists a chart (U, u) of M such that $u(U \cap N) = u(U) \cap (\mathbb{R}^{n'} \times \{0\})$, where $\mathbb{R}^{n'} \times \{0\} \hookrightarrow \mathbb{R}^{n'} \times \mathbb{R}^{n-n'}$. Then N is a n'-dimensional manifolds since it can be equipped with an atlas of charts of the type $(U \cap N, u|_{U\cap N})$ for each $p \in N$.

1.2 SMOOTH MAPS & DIFFEOMORPHISMS

A map $f : M \to N$ between two C^k manifolds is said to be C^k if for each $p \in M$ and each chart (V, v) on N with $f(p) \in V$ there exists a chart (U, u) on M such that $p \in U$, $f(U) \subseteq V$ and $v \circ f \circ u^{-1}$ is C^k. The relationship between f and its proxy $v \circ f \circ u^{-1}$ is illustrated in Figure 1.2. We will denote the set of C^k maps between M and N with $C^k(M, N)$. In the case of C^∞ we simply talk about *smooth maps*. When $N = \mathbb{R}$ we use the abbreviation $C^\infty(M) := C^\infty(M, \mathbb{R})$.

The C^k manifolds together with their C^k diffeomorphisms form the category of manifolds Man^k. This is a category since the composition of two C^k maps is again a C^k map.

A C^k map $f : M \to N$ is called a C^k *diffeomorphism* if there exists an inverse $f^{-1} : N \to M$ that is also a C^k map. If there exists a diffeomorphism between two manifolds we say they are *diffeomorphic*.

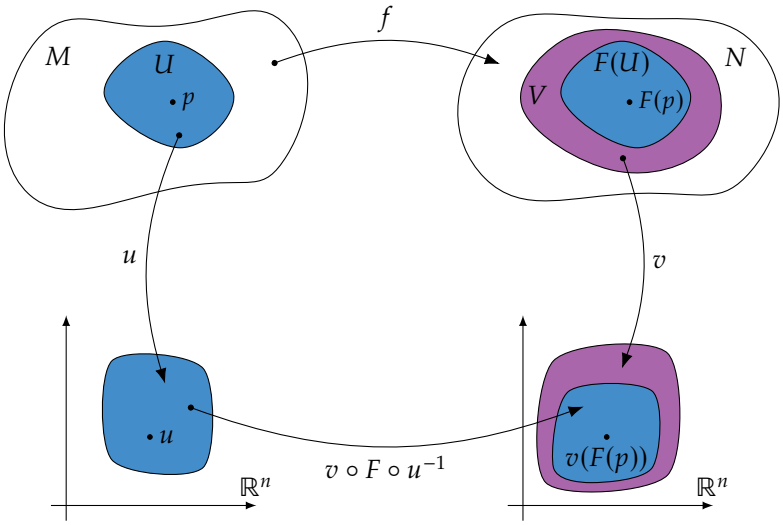


FIGURE 1.2: A C^k map f between two manifolds and its representation $v \circ F \circ u^{-1}$ between chart co-domains.

A map $f : M \to N$ between manifolds of the same dimension is called a *local diffeomorphism* if each $p \in M$ has an open neighborhood U so that $f|_U : U \to f(U) \subset N$ is a diffeomorphism. As an example: S^1 and $\mathbb{R}$ are not diffeomorphic since there exists no continuous bijection between them, but they are locally diffeomorphic.

1.3 TANGENT BUNDLE

There are many equivalent ways of defining tangent vector, we use the algebraic definition here.

DERIVATIONS. Let M be a n-dimensional manifold. A *derivation* with base point $p \in M$ acting on $C^\infty(M)$ is a linear operator $X_p : C^\infty(M) \to \mathbb{R}$ that additionally satisfies *Leibniz's rule* at its base point:

$$X_p(f_1 f_2) = f_1(p)\, X_p f_2 + f_2(p)\, X_p f_1, \tag{1.1}$$

for all $f_1, f_2 \in C^\infty(M)$ and where $f_1 f_2$ is their pointwise multiplication. If the base point is clear from context we may drop the subscripted p and just write X. As a corollary to Leibniz's rule we have that X_p applied to a constant function yields zero:

$$X_p(1) = X_p(1 \cdot 1) = 1 \cdot X_p(1) + 1 \cdot X_p(1) = 2 \cdot X_p(1),$$

which implies that $X_p(1) = 0$. By linearity we see that if f is locally constant around p then $X_p f = 0$.

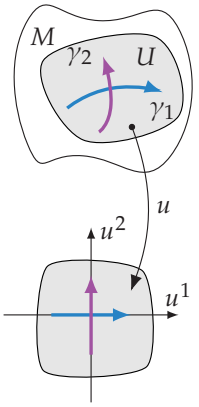
It is easy to check that for any given base point $p \in M$ the derivations at that point form a vector space, we call this vector space the *tangent space* at p and its elements the *tangent vectors* at p. We denote the tangent space of M at the point p with T_pM.

We can make derivations more concrete by constructing some with the help of a chart.

THE TANGENT BUNDLE. For a manifold M we call the disjoint union of all its tangent spaces the *tangent bundle*

$$TM := \bigsqcup_{p \in M} T_pM. \tag{1.2}$$

Per LEE (2012, Prop. 3.18) the tangent bundle TM of a smooth n dimensional manifold has a natural topology and smooth structure that makes it into a smooth vector bundle, which can be shown by constructing a smooth atlas for TM based on the coordinate basis, as we will see next.



Moving along the axes of a chart yields curves on the manifold which can be identified with derivations at $u^{-1}(0)$.

COORDINATE BASIS. Let $n = \dim M$ and without loss of generality let (U, u) be a chart centered at p, i.e. $u(p) = 0$. Now define a set of curves $\gamma_i^u : I \to U$ where I is some interval containing 0 by moving along the axes of the chart:

$$\gamma_i^u(t) := u^{-1}(0, \dots, 0, \underbrace{t}_{\substack{\uparrow \\ i-th \\ place}}, 0, \dots, 0),$$

clearly $\gamma_i^u(0) = p$ for all $i = 1 \dots n$. Based on these curves we can define a set of operators

$$(\partial_{u^i})_p : C^\infty(M) \to \mathbb{R}$$

as follows:

$$(\partial_{u^i})_p := \lim_{t \to 0} \frac{f(\gamma_i^u(t)) - f(p)}{t} \tag{1.3}$$

for all $f \in C^\infty(M)$, where we include u in the notation to emphasize the dependence on the choice of chart.

Clearly these operators are linear maps from $C^\infty(M)$ to $\mathbb{R}$ and also satisfy Leibniz's rule (1.1) since they are just the usual limit

of a difference quotient of an $\mathbb{R} \to \mathbb{R}$ function, hence we can conclude that they are in fact tangent vectors: for $i = 1 \dots n$ we have:

$$(\partial_{u^i})_p \in T_p M.$$

We call these tangent vectors the coordinate basis tangent vectors of the chart (U, u), we will check that they in fact form a basis.

This set of n tangent vectors is linearly independent. To see this suppose that

$$\sum_{i=1}^{n} v^i \, (\partial_{u^i})_p = 0$$

for some scalars $v^i \in \mathbb{R}$. Recall that (U, u) is a smooth map and so its component functions are smooth real-valued functions: $u^i \in C^\infty(U)$. Hence, we can let the above operator act on them to find that for all $j = 1 \dots n$:

$$0 = \sum_{i=1}^{n} v^i \, (\partial_{u^i})_p \, u^j = \sum_{i=1}^{n} v^i \delta^j_i = v^j,$$

from which we conclude that these tangent vectors are linearly independent.

If $f \in C^\infty(M)$ then $f \circ u^{-1} \in C^\infty(u(U) \subseteq \mathbb{R}^n)$ and so we can apply *Hadamard's lemma* around the origin to express the function f as:

$$f \circ u^{-1}(u(q)) = f \circ u^{-1}(0) + \sum_{i=1}^{n} (u^i(q) - 0) \, h_i(u^i(q)),$$

where $q \in U$, $u^i \in C^\infty(U)$ is the i-th component function of u and $h_i \in C^\infty(u(U) \subseteq \mathbb{R}^n)$ for each $i = 1 \dots n$. Simplified we get:

$$f(q) = f(p) + \sum_{i=1}^{n} u^i(q) \, h_i(u^i(q)).$$

Applying the coordinate basis tangent vectors (1.3) of (U, u) to this decomposition of f yields

$$\begin{aligned}(\partial_{u^i})_p f &= 0 + \sum_{j=1}^{n} (\partial_{u^i})_p \left(u^j(\,\cdot\,)\, h_j(u^j(\,\cdot\,)) \right) \\ &= \sum_{j=1}^{n} u^j(p)\ (\partial_{u^i})_p\, h_j(u^j(\,\cdot\,)) + h_j(u^j(p))\ (\partial_{u^i})_p\, u^j(\,\cdot\,) \\ &= \sum_{j=1}^{n} h_j(0)\, \delta_i^j \\ &= h_i(0).\end{aligned}$$

Now the application of some derivation $X_p \in T_pM$ to this decomposition of f yields

$$\begin{aligned}X_p f &= 0 + \sum_{i=1}^{n} X_p \left(u^i(\,\cdot\,)\, h_i(u^i(\,\cdot\,)) \right) \\ &= \sum_{i=1}^{n} u^i(p)\, X_p h_i(u^i(\,\cdot\,)) + h_i(u^i(p))\, X_p u^i \\ &= \sum_{i=1}^{n} h_i(0)\, X_p u^i.\end{aligned}$$

We recognize $h_i(0)$ as the value of applying the coordinate basis tangent vector of (U, u) to f and $X_p u^i$ is just a real number, so we obtain:

$$X_p f = \sum_{i=1}^{n} X_p u^i\ (\partial_{u^i})_p\, f,$$

or more succinctly:

$$X_p = \sum_{i=1}^{n} X_p u^i\ (\partial_{u^i})_p\,. \tag{1.4}$$

This means that we can write any derivation at p as a linear combination of the n coordinate basis tangent vectors of a chart containing p, this proves in particular that $\dim T_pM = \dim M$ for all $p \in M$.

THE TANGENT BUNDLE AS A SMOOTH MANIFOLD. Given a chart (U_α, u_α) on M we can construct a chart $(\pi_M^{-1}(U_\alpha), \mathrm{d}u_\alpha)$ on TM where $\mathrm{d}u_\alpha : \pi_M^{-1}(U_\alpha) \to u_\alpha(U_\alpha) \times \mathbb{R}^n$ is given by

$$\mathrm{d}u_\alpha(X_p) := \left(u_\alpha(\pi_M(X_p)),\ X_p u_\alpha^1,\ \ldots,\ X_p u_\alpha^n\right).$$

If we have another (overlapping) chart (U_β, u_β) and its induced tangent bundle chart $\mathrm{d}u_\beta : \pi_M^{-1}(U_\beta) \to u_\beta(U_\beta) \times \mathbb{R}^n$ we can look at the associated transition maps

$$\mathrm{d}u_{\alpha\beta} = \mathrm{d}u_\beta \circ \mathrm{d}u_\alpha^{-1} : u_\alpha(U_{\alpha\beta}) \times \mathbb{R}^n \to u_\beta(U_{\alpha\beta}) \times \mathbb{R}^n.$$

Let $(x, v) = (x^1, \ldots, x^n, v^1, \ldots, v^n) \in u_\alpha(U_{\alpha\beta}) \times \mathbb{R}^n \subseteq \mathbb{R}^{2n}$ then

$$\begin{aligned}\mathrm{d}u_{\alpha\beta}(x, v) &= \mathrm{d}u_\beta\left(\sum_{i=1}^n v^i \left(\partial_{u_\alpha^i}\right)_{u_\alpha^{-1}(x)}\right)\\ &= \left(u_{\alpha\beta}(x),\ J_x v\right),\end{aligned}$$

where $J_x \in \mathbb{R}^{n\times n}$ is the Jacobian matrix of $u_{\alpha\beta}$ at x:

$$J_x = \begin{bmatrix} (\partial_1)_x\, u_{\alpha\beta}^1 & \cdots & (\partial_n)_x\, u_{\alpha\beta}^1 \\ \vdots & \ddots & \vdots \\ (\partial_1)_x\, u_{\alpha\beta}^n & \cdots & (\partial_n)_x\, u_{\alpha\beta}^n \end{bmatrix}.$$

Clearly then these transition maps are smooth and so give us a smooth structure on the tangent bundle TM. We choose the topology of TM so that all charts $\mathrm{d}u_\alpha$ are homeomorphisms, this again yields a separable Hausdorff space and so the tangent bundle TM is a smooth manifold just like its base space M is.

The coordinate maps given by

$$(p, X_p) \mapsto \left(u^1(p), \ldots, u^n(p), X_p u^1, \ldots, X_p u^n\right)$$

are also referred to a the *natural coordinates* on TM.

SMOOTH VECTOR FIELDS. Now that we have established that the tangent bundle is itself a smooth manifold we can talk about the smoothness of vector fields. A *smooth vector field* X on a manifold M is a smooth section of the tangent bundle TM, i.e. $X : M \to TM$ is smooth and $\pi_M \circ X = \mathrm{id}_M$.

When a vector field is only smooth on an open subset of M we say it is *locally smooth*. We denote the set of all smooth vector fields as $\Gamma(TM)$, i.e. the set of smooth sections of TM. With the usual addition and scalar multiplication $\Gamma(TM)$ is a real vector space.

Previously we identified tangent vector with derivations at particular points of the manifold, we can consequently also identify a vector field with a field of derivations over the whole manifold. We can look at a vector field X as a linear map

$$X : C^\infty(M) \to C^\infty(M)$$

that satisfies the Leibniz rule with respect to pointwise multiplication for every point of the manifold:

$$X(f_1 f_2) = f_1 X f_2 + f_2 X f_1,$$

for all $f_1, f_2 \in C^\infty(M)$. We will use subscripting of a vector field with a manifold element to indicate the tangent vector at that point since we apply a vector field to a function f yielding another function. So the notation becomes $(Xf)(p) = X_p f$ rather than $(Xf)(p) = X(p)f$ where X is applied to both an element of $C^\infty(M)$ and subsequently M, which is confusing notation.

LIE BRACKET. If $X, Y \in \Gamma(TM)$ then the mapping $f \mapsto X(Yf) - Y(Xf)$ is again a derivation of $C^\infty(M)$ and so a vector field on M. We call this vector field the *commutator* or *Lie bracket* of the vector fields X and Y and write

$$[X, Y] := X \circ Y - Y \circ X,$$

where the composition is seen as a composition of linear maps. The Lie bracket $[\,\cdot\,,\,\cdot\,] : \Gamma(TM)^2 \to \Gamma(TM)$ has the following properties for all $X, Y, Z \in \Gamma(TM)$ and all $f \in C^\infty(M)$:

$[X, Y] = -[Y, X]$ (anti-symmetric),

$[X, [Y, Z]] = [[X, Y], Z] + [Y, [X, Z]]$ (the Jacobi identity),

$[fX, Y] = f[X, Y] - (Yf)X$,

$[X, fY] = f[X, Y] + (Xf)Y$.

COTANGENT BUNDLE. Let M be a manifold and $p \in M$, then the *cotangent space* of M at p is the dual of the tangent space, i.e.

$$T_p^* M := (T_p M)^*.$$

Since $T_p M$ is finite dimensional it follows that $T_p M$ and $T_p^* M$ are isomorphic as vector spaces. If the tangent vectors

$$\left\{(\partial_{u^i})_p\right\}_{i=1}^n$$

are the coordinate basis vectors induced by a chart (U, u), then the dual basis is denoted by $\{(\mathrm{d}u^i)_p\}_{i=1}^n$. Since it is the dual basis we have

$$\left(\mathrm{d}u^i\right)_p \left((\partial_{u^j})_p\right) = \delta^i_j.$$

Just like for the tangent spaces, we call the disjoint union of all the cotangent spaces the cotangent bundle:

$$T^* M := \coprod_{p \in M} T_p^* M,$$

Note that the dual coordinate basis covectors $(\mathrm{d}u^i)_p$ are exactly the tangential components of the tangent bundle chart we constructed previously.

which, when equipped with the unique topology and smooth structure per LEE (2012, Prop 11.9), is again a smooth vector bundle.

Each scalar function $f \in C^\infty(M)$ induces a special section of the cotangent bundle called its differential $\mathrm{d}f \in \Gamma(TM)$, which is given by

$$\mathrm{d}f(X) := Xf$$

for $X \in \Gamma(TM)$. We denote pointwise evaluation with different notations depending on the context:

$$(\mathrm{d}f)_p(X) := \mathrm{d}f(X)(p) := X_p f.$$

Naturally we can use the tangent and cotangent spaces to construct tensor spaces. The *tensor space* $T_p^{(s,r)} M$ is defined as

$$T_p^{(s,r)} M := T^{(s,r)}(T_p M) := \underbrace{T_p M \otimes \cdots \otimes T_p M}_{s \text{ copies}} \otimes \underbrace{T_p^* M \otimes \cdots \otimes T_p^* M}_{r \text{ copies}}.$$

As with the tangent and cotangent spaces, the disjoint union of these tensor spaces gives us the *tensor bundle* $T^{(s,r)} M$ when equipped with the natural topology and smooth structure inherited from the tangent and cotangent bundles. The set of smooth sections of any tensor bundle we denote as $\Gamma(T^{(s,r)} M)$.

1.4 PUSHFORWARD & PULLBACK

Smooth maps between manifolds allows (co-)tangent vectors to be transported along them in a natural manner. Let M and N be manifolds and let $\phi : M \to N$ be a smooth map. The *pushforward* of ϕ at a point $p \in M$ is a linear map $\phi_*|_p : T_pM \to T_{\phi(p)}N$ given by

$$\left(\phi_*\big|_p (X_p)\right) f := X_p \left(f \circ \phi\right)$$

for any $f \in C^\infty(N)$ and $X_p \in T_pM$. Naturally the pushforward induces a smooth map between tangent bundles $\phi_* : TM \to TN$ in a way that makes the following diagram commute:

$$\begin{array}{ccc} TM & \xrightarrow{\phi_*} & TN \\ {\scriptstyle \pi_M}\downarrow & & \downarrow{\scriptstyle \pi_N} \\ M & \xrightarrow{\phi} & N \end{array}$$

The pushforward of a composed map obeys a type of chain rule: suppose $\phi : M \to N$ and $\psi : N \to P$ are both smooth maps then $(\psi \circ \phi)_*|_p = \psi_*|_{\phi(p)} \circ \phi_*|_p$.

Similar to the pushforward of tangent vectors we have the *pullback* of cotangent vectors. The pullback $\phi^*|_p : T^*_{\phi(p)}N \to T^*_pM$ of ϕ at p is given by:

$$\left(\phi_*\big|_p \, \omega_{\phi(p)}\right) X_p := \omega_{\phi(p)}(\phi_*\big|_p \, X_p),$$

for any $X_p \in T_pM$ and $\omega_{\phi(p)} \in T^*_{\phi(p)}N$. Higher order contravariant tensor fields can be pulled back in the same way. Let $\omega \in \Gamma(T^{(0,s)}M)$ then we define

$$\left(\phi^*\big|_p \, \omega_{\phi(p)}\right)\left((X_1)_p, \ldots, (X_s)_p\right) := \omega_{\phi(p)}\left(\phi_*\big|_p \, (X_1)_p, \ldots, \phi_*\big|_p \, (X_s)_p\right)$$

for all $X_1, \ldots, X_s \in \Gamma(TM)$, this will be useful for working with Riemannian metrics later.

Only in the case that $\phi : M \to N$ is a diffeomorphism does it make sense to pushforward cotangent vectors and pullback tangent vectors as follows:

$$\phi^*\big|_p \, Y_{\phi(p)} := \left(\phi^{-1}\right)_*\Big|_{\phi(p)} Y_{\phi(p)} \quad \text{and} \quad \phi_*\big|_p \, \xi_p := \left(\phi^{-1}\right)^*\Big|_{\phi(p)} \xi_p \tag{1.5}$$

for any $Y_{\phi(p)} \in T_{\phi(p)}M$ and $\xi_p \in T_p^*M$.

Usually explicitly writing the restriction to a certain point of a pushforward or pullback is redundant as it can be inferred from its argument, hence we usually just write $\phi_*(X_p)$ instead of $\phi_*|_p(X_p)$ for pushforwards and similarly for pullbacks.

FUNCTIONS & CURVES. Let $f \in C^\infty(N)$ then we say the pullback $\phi^* f \in C^\infty(M)$ of f is given by $\phi^* f := f \circ \phi$. Similarly, if we have a curve $\gamma : \mathbb{R} \to M$ then we say its pushforward $\phi_*\gamma : \mathbb{R} \to N$ is given by $\phi_*\gamma = \phi \circ \gamma$.

1.5 IMMERSION, SUBMERSION AND EMBEDDING

A smooth map $\phi : M \to N$ between two smooth manifolds M and N is called an *immersion* if its pushforward $\phi_*|_p : T_pM \to T_{\phi(p)}N$ is injective at all $p \in M$. Conversely, ϕ is called a *submersion* if its pushforward is surjective at all points of M.

Naturally, when ϕ is a immersion then $\dim M \leq \dim N$ and when ϕ is a submersion then $\dim M \geq \dim N$.

A smooth map $\phi : M \to N$ is an *embedding* when its image $\phi(M)$ is a submanifold and the map $M \to \phi(M)$ is a diffeomorphism.

CURVES. A type of map that we often encounter is a curve $\gamma \in C^\infty(I, M)$ where $I \subseteq \mathbb{R}$ is some interval (open or closed). A basis for $\Gamma(TI)$ is given by the tangent vector field $\mathrm{d}/\mathrm{d}t$ that acts on smooth functions on I in the obvious manner

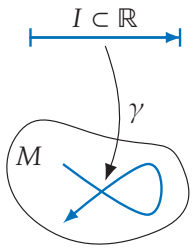
$$\left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_t f := \lim_{h \to 0} \frac{f(t+h) - f(t)}{h}.$$

All $X \in \Gamma(TI)$ can then be written as $X_t = X^1(t)\,(\mathrm{d}/\mathrm{d}t)_t$ for some $X^1 \in C^\infty(I)$.

As is conventional, we use the dot-notation to indicate the pushforward of these tangent vectors along the curve to the manifold as follows:

$$\dot{\gamma}(t) := \gamma_*|_t \left(\left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_t\right), \tag{1.6}$$

we call $\dot{\gamma}$ a *vector field along a curve* γ since $\pi_M \circ \dot{\gamma} = \gamma$. The appli-



A smooth curve that is an immersion but not necessarily an embedding.

cation of this vector field to a smooth function $f \in C^\infty(M)$ is then given by

$$\dot\gamma(t)f = \left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_t (f \circ \gamma) = \lim_{h\to 0} \frac{f(\gamma(t+h)) - f(\gamma(t))}{h}.$$

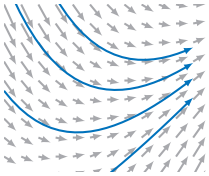
The pushforward γ_* is injective at $t \in I$ provided $\dot\gamma(t) \neq 0$ since for any $X_t, Y_t \in T_tI$ there exists $X^1, Y^1 \in \mathbb{R}$ so that $X_t = X^1 \left(\mathrm{d}/\mathrm{d}t\right)_t$ and $Y_t = Y^1 \left(\mathrm{d}/\mathrm{d}t\right)_t$. Then $\gamma_* X_t = \gamma_* Y_t$ implies $X^1 = Y^1$ and so also $X_p = Y_p$.

Consequently, smooth curves with $\dot\gamma(t) \neq 0$ for all $t \in I$ are immersions of some interval of the real line. However, they are not always embeddings since their image $\gamma(I) \subset M$ need not be a submanifold, see the margin figure for an example.

1.6 FLOWS

INTEGRAL CURVES. A smooth curve $\gamma \in C^\infty(I, M)$ for some open interval of $\mathbb{R}$ is called an *integral curve* or *flow line* of a vector field $X \in \Gamma(TM)$ if we have that $\dot\gamma(t) = X_{\gamma(t)}$ for all $t \in I$. We say γ is a *maximal integral curve* if it is not possible to extend it to a larger open interval, i.e. there is no open interval $J \supset I$ and integral curve $\beta : J \to M$ of X so that $\beta|_I = \gamma$.

Given a smooth vector field $X \in \Gamma(TM)$, then for any $p \in M$ there is an open interval $I_p \subset \mathbb{R}$ containing the origin and an integral curve $\gamma_p : I_p \to M$ of X with $\gamma_p(0) = p$. Moreover if γ_p is maximal then it is unique. This can be shown by looking at a chart (U, u) containing p, in the chart the statement $\dot\gamma(t) = X_{\gamma(t)}$ becomes a system of ordinary differential equations with initial condition $\gamma_p(0) = p$. Since X is smooth the existence of a unique local solution that depends smoothly on the initial condition follows from the *Picard-Lindelöf theorem*. A curve found like that can be extended to another overlapping chart by taking one of the points of the curve in the intersection as the new initial condition of the ODE. Iterating like that we end up with a unique maximal curve, either because we end up with $I_p = \mathbb{R}$ or the curve leaving the manifold (if that is possible). Details and proof of this statement are found in LEE (2012, Thm. 9.12).



Smooth vector fields admit (maximal) integral curves.

FLOW OPERATOR. Let $X \in \Gamma(TM)$, then we define the *flow* of X as

$$\mathrm{Fl}_t^X(p) := \mathrm{Fl}^X(t,p) := \gamma_p(t), \tag{1.7}$$

where $\gamma_p : I_p \to M$ is the maximal integral curve of X starting at $p \in M$ from before. Note that the domain of Fl^X is not the full $\mathbb{R} \times M$ but the potentially smaller $\mathcal{D}(X) := \bigcup_{p\in M} I_p \times \{p\}$ where each I_p is the domain of the maximal integral curve of X started at p. If it happens that $\mathcal{D}(X) = \mathbb{R} \times M$ then we say Fl^X is a *global flow*, if not we say it is a *local flow*.

What these flows give us in particular is a way of moving tensors along a given vector field. Indeed since a Fl_t^X is a smooth mapping from M to M it follows that it has a smooth pushforward $(\mathrm{Fl}_t^X)_* : TM \to TM$ and pullback $(\mathrm{Fl}_t^X)^* : T^*M \to T^*M$. In particular $(\mathrm{Fl}_t^X)_*|_p : T_pM \to T_qM$ and $(\mathrm{Fl}_t^X)^*|_p : T_q^*M \to T_p^*M$ where $q = \mathrm{Fl}^X(t,p)$.

LIE DERIVATIVE. Using flows we can give meaning to difference quotients of vector and covector and higher order covariant or contravariant tensors. Let $X \in \Gamma(TM)$, for scalar functions $f \in C^\infty(M)$ we define the Lie derivative $\mathcal{L}_X : C^\infty(M) \to C^\infty(M)$ as

$$\mathcal{L}_X f := \left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_0 (\mathrm{Fl}_t^X)^* f = Xf,$$

which is identical to the familiar way of using a vector field as a derivation. Going further: for a $Y \in \Gamma(TM)$ we can define the Lie derivative $\mathcal{L}_X : \Gamma(TM) \to \Gamma(TM)$ as

$$\mathcal{L}_X Y := \left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_0 (\mathrm{Fl}_t^X)^* Y.$$

We can perform the pullback on a vector field here since Fl_t^X is a local diffeomorphism for each t. The same works for a covector field $\omega \in \Gamma(T^*M)$ as in

$$\mathcal{L}_X \omega := \left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_0 (\mathrm{Fl}_t^X)^* \omega,$$

or higher order contravariant or covariant tensors.

1.7 RIEMANNIAN METRIC

In Euclidean spaces we have a straightforward way of measuring distances but a manifold requires extra structure before we can talk about distances. A *Riemannian metric* on a smooth manifold M is a symmetric positive definite section of the $(0,2)$-tensor bundle, i.e. $\mathcal{G} \in \Gamma(T^*M \otimes T^*M)$ so that $\mathcal{G}_p(X_p, X_p) \geq 0$ for all $X_p \in T_pM$ and equality only holds if the vector is 0 additionally we have $\mathcal{G}_p(X_p, Y_p) = \mathcal{G}_p(Y_p, X_p)$ for all $X_p, Y_p \in T_pM$. To emphasize how a Riemannian metric is an inner product on each tangent space we also use the inner product notation $(X_p, Y_p)_{\mathcal{G}} := \mathcal{G}_p(X_p, Y_p)$, where we may drop the subscripted $\mathcal{G}$ if it is clear from context which metric we are using.

The Riemannian metric is usually denoted by g, but we reserve g for group elements.

Technically $\mathcal{G}$ is a $(0,2)$ tensor field and not a 'metric'. It is however convention to also call $\mathcal{G}$ a metric since it does directly induce an actual metric.

Since we require $\mathcal{G}$ to be a smooth section we have that $(X, Y)_{\mathcal{G}} : M \to \mathbb{R}$ is a smooth function for all smooth vector fields $X, Y \in \Gamma(TM)$. A manifold equipped with a Riemannian metric is called a *Riemannian manifold*.

After a choice of chart (U, u) we can express the Riemannian metric in terms of the coordinate cotangent basis at $p \in U$ as follows for some $X_p, Y_p \in T_pM$:

$$\begin{aligned}\left(X_p, Y_p\right)_{\mathcal{G}} &= \left(\sum_{i=1}^{n} X_p^i \left(\partial_{u^i}\right)_p, \sum_{j=1}^{n} Y_p^j \left(\partial_{u^j}\right)_p\right)_{\mathcal{G}} \\ &= \sum_{i,j=1}^{n} X_p^i Y_p^j \left(\left(\partial_{u^i}\right)_p, \left(\partial_{u^j}\right)_p\right)_{\mathcal{G}},\end{aligned}$$

where we let $X_p^i = (\mathrm{d}u^i)_p(X_p)$ and $Y_p^j = (\mathrm{d}u^j)_p(X_p)$ be the components of the tangent vectors in the coordinate basis of the chart. When we define $\mathcal{G}_{ij} \in C^\infty(U)$ as

$$\mathcal{G}_{ij}(p) := \left(\left(\partial_{u^i}\right)_p, \left(\partial_{u^j}\right)_p\right)_{\mathcal{G}}$$

for each $i, j = 1 \ldots n$ then we can write the metric on U as

$$\mathcal{G}|_U = \sum_{i,j=1}^{n} \mathcal{G}_{ij}\, \mathrm{d}u^i \otimes \mathrm{d}u^j.$$

We call the functions $\mathcal{G}_{ij} : U \to \mathbb{R}$ the components of the metric in the chart (U, u). It is often convenient to organize the component

functions of a metric in a matrix (the so-called *Gram matrix*) which we denote with

$$[\mathcal{G}_{ij}] := \begin{bmatrix} \mathcal{G}_{11} & \cdots & \mathcal{G}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{G}_{n1} & \cdots & \mathcal{G}_{nn} \end{bmatrix}.$$

Note that by construction for each $p \in U$, the matrix $[\mathcal{G}_{ij}(p)]$ is symmetric and positive definite.

LENGTH OF CURVES & DISTANCE. With a Riemannian metric $\mathcal{G}$ in hand we can define the length of a curve. Let $\gamma \in C^\infty(I, M)$ where I is an interval in $\mathbb{R}$ then we define the length of γ as

$$\operatorname{Len}_{\mathcal{G}}(\gamma) := \int_I \|\dot\gamma(t)\|_{\mathcal{G}} \, \mathrm{d}t$$

where $\|\dot\gamma(t)\|_{\mathcal{G}} := \sqrt{(\dot\gamma(t), \dot\gamma(t))_{\mathcal{G}}}$ and $\dot\gamma(t) \in T_{\gamma(t)}M$ so that for all $f \in C^\infty(M)$

$$\dot\gamma(t) f = \lim_{h \to 0} \frac{f(\gamma(t+h)) - f(\gamma(t))}{h}.$$

It can easily be verified that this definition is independent of the choice of parameterization of the curve. Using this definition of length of curves we can define the distance between two point of a Riemannian manifold as the infinum of the length of all curves between those points:

$$d_{\mathcal{G}}(p, q) := \inf \left\{ \operatorname{Len}_{\mathcal{G}}(\gamma) \;\middle|\; \gamma \in \mathrm{PC}([0,1], M),\ \gamma(0) = p,\ \gamma(1) = q \right\}, \tag{1.8}$$

where $\mathrm{PC}([0,1], M)$ are the piecewise smooth functions from $[0,1]$ to M. This distance $d_{\mathcal{G}}$ is a *metric* in the formal sense and so is symmetric, reflexive and satisfies the triangle inequality, consequently $(M, d_{\mathcal{G}})$ forms a metric space.

CUT LOCUS. As metrics and their smoothness play a role in our later work, we need to take into account where that smoothness fails. For any give $p \in M$ the *cut locus* $\operatorname{cut}(p) \subset M$ is the set of points where the distance map $\mathrm{d}_{\mathcal{G}}(p, \cdot)$ is not smooth (excluding p itself). As long as we stay away from the cut locus, the infinum from (1.8) gives a unique geodesic.

VOLUME. Besides a notion of length, the Riemannian metric also gives us a way of measuring volume, exactly how it does this depends on whether the manifold is orientable or not. If a n-dimensional manifold M admits a smooth non-vanishing n-form (nowhere identical to the zero mapping) we say the manifold is *orientable*, if no such form exists we say the manifold is not orientable.

If the manifold is orientable, we can choose a smooth non-vanishing n-form to establish the orientation of a given (local) basis of the tangent bundle. Let $\mu \in \Omega^n(M)$ be a smooth non-vanishing n-form and let $(E_{1,p}, \ldots, E_{n,p})$ be a basis of T_pM then we say $(E_{1,p}, \ldots, E_{n,p})$ is *positively oriented* if

$$\mu(E_{1,p}, \ldots, E_{n,p}) > 0$$

and is *negatively oriented* if

$$\mu(E_{1,p}, \ldots, E_{n,p}) < 0.$$

Since μ is non-vanishing every basis at every point is either positively or negatively oriented. After we have chosen in this manner which bases are positively oriented we say we have an *oriented manifold*.

An oriented Riemannian manifold $(M, \mathcal{G})$ admits a unique n-form $\mathrm{vol}_M \in \Omega^n(M)$ so that if $(E_1, \ldots, E_n)$ is a local positively oriented orthonormal frame then

$$\mathrm{vol}_{\mathcal{G}}(E_1, \ldots, E_n) = 1,$$

or alternatively, if $(\varepsilon^1, \ldots, \varepsilon^n)$ is local positively oriented orthonormal coframe then

$$\mathrm{vol}_{\mathcal{G}} = \varepsilon^1 \wedge \cdots \wedge \varepsilon^n,$$

we call vol_M the *volume form*. If (U, u) is a chart of M then the volume form is given by

$$\mathrm{vol}_{\mathcal{G}} = \sqrt{\det[\mathcal{G}_{ij}]}\, \mathrm{d}u^1 \wedge \cdots \wedge \mathrm{d}u^n.$$

In the case that the manifold is not orientable we cannot produce a signed volume form but we can still define an unsigned density that works similarly. Indeed let $(\varepsilon^1, \ldots, \varepsilon^n)$ be a local orthonormal

coframe then we call the smooth pseudo-form $\overline{\mathrm{vol}}_g = |\varepsilon^1 \wedge \cdots \wedge \varepsilon^n|$ the *Riemannian density*. It is easy to check that $\overline{\mathrm{vol}}_g(E_1, \ldots, E_n) = 1$ for any orthonormal frame $(E_1, \ldots, E_n)$ not just those that are positively oriented. In the case that the manifold is oriented we have $\overline{\mathrm{vol}}_g = |\mathrm{vol}_g|$, but the density has the advantage of being well defined either way.

Both the volume form and Riemannian density can be integrated over, in the first case as an integral over n-forms, in the second case as an integral over a density. For the purpose of integration of scalar functions these integrals agree, i.e. $\int_M f \,\mathrm{vol}_g = \int_M f \,\overline{\mathrm{vol}}_g$ for any appropriate $f : M \to \mathbb{R}$. For that reason the same vol_g symbol is often used for both the volume form and density.

For a nice overview of the different types of integration see Tao (2010).

1.8 LINEAR CONNECTIONS & THE LEVI-CEVITA CONNECTION

We know how to differentiate scalar functions along vector fields on a manifold, but we also want to differentiate vector fields (and other tensor fields) as we can do in Euclidean spaces per

$$\nabla_{X_p} Y = \lim_{t \to 0} \frac{Y_{p+tX_p} - Y_p}{t}$$

where $Y \in \Gamma(T\mathbb{R}^n)$ and $X_p \in T_p\mathbb{R}^n \simeq \mathbb{R}^n$. This construction does not translate to manifolds since there is no canonical mechanism to subtract vectors living in two different tangent spaces, i.e. when $X_p \in T_pM$ and $X_q \in T_qM$ then we do not know what $X_p - X_q$ means. To be able to differentiate vector fields and other types of tensor fields we need an additional structure on our manifold that allows us to *connect* tangent spaces in the same way as the above difference quotient does.

A *linear connection* on a manifold M is a map

$$\nabla : \Gamma(TM) \times \Gamma(TM) \to \Gamma(TM),$$

which satisfies

A linear connection is also known as a Koszul connection.

1. $\nabla_{f_1X_1 + f_2X_2}(Y) = f_1 \nabla_{X_1} Y + f_2 \nabla_{X_2}$ (total linearity in the first argument),
2. $\nabla_X(fY) = (Xf)Y + f\nabla_X Y$ (Leibniz rule),

3. $\nabla_X(\lambda_1 Y_1 + \lambda_2 Y_2) = \lambda_1 \nabla_X Y_1 + \lambda_2 \nabla_X Y_2$ (scalar linearity in the second argument),

for all $\lambda_2, \lambda_2 \in \mathbb{R}$, $f, f_1, f_2 \in C^1(M)$ and $X, X_1, X_2, Y, Y_1, Y_2 \in \Gamma(M)$. Pay attention to the fact that the pointwise evaluation of the resulting vector field happens in the first argument: $(\nabla_X Y)_p = \nabla_{X_p} Y$. Only having an isolated tangent vector in the second argument (Y above) makes no sense since we want to quantify exactly how Y changes in the direction of X_p at the point $p \in M$.

A linear connection is essentially the same thing as a *covariant derivative* which is a $\mathbb{R}$-linear map $\nabla : \Gamma(TM) \to \Gamma(T^*M \otimes TM)$ that satisfies the product rule

$$\nabla(fX) = \mathrm{d}f \otimes X + f\nabla X$$

for all $f \in C^\infty(M)$ and $X \in \Gamma(TM)$. It is apparent that this is the same definition as that of the linear connection but with currying the first argument.

When we have a curve $\gamma \in C^\infty(I, M)$ for some interval $I \subset \mathbb{R}$ and a vector field $X \in \Gamma(TM)$ we say X is *parallel along* γ if $\nabla_{\dot\gamma(t)} X = 0$ for all $t \in I$ (or $\nabla_{\dot\gamma} X = 0$ for short). When we have a vector field X that satisfies $X_{\gamma(t)} = \dot\gamma(t)$ for all $t \in I$ then we say γ is an *autoparallel curve* with respect to ∇ if $\nabla_{\dot\gamma}\dot\gamma \equiv \nabla_{\dot\gamma} X = 0$.

TENSOR CONNECTIONS. A linear connection implicitly defines a *dual connection* $\nabla^* : \Gamma(TM) \times \Gamma(T^*M) \to \Gamma(T^*M)$ by

$$\mathrm{d}(\omega(Y))(X) = (\nabla^*_X \omega)(Y) + \omega(\nabla_X Y)$$

for each $X, Y \in \Gamma(TM)$ and $\omega \in \Gamma(T^*M)$.

From the linear connection and its dual we then also get connections for higher order tensor bundles by recursively applying the following *tensor product connection*. Let $\nabla^{(1)}$ and $\nabla^{(2)}$ be connections on some ranks of tensor bundles of M and s_1 and s_2 smooth section of those tensor bundles, then the tensor product connections $\nabla^{(1)} \otimes \nabla^{(2)}$ is defined as

$$\left(\nabla^{(1)} \otimes \nabla^{(2)}\right)_X (s_1 \otimes s_2) := \nabla^{(1)}_X s_1 \otimes s_2 + s_1 \otimes \nabla^{(2)}_X s_2$$

for each $X \in \Gamma(TM)$.

From now on we will use the same symbol to denote the initial linear connection on the tangent bundle and its derived connections on the cotangent and higher order bundles.

CHRISTOFFEL SYMBOLS. Given a chart (U, u) of M we see how a linear connection can be expressed in terms of the coordinate basis of the chart. Let $X, Y \in \Gamma(TM)$ so that $X = \sum_{i=1}^n X^i \partial_{u^i}$ and $Y = \sum_{j=1}^n Y^j \partial_{u^j}$ on U, then

$$\begin{aligned}\nabla_X Y &= \nabla_{\sum_{i=1}^n X^i \partial_{u^i}} \left(\sum_{j=1}^n Y^j \partial_{u^j} \right) \\ &= \sum_{i,j=1}^n X^i \nabla_{\partial_{u^i}} \left(Y^j \partial_{u^j} \right) \\ &= \sum_{j=1}^n X Y^j \partial_{u^j} + \sum_{i,j=1}^n X^i Y^j \nabla_{\partial_{u^i}} \partial_{u^j}.\end{aligned}$$

Now $\nabla_{\partial_{u^i}} \partial_{u^j} \in \Gamma(TU)$ for $i, j = 1 \dots n$ so these vector fields can again be expressed in the coordinate basis of (U, u):

$$\nabla_{\partial_{u^i}} \partial_{u^j} = \sum_{k=1}^n \Gamma_{ij}^k \partial_{u^k}$$

for some $\Gamma_{ij}^k \in C^\infty(U)$. We call these local smooth functions Γ_{ij}^k the *Christoffel symbols* of ∇ on the chart (U, u). The expression for the linear connection in the coordinate basis can then be cleaned up to

$$\nabla_X Y = \sum_{k=1} \left(X Y^k + \sum_{i,j=1}^n \Gamma_{ij}^k X^i Y^j \right) \partial_{u^k}.$$

PARALLEL TRANSPORT. A linear connection connects tangent spaces in a more literal sense by giving a way of transporting tangent vectors from one tangent space to another. Let $p, q \in M$ and let $\gamma \in C^\infty([0,1], M)$ be an embedding so that $\gamma(0) = p$ and $\gamma(1) = q$. Now define the *parallel transport* operator $\mathcal{P}_\gamma : T_pM \to \Gamma(T\gamma)$ so that it is linear and satisfies

We abbreviate $T\gamma([0,1])$, i.e. the tangent bundle along the curve γ to $T\gamma$.

$$\begin{aligned}&\left(\mathcal{P}_\gamma X_p\right)_p = X_p \text{ and} \\ &\nabla_{\dot\gamma(t)} \mathcal{P}_\gamma X_p = 0\end{aligned}$$

for all $X_p \in T_pM$ and $t \in [0,1]$.

We can see that $t \mapsto \left(\mathcal{P}_\gamma X_p\right)_{\gamma(t)}$ exists and is unique by looking at it in a chart. Without loss of generality consider a chart (U, u) of M that fully contains γ then $\mathcal{P}_\gamma$ is determined by its effect on the coordinate basis:

$$\left(\mathcal{P}_\gamma \left(\partial_{u^i}\right)_p\right)_{\gamma(t)} = \sum_{j=1}^{n} P_i^j(t) \left(\partial_{u^j}\right)_{\gamma(t)}$$

where $P_i^j : [0,1] \to \mathbb{R}$ for $i, j = 1 \ldots n$. The two defining conditions of parallel transport give use an initial value problem for the functions P_i^j. The first condition translates to the initial conditions $P_i^j(0) = \delta_i^j$. The second condition $\nabla_{\dot\gamma(t)} \mathcal{P}_\gamma X_p = 0$ translates to the following system of equations:

$$\dot P_i^j(t) = -\sum_{k,\ell=1}^{n} P_i^k(t)\, \dot\gamma^\ell(t)\, \Gamma_{\ell k}^j(\gamma(t))$$

for $i, j = 1 \ldots n$. Applying the *Picard-Lindelöf theorem* gives us the existence, uniqueness of the solution to this system of ordinary differential equations, additionally the solution is smooth since the Christoffel symbols and the curve coordinates are smooth. Since the points p, q and the curve γ were chosen arbitrarily we can now have a general method of transporting tangent vectors along the manifold. Note that the parallel transport of the same tangent vector between the same two points can still differ depending on the choice of curve between those points.

Conversely, we can recover the linear connection back from the parallel transport map. Say we have a basis $\{E_{i,p}\}_{i=1}^n$ of T_pM then parallel transport gives us a basis $\{\left(\mathcal{P}_\gamma E_{i,p}\right)_{\gamma(t)}\}_{i=1}^n$ of $T_{\gamma(t)}M$ for all $t \in [0,1]$. So, at least along the curve, we can express the vector field Y as $Y_{\gamma(t)} = \sum_{i=1}^n Y^i(\gamma(t)) \left(\mathcal{P}_\gamma E_{i,p}\right)_{\gamma(t)}$ for some appropriate $Y^i \in C^\infty(U)$. Let $X_p = \dot\gamma(0)$ then we find

$$\nabla_{X_p} Y = \sum_{i=1}^{n} \left(X_p Y^i\right) E_{i,p} + Y^i(p) \nabla_{X_p} \left(\mathcal{P}_\gamma E_{i,p}\right)$$

but by definition $\nabla_{X_p} \mathcal{P}_\gamma E_{i,p} = 0$ so we are left with

$$\nabla_{X_p} Y = \sum_{i=1}^{n} X_p Y^i E_{i,p}$$

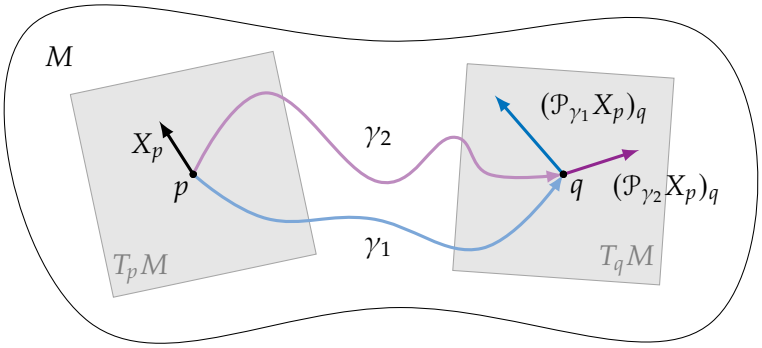


FIGURE 1.3: How parallel transport moves a tangent vector from one point of the manifold to another depends on the curve along which we transport the tangent vector.

as the expression for the connection in this frame.

We will compare this expression to the one we get by differentiating using parallel transport. If $\mathcal{P}_\gamma : T_pM \to \Gamma(T\gamma)$ is as before we can extend its domain to all tangent spaces along the curve γ instead of just T_pM by specifying that

$$\mathcal{P}_\gamma \left(\mathcal{P}_\gamma X_p\right)_{\gamma(t)} = \mathcal{P}_\gamma X_p$$

for all $X_p \in T_pM$ and $t \in [0,1]$. In effect we can recover the whole vector field $\mathcal{P}_\gamma X_p$ from any of its individual tangent vectors $\mathcal{P}_\gamma \left(\mathcal{P}_\gamma X_p\right)_{\gamma(t)}$ not just the initial X_p. In particular this allows us to transport any tangent vector along the curve back to its starting point p since

$$\left(\mathcal{P}_\gamma Y_{\gamma(t)}\right)_p \in T_pM$$

for any $Y_{\gamma(t)} \in T_{\gamma(t)}M$.

With parallel transport we can now give meaning to taking the derivative of a vector field:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathcal{P}_\gamma Y_{\gamma(t)}\right)_p\Big|_{t=0} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathcal{P}_\gamma \sum_{i=1}^n Y^i(\gamma(t))\left(\mathcal{P}_\gamma E_{i,p}\right)_{\gamma(t)}\right)_p\Bigg|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=1}^n Y^i(\gamma(t))\left(\mathcal{P}_\gamma\left(\mathcal{P}_\gamma E_{i,p}\right)_{\gamma(t)}\right)_p\Bigg|_{t=0} \\
&= \sum_{i=1}^n \frac{\mathrm{d}}{\mathrm{d}t} Y^i(\gamma(t))\Big|_{t=0} E_{i,p} \\
&= \sum_{i=1}^n X_p Y^i\, E_{i,p},
\end{aligned}$$

which is identical to the expression we found for $\nabla_{X_p} Y$. We can conclude that linear connections and a parallel transport maps are interchangeable.

LEVI-CEVITA CONNECTION. In principle, we can in any chart (U, u) select some Christoffel symbols $\Gamma_{ij}^k \in C^\infty(U)$ and this (locally) defines a connection as long as $\Gamma_{ij}^k = \Gamma_{ji}^k$. In practice we will want to choose the connection so that it matches some other structure of the manifold, in the case of a Riemannian manifold we can first of all choose the connection to be compatible with the metric.

A connection ∇ is *metric compatible* with a Riemannian metric $\mathcal{G}$ if $\nabla\mathcal{G} = 0$, that is

$$X(\mathcal{G}(Y, Z)) = \mathcal{G}(\nabla_X Y, Z) + \mathcal{G}(Y, \nabla_X Z), \tag{1.9}$$

for all $X, Y, Z \in \Gamma(TM)$. This property can be more easily understood in terms of parallel transport: a connection is metric compatible if its associated transport maps are isometries. Meaning that for any $\gamma \in C^\infty([0,1], M)$ with $\gamma(0) = p$ we have

$$\left((\mathcal{P}_\gamma X_p)_{\gamma(t)}, (\mathcal{P}_\gamma Y_p)_{\gamma(t)}\right)_{\mathcal{G}} = \left(X_p, Y_p\right)_{\mathcal{G}},$$

for all $t \in [0,1]$ and all $X_p, Y_p \in T_pM$. A corollary to this is that orthonormal frames are always transported to orthonormal frames.

Metric compatibility does not yet uniquely determine a connection; an additional property we can impose is that of being *torsion free*. We say a connection ∇ is torsion free if

$$\nabla_X Y - \nabla_Y X = [X, Y] := X \circ Y - Y \circ X \tag{1.10}$$

for all $X, Y \in \Gamma(TM)$ seen as linear $C^\infty(M) \to C^\infty(M)$ maps. Essentially we want the non-commutativity of the connection to match the non-commutativity of the vector fields of the manifold. The non-commutativity of applying vector fields as expressed by the Lie bracket is intrinsic to the smooth structure of the manifold, we would like the connection we impose on top of the manifold to respect this pre-existing structure.

Together the metric compatibility (1.9) and torsion freeness (1.10) requirements determine a unique connection called the *Levi-Cevita connection*, which we will denote by $\nabla^{\mathcal{G}}$.

In a chart (U, u), the Christoffel symbols of the Levi-Cevita connection are given in term of the metric tensor components per

$$\Gamma^k_{ij} = \frac{1}{2} \sum_{\ell=1}^{n} \left(\partial_{u^j} \mathcal{G}_{\ell i} + \partial_{u^i} \mathcal{G}_{\ell j} - \partial_{u^\ell} \mathcal{G}_{ij} \right) \mathcal{G}^{k\ell} \tag{1.11}$$

where $\mathcal{G}^{kl} \in C^\infty(U)$ are the components of the dual of $\mathcal{G}$, i.e.

$$\sum_{k=1}^{n} \mathcal{G}^{ik} \mathcal{G}_{kj} = \delta^i_j \tag{1.12}$$

for all $i, j, k = 1 \dots n$.

1.9 GRADIENT, HESSIAN & LAPLACIAN

With the Riemannian metric and Levi-Cevita connection in hand we can now generalize many differential operators that are useful for constructing partial differential equations to Riemannian manifolds.

GRADIENT. The Riemannian metric being an inner product field gives us a canonical isomorphism between the tangent and cotangent bundle given by $X_p \mapsto (X_p, \cdot)_{\mathcal{G}} \in T^*_p M$ for each $X_p \in T_p M$.

The *gradient* of a function $f \in C^\infty(M)$ is then the unique vector field $\operatorname{grad} f \in \Gamma(TM)$ so that

$$\big(\operatorname{grad} f, \cdot\big)_{\mathcal{G}} = \mathrm{d}f. \tag{1.13}$$

The gradient is essentially the Riesz representative of $\mathrm{d}f_p$ relative to the inner product $\mathcal{G}_p$.

We can examine what this expression evaluates to in a chart. Let (U, u) be some chart of M, then we can write any $X \in \Gamma(TU)$ using the coordinate basis vectors as

$$X = \sum_{i=1}^{n} X^i \partial_{u^i}.$$

for some $X^i \in C^\infty(U)$. Evaluating (1.13) with this vector field gives $(\operatorname{grad} f, X)_{\mathcal{G}} = \mathrm{d}f(X)$ which expands to

$$\left(\sum_{i=1}^{n} \mathrm{d}u^i(\operatorname{grad} f)\, \partial_{u^i}, \sum_{j=1}^{n} X^j \partial_{u^j} \right)_{\mathcal{G}} = \sum_{i=1}^{n} X^i \frac{\partial f}{\partial u^i},$$

the left side of which becomes

$$\sum_{i,j=1}^{n} \mathrm{d}u^i(\operatorname{grad} f) X^j \, (\partial_{u^i}, \partial_{u^j})_{\mathcal{G}} = \sum_{i,j=1}^{n} \mathcal{G}_{ij} (\operatorname{grad} f)^i X^j,$$

When opportune we will contract the notation for $\partial_{u^i} f$ to the usual $\frac{\partial f}{\partial u^i}$.

where we abbreviate $(\operatorname{grad} f)^i = \mathrm{d}u^i(\operatorname{grad} f) \in C^\infty(U)$. We can better see what this means for the previous equality in matrix notation:

$$\begin{bmatrix} X^1 & \cdots & X^n \end{bmatrix} \begin{bmatrix} \mathcal{G}_{11} & \dots & \mathcal{G}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{G}_{n1} & \dots & \mathcal{G}_{nn} \end{bmatrix} \begin{bmatrix} (\operatorname{grad} f)^1 \\ \vdots \\ (\operatorname{grad} f)^n \end{bmatrix} = \begin{bmatrix} X^1 & \dots & X^n \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial u^1} \\ \vdots \\ \frac{\partial f}{\partial u^n} \end{bmatrix}.$$

This can only hold for all X if

$$\begin{bmatrix} \mathcal{G}_{11} & \dots & \mathcal{G}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{G}_{n1} & \dots & \mathcal{G}_{nn} \end{bmatrix} \begin{bmatrix} (\operatorname{grad} f)^1 \\ \vdots \\ (\operatorname{grad} f)^n \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial u^1} \\ \vdots \\ \frac{\partial f}{\partial u^n} \end{bmatrix},$$

from which we can solve for the components of $\operatorname{grad} f$ by inverting the matrix with the components of the metric tensor:

$$\begin{bmatrix} (\operatorname{grad} f)^1 \\ \vdots \\ (\operatorname{grad} f)^n \end{bmatrix} = \begin{bmatrix} \mathcal{G}_{11} & \dots & \mathcal{G}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{G}_{n1} & \dots & \mathcal{G}_{nn} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial f}{\partial u^1} \\ \vdots \\ \frac{\partial f}{\partial u^n} \end{bmatrix}.$$

Since $\mathcal{G}$ is positive definite everywhere its associated matrices are always invertible. We will denote the components of these inverted matrices by moving the indices from subscript to superscript, i.e. we define

$$\begin{bmatrix} \mathcal{G}^{11} & \dots & \mathcal{G}^{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{G}^{n1} & \dots & \mathcal{G}^{nn} \end{bmatrix} := \begin{bmatrix} \mathcal{G}_{11} & \dots & \mathcal{G}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{G}_{n1} & \dots & \mathcal{G}_{nn} \end{bmatrix}^{-1},$$

as in (1.12).

We can now write the gradient of a function f in the local coordinate basis as

$$\operatorname{grad} f|_U = \sum_{i,j=1}^{n} \mathcal{G}^{ij} \frac{\partial f}{\partial u^i} \partial_{u^j} \tag{1.14}$$

It is common to overload to the connection symbol ∇ to indicate the gradient when applied to a scalar function, i.e. if $f \in C^\infty(M)$ then $\nabla f \equiv \operatorname{grad} f$. In cases where the metric being used is ambiguous we can include it in the notation as $\nabla_{\mathcal{G}} \equiv \operatorname{grad}$ with respect to a Riemannian metric $\mathcal{G}$.

HESSIAN. On a manifold M with a Levi-Cevita connection $\nabla^{\mathcal{G}}$, we define the *Hessian* of a smooth function $f \in C^\infty(M)$ as the tensor field $\operatorname{Hess} f \in \Gamma(T^*M \otimes T^*M)$ so that

$$\operatorname{Hess} f(X, Y) := (\nabla^{\mathcal{G}}_X \, \mathrm{d}f)(Y) = X(Yf) - (\nabla^{\mathcal{G}}_X Y) f \tag{1.15}$$

for any $X, Y \in \Gamma(TM)$. Alternatively the Hessian with respect to the Levi-Cevita connection can be expressed as $\operatorname{Hess} f(X, Y) = \left(\nabla^{\mathcal{G}}_X \operatorname{grad} f, Y\right)_{\mathcal{G}}$.

On a Riemannian manifold the *Hessian generally means the one with respect to the Levi-Cevita connection as we formulate here, but the Hessian exists for any connection and in our work we will find use for Hessians with respect to other connections.*

In a coordinate chart (U, u) the Hessian can be expressed with the Christoffel symbols as

$$\begin{aligned} \operatorname{Hess} f &= \sum_{i,j=1}^{n} \left(\nabla^{\mathcal{G}}_{\partial_{u^i}} \mathrm{d}f\right)(\partial_{u^j}) \, \mathrm{d}u^i \otimes \mathrm{d}u^j \\ &= \sum_{i,j=1}^{n} \left(\frac{\partial^2 f}{\partial u^i \, \partial u^j} - \sum_{k=1}^{n} \Gamma^k_{ij} \frac{\partial f}{\partial u^k} \right) \mathrm{d}u^i \otimes \mathrm{d}u^j. \end{aligned} \tag{1.16}$$

LAPLACIAN. On a Riemannian manifold $(M, \mathcal{G})$ with Levi-Cevita connection $\nabla^{\mathcal{G}}$ the *Laplace-Beltrami operator*, or *Laplacian* for short, of a smooth function $f \in C^\infty(M)$ is a denoted as $\Delta f \in C^\infty(M)$ and given by the trace of the Hessian with respect to the Levi-Cevita connection:

$$\Delta f := \operatorname{trace}(\operatorname{Hess} f). \tag{1.17}$$

In a chart (U, u) this becomes

$$\Delta f = \sum_{i=1}^{n} \operatorname{Hess} f\,(\partial_{u^i}, \partial_{u^j}) = \sum_{i=1}^{n} \left(\frac{\partial^2 f}{\partial u^i\, \partial u^j} - \sum_{k=1}^{1} \Gamma_{ii}^k \frac{\partial f}{\partial u^k} \right),$$

where we recognize the familiar sum of second order derivatives but with some first order compensatory terms that account for the curvature of the manifold. In terms of the components of the Riemannian metric $\mathcal{G}$ the Laplacian can also be written as

$$\Delta f = \frac{1}{\sqrt{\det[\mathcal{G}_{ij}]}} \sum_{i,j=1}^{n} \partial_{u^i} \left(\mathcal{G}^{ij} \sqrt{\det[\mathcal{G}_{ij}]} \frac{\partial f}{\partial u^j} \right).$$

This concludes our recapitulation of the for us relevant topics in Riemannian geometry.

2

LIE GROUPS & HOMOGENEOUS SPACES

In the context of both deep learning and image processing we want to talk about symmetries, specifically how some transformation of the input of a system is reflected in its output. In our work we consider continuous transformations of data that are given by a Lie group. In this chapter we summarize what Lie groups are and how they interact with homogeneous spaces. The main references for this chapter are Lee (2012, Ch. 7, 20, 21) and Koda (2009).

2.1 LIE GROUPS

A *Lie group* G is a smooth manifold that is also an algebraic group so that its multiplication $\mu : G \times G \to G$ and inversion $\nu : G \to G$ are smooth. Multiplication of group elements will be denoted by juxtaposition as $g_1 g_2 \equiv \mu(g_1, g_2)$ and inversion $\nu : G \to G$ as $g^{-1} \equiv \nu(g)$. We will denote the unit element by $e \in G$ so that for all $g \in G$ we have $gg^{-1} = g^{-1}g = e$. Since it is a group, the inverse g^{-1} exists for all $g \in G$ and multiplication is associative, i.e. $(g_1 g_2)g_3 = g_1(g_2 g_3)$ for all $g_1, g_2, g_2 \in G$. Multiplication need not be commutative however, and in our applications it will generally not be.

TRANSLATIONS. To every element $g \in G$ there are associated two special maps: *left translation* $L_g : G \to G$ given by $L_g h := gh$ and *right translation* $R_g : G \to G$ given by $R_g h := hg$. These left and right translations are smooth group homomorphisms of the group to itself since they can be shown to be bijective and also smooth since $L_g = \mu(g, \cdot)$ and $R_g = \mu(\cdot, g)$ and μ is smooth. Their inverses are equally smooth since $L_g^{-1} = L_{g^{-1}}$ and $R_g^{-1} = R_{g^{-1}}$.

Furthermore, it is easy to check that $L_g \circ L_h = L_{gh}$ and $R_g \circ R_h = R_{hg}$.

INVARIANT VECTOR FIELDS. Being diffeomorphisms, the left and right translations induce pushforwards. Vector fields that stay invariant under these pushforwards are of special interest. We say a tangent vector field $X \in \Gamma(TM)$ is *left-invariant* if for all $g \in G$ we have $(L_g)_* X = X$. We denote the set of left-invariant vector fields as

$$\mathfrak{X}(G) := \left\{ X \in \Gamma(TG) \,\middle|\, \forall g \in G : (L_g)_* X = X \right\}. \tag{2.1}$$

Similarly, we say X is *right-invariant* if $(R_g)_* X = X$. We will be working only with left-invariant vector fields so we reserve no symbol for the set of right-invariant vector fields.

Left- and right-invariant vector fields are fully identified by their value in any individual tangent space. Indeed, if $X \in \Gamma(TM)$ is left-invariant then $X_g = ((L_h)_* X)_g = (L_h)_* X_{h^{-1}g}$ for all $g, h \in G$, hence it is sufficient to specify X in, say, $T_e G$ and then obtain $X_g = (L_g)_* X_e$ for all $g \in G$. Which means that $\mathfrak{X}(G)$ is isomorphic to $T_e G$ or any other individual tangent space. The same observation applies to right invariant vector fields.

A vector field being left or right invariant is equivalent to saying that it commutes with the left respectively right translations. Say $X \in \mathfrak{X}(G)$ then for all $f \in C^\infty(G)$ and $g, h \in G$ we have

$$\begin{aligned} &\left(\left(L_g\right)_* X \right)_h f = X_h f \\ &\iff X_{g^{-1}h}(f \circ L_g) = X_h f \\ &\iff \left(\left(X(f \circ L_g)\right) \circ L_{g^{-1}} \right)(h) = X_h f, \end{aligned}$$

and so

$$X(f \circ L_g) = (Xf) \circ L_g$$

for all $f \in C^\infty(G)$ and $g \in G$. If X was a right invariant vector field then we similarly have $X(f \circ R_g) = (Xf) \circ R_g$ for all $f \in C^\infty(G)$ and $g \in G$.

LIE ALGEBRA. Let G be a Lie group and let $X, Y \in \mathfrak{X}(G)$ be left-invariant vector fields, then the Lie bracket $[X, Y]$ also yields a

left-invariant vector field since for every $f \in C^\infty(G)$ and $g \in G$ we have

$$\begin{aligned}[X,Y](f \circ L_g) &= X\left(Y(f \circ L_g)\right) - Y\left(X(f \circ L_g)\right) \\ &= X\left((Yf) \circ L_g\right) - Y\left((Xf) \circ L_g\right) \\ &= (X(Yf)) \circ L_g - (Y(Xf)) \circ L_g \\ &= ([X,Y]f) \circ L_g\end{aligned}$$

and so $[X,Y] \in \mathfrak{X}(G)$. This turns the set $\mathfrak{X}(G)$ of left-invariant vector fields into a Lie algebra, i.e. a vector space equipped with a Lie bracket $[\,\cdot\,,\,\cdot\,]$ that satisfies the properties we detailed in Section 1.3. Since left-invariant vector fields are fully determined by their value in T_eG we can also identify $\mathfrak{X}(G) \equiv T_eG$ and see T_eG as a Lie algebra with the Lie bracket on T_eG being inherited from the Lie bracket (i.e. commutator) of the corresponding left-invariant vector fields as

$$[X_e, Y_e] := \left[g \mapsto (L_g)_* X_e, g \mapsto (L_g)_* Y_e\right]_e \in T_eG \tag{2.2}$$

for all $X_e, Y_e \in T_eG$. The tangent space at the unit element together with the inherited Lie bracket (2.2) is usually called *the* Lie algebra of G and denoted by $\mathfrak{g} := (T_eG, [\,\cdot\,,\,\cdot\,])$. Given that the Lie bracket of $\mathfrak{g}$ comes from the commutator of $\mathfrak{X}(G)$ it is natural that $\mathfrak{g}$ and $(\mathfrak{X}(G), [\,\cdot\,,\,\cdot\,])$ are also isomorphic as Lie algebras.

STRUCTURE CONSTANTS. Suppose we have a basis $(E_i)_{i=1}^n$ of $\mathfrak{X}(G)$, then we can express $X, Y \in \mathfrak{X}(G)$ as $X = \sum_{i=1}^n X^i E_i$ and $Y = \sum_{i=1}^n Y^i E_i$ for some $X^i, Y^i \in \mathbb{R}$ for all $i = 1 \dots n$. Then by linearity of the Lie bracket we get

$$[X,Y] = \left[\sum_{i=1}^n X^i E_i, \sum_{j=1}^n Y^j E_j\right] = \sum_{i,j=1}^n X^i Y^i [E_i, E_j].$$

Since we know the Lie bracket of two left-invariant vector fields yields another left-invariant vector field we can re-express the Lie bracket of the basis vectors in terms of the same basis, i.e. there exist constants $c_{ij}^k \in \mathbb{R}$ for all $i, j, k = 1 \dots n$ so that

$$[E_i, E_j] = \sum_{k=1}^k c_{ij}^k E_k, \tag{2.3}$$

the coefficients c_{ij}^k are called the *structure constants* of G with respect to the basis $(E_i)_{i=1}^n$.

EXPONENTIAL MAPS . Let G be a Lie group, then a group homomorphism $\gamma : (\mathbb{R}, +) \to G$ is called a *one-parameter subgroup* or *exponential curve* of G. Being group homomorphisms, such curves satisfy $\gamma(0) = e$ and $\gamma(s + t) = \gamma(s)\gamma(t)$ for all $t, s \in \mathbb{R}$.

Left-invariant vector fields naturally have an exponential curve associated to them in the sense that if X is a left-invariant vector field and γ is an exponential curve with $\dot\gamma(0) = X_e$ then for all $t \in \mathbb{R}$ we have $\dot\gamma(t) = X_{\gamma(t)}$. Indeed exponential curves are exactly the flow lines of left-invariant vector fields through the identity. This can be verified as follows for any $f \in C^\infty(G)$:

$$\begin{aligned} X_{\gamma(t)} f &= \left(\left(L_{\gamma(t)}\right)_* X_e \right) f \\ &= X_e(f \circ L_{\gamma(t)}) \\ &= \dot\gamma(0)(f \circ L_{\gamma(t)}) \\ &= \left(\frac{\mathrm{d}}{\mathrm{d}s}\right)_0 (f \circ L_{\gamma(t)})(\gamma(s)) \\ &= \left(\frac{\mathrm{d}}{\mathrm{d}s}\right)_0 f(\gamma(t+s)) \\ &= \left(\frac{\mathrm{d}}{\mathrm{d}s}\right)_t (f \circ L_{\gamma(s)}) \\ &= \dot\gamma(t) f. \end{aligned}$$

Consequently, each exponential curve can be uniquely identified with its tangent vector at the identity $\dot\gamma(0)$ since each such tangent vector can be uniquely extended to a left-invariant vector field $g \mapsto (L_g)_* \dot\gamma(0)$ and this left-invariant vector field subsequently yields the original curve again as an integral curve. Using this we define the *exponential map* $\exp_G : \mathfrak{g} \to G$ as

$$\exp_G(X) := \gamma(1) \tag{2.4}$$

for every $X \in \mathfrak{g}$ where γ is the unique exponential curve with $\dot\gamma(0) = X_e$. This can be restated in terms of a flow (1.7) as

$$\exp_G(tX) = \mathrm{Fl}_t^X(e).$$

The exponential curve can easily be recovered as $t \mapsto \exp_G(tX)$, from which we can also see that $\exp_G((s+t)X) = \exp_G(sX)\exp_G(tX)$ and $\exp_G(X)^{-1} = \exp_G(-X)$.

From the fact that $\exp_G(X+Y) = \exp_G(Y+X)$ but in general we have $\exp_G(X)\exp_G(Y) \neq \exp_G(Y)\exp_G(X)$ we can surmise that in general $\exp_G(X+Y) \neq \exp_G(X)\exp_G(Y)$ unless $[X,Y] = 0$.

Finally, we observe that for any $g \in G$ and left-invariant vector field $X \in \mathfrak{g}$ left multiplication L_g must map integral curves of X to integral curves of X. Consequently, $t \mapsto L_g \exp_G(tX)$ is the integral curve of X started at g, which means it corresponds to the flow $t \mapsto \mathrm{Fl}_t^X(g)$. We conclude that the flow of X is given by right multiplication by $\exp_G(tX)$ since $\mathrm{Fl}_t^X(g) = L_g \exp_G(tX) = g \exp_G(tX) = R_{\exp_G(tX)}g$.

LOGARITHMIC MAP The exponential map is not necessarily injective nor surjective but it can be restricted to a diffeomorphism from a neighbourhood of $0 \in \mathfrak{g}$ to a neighbourhood of $e \in G$ (LEE, 2012, Prop. 20.8). This allows us to locally invert the exponential map. Say U is a neighbourhood of $0 \in \mathfrak{g}$ and V is a neighbourhood of $e \in G$ so that the exponential map is a diffeomorphism between them then we call its inverse the *logarithmic map* which we denote by $\log_G : V \to U$.

In our applications the exponential map will generally be surjective but not injective, meaning we often have $V = G$.

REPRESENTATIONS. Lie groups are often studied through their actions on vector spaces. Given a Lie group G and vector space V, then a *representation* is a pair (V, ϕ) with $\phi : G \to \mathrm{GL}(V)$ a group homomorphism, i.e. it has the property that $\phi(g_2 g_1) = \phi(g_2) \circ \phi(g_1)$ for all $g_1, g_2 \in G$. We say a representation is *effective* or *faithful* if it is injective.

An *irreducible representation*, or *irrep* for short, is a representation (V, ϕ) such that there is no (non-trivial) linear subspace $W \subset V$ that is closed under ϕ, i.e. $\phi(g)w \in W$ for all $w \in W$ and $g \in G$.

LIE SUBGROUPS. A Lie group H is a Lie subgroup of a Lie group G if (i) it is an immersed submanifold of G and (ii) it is a subgroup of G seen as an abstract group. In practice, most Lie subgroups are identified through *Cartan's closed subgroup theorem*, which states that any closed (in the topological sense) subgroup of a Lie group

is a Lie subgroup. In this case the subgroup inherits the subspace topology and differential structure from the Lie group.

One way to generate Lie subgroups is through the Lie algebra. Let $\mathfrak{g}$ be the Lie algebra of a Lie group G and $\mathfrak{h} \subset \mathfrak{g}$ is a Lie subalgebra, then there is a unique (connected) Lie subgroup $H \subset G$ that has $\mathfrak{h}$ as its Lie algebra. The reverse is also true, if H is a Lie subgroup of G then its Lie algebra $\mathfrak{h}$ is a subalgebra of the Lie algebra $\mathfrak{g}$ of G (Koda, 2009, Thm. 1.5).

2.2 GROUP ACTIONS & HOMOGENEOUS SPACES

Lie groups are mostly interesting for how they can act on other spaces and we will almost exclusively work in a setting where data lives on a space that has a Lie group acting on it. Let G be a Lie group and M a smooth manifold, then we say a smooth map $\rho : G \times M \to M$ is a smooth left *group action* if

- $\rho(g_2, \rho(g_1, p)) = \rho(g_2 g_1, p)$ for all $g_1, g_2 \in G$ and $p \in M$ and
- $\rho(e, p) = p$ for all $p \in M$.

Right actions can be defined similarly, but in our work we only require left actions, henceforth for the sake of brevity we will understand *action* to mean smooth left group action.

A left action can be converted into a right action and vice versa, hence focusing on one does not represent a loss of generality.

Next to the notation above it is also common to define $\rho_g := \rho(g, \cdot) : M \to M$ which is a diffeomorphism for each $g \in G$. When it is understood what the action is we can also write $g \cdot p := \rho(g, p)$. We use these three notations interchangeably depending on which is most convenient in any given setting.

Given an action we call $G \cdot p := \{g \cdot p \mid g \in G\}$ the *orbit* of the point $p \in M$ under the action of G.

For each $p \in M$, the *stabilizer* or *isotropy group* of p is given by

$$G_p := \{g \in G \mid g \cdot p = p\}, \tag{2.5}$$

which is always a subgroup of G due to the properties of the action. Moreover, since the action is smooth if follows that if $(g_n)_{n \in \mathbb{N}}$ is a sequence in G_p with $\lim_{n\to\infty} g_n = g \in G$ then

$$\rho(g, p) = \rho\left(\lim_{n\to\infty} g_n, p\right) = \lim_{n\to\infty} \rho(g_n, p) = p.$$

Hence G_p is closed and so by Lie's closed subgroup theorem we can conclude that G_p is a Lie subgroup of G.

By definition we have that $\rho_g(p) = p$ for all $g \in G_p$, so we also have that $(\rho_g)_*|_p : T_pM \to T_pM$ for all $g \in G_p$. Since by the properties of group actions we have $\rho_h \circ \rho_g = \rho_{hg}$ it follows that $(\rho_h)_* \circ (\rho_g)_* = (\rho_{hg})_*$. From this we conclude that the map $I_p : G_p \to \mathrm{GL}(T_pM)$ given by $I_p(g) := (\rho_g)_*$ is a representation, we call this map the *isotropy representation* at p.

If we have a Lie group G, a smooth manifold M and an action, then we say M is a *homogeneous space* with respect to G if the action is *transitive*, meaning: for all $p_1, p_2 \in M$ there exists at least one $g \in G$ so that $g \cdot p_1 = p_2$. More colloquially: for every two points in the homogeneous space there is at least one group element whose action takes the first point to the second.

PRINCIPAL & TRIVIAL HOMOGENEOUS SPACES. A Lie group is a homogeneous space with respect to itself with the action being simply group multiplication, i.e. $g \cdot h := gh$. This is called the *principal homogeneous space*. On the other end is the *trivial homogeneous space* $\{0\}$ with the trivial action $g \cdot 0 = 0$.

LIE GROUP QUOTIENTS. An important property of homogeneous spaces is that all of them are diffeomorphic to a group quotient of G by one of its closed subgroups H, i.e. to some G/H. We can see this by choosing a reference element $p_0 \in M$ and considering its stabilizer G_{p_0} per (2.5). Recall that the stabilizer is a closed group since $g \mapsto \rho(g, p_0)$ is continuous and so $\rho(\,\cdot\,, p_0)^{-1}$ is closed in G since $\{p_0\}$ is closed in M.

Now any $p \in M$ can be identified with a coset of G_{p_0} in the form $g_1 G_{p_0}$ where $g_1 \in G$ is any group element so that $g_1 \cdot p_0 = p$ (which always exists since the action is assumed to be transitive). This mapping is well defined since in the first place:

$$(g_1 G_{p_0}) \cdot p_0 = g_1 \cdot (G_{p_0} \cdot p_0) = g_1 \cdot p_0 = p,$$

and in the second place: if we have another $g_2 \in G$ with $g_2 \cdot p_0 = p$ then $g_1 \cdot p_0 = g_2 \cdot p_0$ and so $g_1^{-1} g_2 \cdot p_0 = p_0$ from which we conclude that $g_1^{-1} g_2 \in G_{p_0}$. Since G_{p_0} is a group itself we have $g_1^{-1} g_2 G_{p_0} = G_{p_0}$ then we have

$$g_1 G_{p_0} = g_1 g_1^{-1} g_2 G_{p_0} = g_2 G_{p_0}.$$

The map from the left cosets of G_{p_0} is a bijection that can be inverted by $p \mapsto g_p G_{p_0}$ where g_p is any group element so that $g_p \cdot p_0 = p$. Since the action is smooth this map forms a diffeomorphism between G/G_{p_0} and M, for formal proof see Lee (2012, Thm. 21.18).

For brevity we can denote the left cosets of G_{p_0} as

$$G_{p_0 \to p} := \{g \in G \mid g \cdot p_0 = p\} \tag{2.6}$$

and identify $p \equiv G_{p_0 \to p}$. Then the action of a $g \in G$ on a $p \in M$ is just left multiplication with the corresponding $G_{p_0 \to p}$, i.e. $g \cdot p \equiv g G_{p_0 \to p} = G_{p_0 \to g \cdot p}$.

Remark that the reference point $p_0 \in M$ is entirely arbitrary in this construction, so there is no special meaning to G_{p_0} being the only coset that is itself a subgroup of G. Choosing another reference point will make that point's corresponding coset a group and the coset corresponding to p_0 will (generally) stop being a group.

THE QUOTIENT SPACE IS ALL YOU NEED. Since any homogeneous space is diffeomorphic to some group quotient G/H with a closed subgroup H we can also just work with the quotient space and disregard the homogeneous space as a separate space. Any $p \in G/H$ is then a subset $p \subset G$ so that there exists at least one $g_p \in G$ so that $p = gH$. It then makes sense to say $g \in p$ in the same way as we could say $g \in G_{p_0 \to p}$ before. In this setting the reference element p_0 from before is exactly the closed subgroup H. We denote the quotient map as $\pi : G \to G/H$ given by

$$\pi(g) := gH. \tag{2.7}$$

The quotient map π is a smooth submersion by the *homogeneous space construction theorem* in Lee (2012, Thm. 21.17).

2.3 SOME EXAMPLES

The main Lie group that we use in this work is the *special Euclidean group* in two dimensions, or $SE(2)$ for short, which we will discuss extensively in Chapter 4. For now we introduce some other pairings of Lie groups and homogeneous spaces that are also relevant

for image processing applications but not directly used in our work.

TRANSLATION GROUP. The space $\mathbb{R}^n$ can itself be seen as a Lie group of translations where its group product is the usual vector addition, i.e. $xy := x + y$ for all $x, y \in \mathbb{R}^n$. When we want to emphasize the interpretation of $\mathbb{R}^n$ as a group of translations we denote it as $T_n := (\mathbb{R}^n, +)$. The translation action on itself as the principal homogeneous space is the obvious $x \cdot y = x + y$ for every $x \in \mathbb{T}_n$ and $y \in \mathbb{R}^n$.

SCALING AND TRANSLATION GROUP. The n-dimensional Lie group of scalings and translations is defined as $\mathrm{ST}(n) = \mathbb{R}^n \rtimes \mathbb{R}_{>0}$ where the group product is given by $(x_2, a_2)(x_1, a_1) = (x_2 + a_2 x_1, a_2 a_1)$ for all $(x_1, a_1), (x_2, a_2) \in \mathbb{R}^n \times \mathbb{R}_{>0}$. This group has as unit element $e = (0, 1)$ and the inverse is given by $(x, a)^{-1} = (-x/a, 1/a)$. Notice that this group is not commutative.

The space $\mathbb{R}^n$ is a homogeneous space of $\mathrm{ST}(n)$ under the action $(x, a) \cdot y = x + ay$ for all $(x, a) \in \mathrm{ST}(n)$ and $y \in \mathbb{R}^n$.

SIMILARITY GROUP. The n-dimensional Lie group of similarity transforms is defined as $\mathrm{SIM}(n) = \mathbb{R}^n \rtimes (\mathrm{SO}(n) \times \mathbb{R}_{>0})$ where the group product is given by

$$(x_2, R_2, a_2)(x_1, R_1, a_1) = (x_2 + a_2 R_2 x_1, R_2 R_1, a_2 a_1)$$

for all $(x_2, R_2, a_2), (x_1, R_1, a_1) \in \mathbb{R}^n \times \mathrm{SO}(n) \times \mathbb{R}_{>0}$. The unit element of $\mathrm{SIM}(n)$ is $e = (0, I, 1)$ and the inverse can be calculated as

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

and again the group is not commutative.

The space $\mathbb{R}^n$ is a homogeneous space of $\mathrm{SIM}(n)$ under the action

$$(x, R, a) \cdot y = x + aRy$$

for all $(x, R, a) \in \mathrm{SIM}(n)$ and $y \in \mathbb{R}^n$.

2.4 GROUP INVARIANT STRUCTURES

Certain structures on the homogeneous space such as vector fields, tensor fields, and metrics are special in that they are unaffected by the action of the group.

INVARIANT RIEMANNIAN STRUCTURE. Being smooth manifolds, homogeneous spaces can be equipped with a Riemannian metric. In this setting we are interested in the Riemannian metric being (left) *invariant* under the (left) action of the group. This means that under a group action ρ of a group G the Riemannian metric $\mathcal{G} \in \Gamma(T^*M \otimes T^*M)$ satisfies $(\rho_g)^*\mathcal{G} = \mathcal{G}$ for all $g \in G$.

If we are working in the principal homogeneous space, i.e. the group G itself, then the group action is just left multiplication, so the (left) invariance requirement can be phrased as $(L_g)^*\mathcal{G} = \mathcal{G}$ for all $g \in G$. The invariance requirement can also be stated in terms of the Lie algebra/left-invariant vector fields $\mathfrak{X}(G)$ by requiring that $\mathcal{G}(X, Y)$ must yield a constant function for all $X, Y \in \mathfrak{X}(G)$. This implies that there is a natural identification of left-invariant Riemannian metrics and inner products on $\mathfrak{X}(G)$.

This construction can be repeated for right invariance and right invariant vector fields.

In the general setting, where the homogeneous space M under consideration is not necessarily the group itself, the existence of an invariant Riemannian metric depends on the isotropy representation. Let $I_p : G_p \to \mathrm{GL}(T_pM)$ be the isotropy representation at $p \in M$, then there exists a G-invariant Riemannian metric on M if and only if the closure of $I_p(G_p)$ is compact in $\mathrm{GL}(T_pM)$. This result is detailed in LEE (2018, Thm. 3.17). In our applications the isotropy group itself will usually be compact, in which case $I_p(G_p)$ will also be compact by continuity and an invariant Riemannian metric exists.

PSEUDOMETRICS ON THE GROUP. If a G-invariant metric $\mathcal{G}$ exists on a homogeneous space G/H then it has a natural counterpart on G. Recall the quotient map $\pi : G \to G/H$ (2.7) given by $\pi(g) = gH$, then

$$\tilde{\mathcal{G}} := \pi^*\mathcal{G} \in \Gamma\left(T^{(0,2)}G\right) \tag{2.8}$$

is a *pseudometric tensor field* on the group G. It is not a full Riemannian metric since it clearly is degenerate in the direction of H, i.e.

$\|\log_G(h)\|_{\tilde{\mathfrak{g}}} = 0$ for all $h \in H$. As such $\|\cdot\|_{\tilde{\mathfrak{g}}}$ is a *seminorm* on any given tangent space and not a full norm.

In the same fashion as § 1.7 this pseudometric tensor field induces a *pseudometric* per

$$\mathrm{d}_{\tilde{\mathfrak{g}}}(g_1, g_2) := \inf_{\substack{\gamma \in \mathrm{Lip}([0,1],\, G) \\ \gamma(0)=g_1,\ \gamma(1)=g_2}} \int_0^1 \|\dot{\gamma}(t)\|_{\tilde{\mathfrak{g}}}\, \mathrm{d}t. \tag{2.9}$$

By construction we have that

$$\mathrm{d}_{\tilde{\mathfrak{g}}}(g_1, g_2) = \mathrm{d}_{\mathfrak{g}}(g_1 H, g_2 H)$$

and in particular $\mathrm{d}_{\tilde{\mathfrak{g}}}(g_1, g_2) = 0$ if $g_1, g_2 \in p$ for some $p \in G/H$.

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

LEMMA 2.1 Let $g_1, g_2 \in G$ so that $\pi(g_2)$ is away from the cut locus of $\pi(g_1)$, then:

$$d_{\tilde{\mathfrak{g}}}(g_1, g_2) = d_{\mathfrak{g}}(\pi(g_1), \pi(g_2)).$$

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

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

Denote the length functionals with:

$$\mathrm{Len}_G(\gamma) := \int_0^1 \|\dot{\gamma}(t)\|_{\tilde{\mathfrak{g}}}\, \mathrm{d}t,$$

$$\mathrm{Len}_{G/H}(\beta) := \int_0^1 \|\dot{\beta}(t)\|_{\mathfrak{g}}\, \mathrm{d}t.$$

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

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

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

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

Now we can compose these three curves:

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

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

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

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

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

□

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

INVARIANT VECTOR FIELDS. A vector field $X \in \Gamma(TM)$ is said to be G-invariant (the action ρ is inferred) if

$$\left(\rho_g\right)_* X = X$$

for all $g \in G$. In the case of the principal homogeneous space these are exactly the left-invariant vector fields but in general it may be that very few vector fields satisfy this condition, down to only the trivial vector field (i.e. the zero vector field) doing so.

INVARIANT TENSOR FIELDS. Since ρ_g is a diffeomorphism for any $g \in G$ the two specific cases above can be generalized to mixed higher-order tensors that can be both pushed forward and pulled back per (1.5).

3

EQUIVARIANCE

An image of an apple (top) follows by three geometric transformations (translation, rotation and scaling), the bottom image is an example of a non-geometric transformation that we will not consider.

A core notion in our work is that of *equivariance*, which is about systems whose output transforms in a predictable manner when their input is transformed in certain ways. Equivariance can be understood very broadly but for our purposes we consider a very specific type of geometric equivariance based on the Lie group and homogeneous space theory of Chapter 2. This chapter is based on Lee (2018, Ch. 7), Federer (2014, Ch. 2.7) , Cohen (2021) and Smets (2023, Ch. 3).

In the first two sections we make formal what we mean by equivariance in the homogeneous space setting, with emphasis on equivariant linear operators. In § 3.3 we cover our equivariance theorem in the setting of scalar valued maps on homogeneous spaces that admit covariant integrals. This theorem underpins our later design of equivariant neural networks.

3.1 EQUIVARIANT MAPS & OPERATORS

Suppose G is a Lie group and M and N are smooth manifolds with smooth (left) actions ρ^M and ρ^N. Then we can consider maps $F : M \to N$ that are *equivariant* with respect to those group actions, i.e.

$$F(\rho^M(g, p)) = \rho^N(g, F(p))$$

for all $g \in G$ and $p \in M$, or more concisely:

$$F(g \cdot p) = g \cdot F(p).$$

Equivalently, F is equivariant if the following diagram commutes for each $g \in G$:

$$\begin{array}{ccc} M & \xrightarrow{F} & N \\ {\scriptstyle \rho_g^M}\downarrow & & \downarrow{\scriptstyle \rho_g^N} \\ M & \xrightarrow{F} & N. \end{array}$$

ACTIONS ON FUNCTIONS. This idea extends naturally to operators between function spaces on those manifolds. Suppose X is some function space on M with arbitrary codomain and $f \in X$ then the action $\rho^M : G \times M \to M$ naturally induces an action $\rho^X : G \times X \to X$ by

$$\rho^X(g, f) := f \circ \rho^M(g^{-1}, \cdot). \tag{3.1}$$

Straightforward computation allows us to verify that ρ^X is indeed a left action. We call ρ^X the action on X *induced* by the action ρ^M on M.

If the space of functions X is a vector space, which it usually is, then the action ρ^X is linear in its second argument and so it is a representation.

EQUIVARIANT OPERATORS. Let X be a function space on M and Y a function space on N equipped with the corresponding representations ρ^X and ρ^Y per (3.1).

Then an operator $A : X \to Y$ is equivariant if

$$A \circ \rho_g^X = \rho_g^Y \circ A, \qquad \forall g \in G. \tag{3.2}$$

Or in words: for every group element doing the corresponding transform on the input space X and then applying the operator A gives the same results as first applying the operator A and then performing the transform corresponding to the group element on the output space Y. Or in words: letting a group element act on an input and then applying the operator A yields the same results as first applying the operator A and then letting the group element act on its output.

3.2 INTEGRAL OPERATORS & COVARIANT INTEGRATION

Let us now examine what linear operators between functions on homogeneous spaces are equivariant. Assume again that we

have a Lie group G and two of its homogeneous spaces M and N under the actions ρ^M respectively ρ^N. Denote the vector space of bounded and continuous functions on a space M by

$$\mathrm{BC}(M) := B(M) \cap C(M). \tag{3.3}$$

Let $X = \mathrm{BC}(M)$ and $Y = \mathrm{BC}(N)$ be the spaces of bounded and continous functions on M respectively N which are equipped with the actions induced on them per (3.1). Note that the space of bounded and continous functions $\mathrm{BC}(M)$ is a Banach space under the supremum norm $\|f\|_\infty := \sup_{p\in M} |f(p)|$.

INTEGRAL OPERATORS. An *integral operator* $A : X \to Y$ is given by

$$(Af)(q) := \int_M k_A(p,q) f(p)\, \mathrm{d}\mu_M(p) \tag{3.4}$$

where a measurable $k_A : M \times N \to \mathbb{R}$ is the so-called *kernel* of A and μ_M is a choice of Radon measure on M. As long as the kernel k_A satisfies

$$\sup_{q\in N} \int_M |k_A(p,q)|\, \mathrm{d}\mu_M(p) < \infty \tag{3.5}$$

then A is a bounded (and so continous) linear operator from X to Y in the supremum norm (we prove this statement in Lemma 3.1).

We can now ask ourselves when operators of the type (3.4) are equivariant, i.e. under what conditions do we have

$$A(g \cdot f) = A(\rho^X(g, f)) = \rho^Y(g, Af) = g \cdot (Af)$$

for all $f \in X$. It turns out we need two things for equivariance:

1. the integral needs to be *invariant* or *covariant*,
2. the kernel k_A needs to have the correct symmetry.

We tackle the invariant/covariant integral first.

INVARIANT INTEGRALS. When integrating over the *Lebesgue measure* on $\mathbb{R}^n$ we obtain an integral that is translation equivariant. Ideally we want the integral on M to be similarly invariant under action by G, or formally

$$\int_M (g \cdot f)(p)\, \mathrm{d}\mu_M(p) = \int_M f(g^{-1} \cdot p)\, \mathrm{d}\mu_M(p) = \int_M f(p)\, \mathrm{d}\mu_M(p) \tag{3.6}$$

for any integrable function $f : M \to \mathbb{R}$ and all $g \in G$. This imposes a condition on the measure μ_M, namely: for all measurable subsets $S \subset M$ and $g \in G$ we require $\mu_M(g \cdot S) = \mu_M(S)$ where $g \cdot S := \{g \cdot p \mid p \in S\}$. In other words we need a (non-trivial) group invariant measure to get the desired integral. When we integrate with respect to such a measure we say we are using an *invariant integral*. These G-invariant measures, or just *invariant measures*, do not always exists but when they do not it is sometimes still possible to obtain a covariant measure instead.

COVARIANT INTEGRALS. A *linear character* or just *character* of a Lie group G is a continuous group homomorphism from G to the multiplicative group of positive real numbers $(\mathbb{R}_{>0}, \cdot)$. A measure μ_M on M is said to be *covariant* with respect to a character $\chi : G \to \mathbb{R}_{>0}$ if

$$\mu_M(g \cdot S) = \chi(g)\, \mu_M(S)$$

χ being a character means $\chi(e) = 1$ and $\chi(gh) = \chi(g)\chi(h)$.

for all $g \in G$ and measurable $S \subset M$. Integrating with a covariant measure yields a *covariant integral* that satisfies

$$\int_M (g \cdot f)(p)\, \mathrm{d}\mu_M(p) = \chi(g) \int_M f(p)\, \mathrm{d}\mu_M(p) \tag{3.7}$$

for any integrable $f : M \to \mathbb{R}$ and all $g \in G$. The invariant measure/integral is then just a special case of the covariant measure/integral where $\chi \equiv 1$.

HAAR INTEGRALS. In the special case of the principal homogeneous space, i.e. $M = G$ with left multiplication as the action, there always exists an invariant measure and therefor an invariant integral. This measure is unique up to scalar multiplication and is called the (left) *Haar measure* of G. Integrating with the (left) Haar measure yields an invariant integral that we call the (left) *Haar integral*. The right Haar measure and integral similarly exist and are unique (up to scalar multiplication) but need not coincide with their left-invariant counterparts. See FEDERER (2014, Ch. 2.7) for details on this topic. We will only be use the left Haar integral so we will just refer to it as the Haar integral.

NOTATION. We will only be working in homogeneous spaces that admit invariant or covariant integrals and only ever consider one

per space, so we write integrals without naming the measure. Hence $\int_M \dots \mathrm{d}p$ will by default mean an invariant/covariant integral and $\int_G \dots \mathrm{d}g$ will mean the (left) Haar integral.

3.3 EQUIVARIANT LINEAR OPERATORS

Now that we have established the necessary properties of the integrals we can move on to examining what kernel symmetries are required for an integral operator to be equivariant. Many equivariance results of this type exist for a plethora of settings, hence ours is similar to existing work. Indeed, what we present here is almost the general result from COHEN (2021, Ch. 9.4) restricted to the scalar codomain, but ours is distinct in that we include the option of a covariant integral and we make a statement about the boundedness/continuity of our operators. Another similar result is by BEKKERS (2019) which uses the Radon-Nikodym derivative to deal with non-invariant integration instead of our covariant approach and uses different function spaces. For those reasons we present our specific results as propositions with proofs.

Our main result is Theorem 3.3, which will be an important tool in designing equivariant neural networks. We start with a lemma that specifies which symmetries the 2-argument kernel from (3.4) needs to have for its corresponding operator to be equivariant.

LEMMA 3.1 Let M and N be homogeneous spaces of a Lie group G so that M admits a covariant integral with character χ_M. Let A be an integral operator (3.4) from $\mathrm{BC}(M)$ to $\mathrm{BC}(N)$ with a kernel $k_A \in C(M \times N)$. Then

$$A(g \cdot f) = g \cdot (Af)$$

for all $g \in G$ and $f \in \mathrm{BC}(M)$ if and only if

$$\chi_M(g)\, k_A(g \cdot p, g \cdot q) = k_A(p, q) \tag{3.8}$$

for all $g \in G$, $p \in M$ and $q \in N$.
Moreover A is bounded (and so continuous) in the supremum norm if

$$\sup_{q \in N} \int_M |k_A(p, q)|\, \mathrm{d}p < \infty. \tag{3.9}$$

Proof.

"$\Rightarrow$"

Assuming A to be equivariant, take an arbitrary $g \in G$ and $f \in \mathrm{BC}(M)$ and substitute the definition of the group representation and A in the equality $g^{-1} \cdot A(g \cdot f) = Af$ to find

$$\int_M k_A(p, g \cdot q)\, f(g^{-1} \cdot p)\, \mathrm{d}p = \int_M k_A(p, q)\, f(p)\, \mathrm{d}p \tag{3.10}$$

for all $q \in N$.

Fix $q \in N$ and let $F(p) := k_A(g \cdot p, g \cdot q) f(p)$. Then observe that

$$(g \cdot F)(p) = k_A(g \cdot g^{-1} \cdot p, g \cdot q) f(g^{-1} \cdot p) = k_A(p, g \cdot q)\, f(g^{-1} \cdot p),$$

which is the left integrand from (3.10). Since we have assumed covariant integration we have

$$\int_M (g \cdot F)(p)\, \mathrm{d}p = \chi_M(g) \int_M F(p)\, \mathrm{d}p.$$

Applying this to (3.10) we find

$$\chi_M(g) \int_M k_A(g \cdot p, g \cdot q)\, f(p)\, \mathrm{d}p = \int_M k_A(p, q)\, f(p)\, \mathrm{d}p. \tag{3.11}$$

Since f was arbitrary and $p \mapsto k_A(p,q)$ continuous it follows that

$$\chi_M(g)\, k_A(g\cdot p, g\cdot q) = k_A(p,q)$$

for all $p \in M$.

"$\Leftarrow$"

Assuming $\chi_M(g)\, k_A(g\cdot p, g\cdot q) = k_A(p,q)$ for all $g \in G$, $p \in M$ and $q \in N$ then (3.11) follows for any choice of $f \in \mathrm{BC}(M)$, $g \in G$ and $q \in N$. Substituting the covariant integral the other way yields (3.10), which implies $g^{-1}\cdot A(g\cdot f) = Af$ since $q \in N$ is arbitrary. The function f and group element g were also chosen arbitrarily so the result follows for all $f \in \mathrm{BC}(M)$ and $g \in G$.

Finally, boundedness of A in the supremum norm follows from

$$\begin{aligned}\|Af\|_\infty &= \sup_{q\in N}\left|\int_M k_A(p,q)\, f(p)\,\mathrm{d}p\right| \le \sup_{q\in N}\int_M |k_A(p,q)|\,|f(p)|\,\mathrm{d}p \\ &\le \|f\|_\infty \cdot \sup_{q\in N}\int_M |k_A(p,q)|\,\mathrm{d}p \overset{(3.9)}{<} \infty.\end{aligned}$$

□

The condition on the kernel (3.9) is partially redundant with the symmetry requirement as the following lemma shows.

LEMMA 3.2 In the same setting as Lemma 3.1. If the kernel $k_A \in C(M\times N)$ satisfies the symmetry (3.8) and for some $q \in N$ satisfies

$$\int_M |k_A(p,q)|\,\mathrm{d}p < \infty$$

then

$$\|k_A(\cdot, q_1)\|_{L^1(M)} = \|k_A(\cdot, q_2)\|_{L^1(M)}$$

for all $q_1, q_2 \in N$.

Proof. Since N is a homogeneous space then for all $q_1, q_2 \in N$ there exists a $g \in G$ so that $q_1 = g \cdot q_2$, then

$$\begin{aligned}
\int_M |k_A(p, q_1)| \, \mathrm{d}p &= \int_M |k_A(p, g \cdot q_2)| \, \mathrm{d}p \\
&= \int_M \left|k_A(g \cdot g^{-1} \cdot p, g \cdot q_2)\right| \mathrm{d}p \\
(3.8) &= \frac{1}{\chi_M(g)} \int_M \left|k_A(g^{-1} \cdot p, q_2)\right| \mathrm{d}p \\
(3.7) &= \frac{\chi_M(g)}{\chi_M(g)} \int_M |k_A(p, q_2)| \, \mathrm{d}p \\
&= \int_M |k_A(p, q_2)| \, \mathrm{d}p.
\end{aligned}$$

□

The condition on the kernel from Lemma 3.1 can be exploited to express it as a function on M instead of $M \times N$. If we fix a $q_0 \in N$ and for all $q \in N$ we choose a $g_q \in G_{q_0 \to q}$ (i.e. so that $g_q \cdot q_0 = q$) then by (3.8) we have

$$\begin{aligned}
k_A(p, q) &= \chi_M(g_q^{-1}) \, k_A(g_q^{-1} \cdot p, g_q^{-1} \cdot q) \\
&= \chi_M(g_q^{-1}) \, k_A(g_q^{-1} \cdot p, q_0),
\end{aligned}$$

which fixes the second input of k_A. Consequently we could contain all the information of our kernel in a function that exists only on M as $\kappa_A(p) := k_A(p, q_0)$. This reduced 1-argument kernel κ_A still has some restrictions placed on it for the resulting operator to be equivariant, as the following theorem makes precise.

THEOREM 3.3 (Equivariant linear operators) Let M and N be homogeneous spaces of a Lie group G so that M admits a covariant integral with respect to a character χ_M of G. Fix a $q_0 \in N$ and let $\kappa_A \in C(M) \cap L^1(M)$ be *compatible*, i.e. have the property that

$$\forall h \in G_{q_0} : h \cdot \kappa_A = \chi_M(h)\, \kappa_A. \tag{3.12}$$

For each $q \in N$ choose a $g_q \in G$ so that $g_q \cdot q_0 = q$, then the operator A defined by

$$(Af)(q) := \frac{1}{\chi_M(g_q)} \int_M (g_q \cdot \kappa_A)(p)\, f(p)\, \mathrm{d}p$$

is a *well defined bounded linear operator* from $\mathrm{BC}(M)$ to $\mathrm{BC}(N)$ that is *equivariant* with respect to G. Conversely every equivariant integral operator with a kernel $k_A \in C(M \times N)$ and with $k_A(\cdot, q) \in L^1(M)$ for some $q \in N$ is of this form.

Proof.
"⇒"

Assuming we have a $\kappa_A \in C(M) \cap L^1(M)$ that satisfies (3.12). Define $k_A \in C(M \times N)$ by

$$k_A(p, q) := \frac{1}{\chi_M(g_q)} (g_q \cdot \kappa_A)(p).$$

Then k_A is well defined since it does not depend on the choice of g_q for a given $q \in N$. If g'_q is another group element with $g'_q \cdot q_0 = q$ then there exists a $h \in G_{q_0}$ so that $g'_q = g_q h$ (recall § 2.2). We can check k_A is invariant under choice of $h \in G_{q_0}$:

$$\begin{aligned} \frac{1}{\chi_M(g_q h)} (g_q \cdot h \cdot \kappa_A)(p) &= \frac{\chi_M(h)}{\chi_M(g_q)\chi_M(h)} (g_q \cdot \kappa_A)(p) \\ &= \frac{1}{\chi_M(g_q)} (g_q \cdot \kappa_A)(p). \end{aligned}$$

Note that k_A is measurable even if $q \mapsto g_q$ is not. The latter map could be made not measurable by pathological choices of g_q in each equivalence class but per (3.12) this does not affect k_A.

The kernel k_A also satisfies the symmetry requirement (3.8) from Lemma 3.1:

$$\begin{aligned}
\chi_M(g)\, k_A(g\cdot p, g\cdot q) &= \chi_M(g)\,\frac{1}{\chi_M(g_{(g\cdot q)})}(g_{(g\cdot q)}\cdot \kappa_A)(g\cdot p)\\
&= \chi_M(g)\,\frac{1}{\chi_M(gg_q)}(g\cdot g_q\cdot \kappa_A)(g\cdot p)\\
&= \frac{\chi_M(g)}{\chi_M(g)\chi_M(g_q)}(g_q\cdot \kappa_A)(g^{-1}g\cdot p)\\
&= \frac{1}{\chi_M(g_q)}(g_q\cdot \kappa_A)(p)\\
&= k_A(p,q).
\end{aligned}$$

By Lemma 3.2 we have

$$\sup_{q\in N}\int_M |k_A(p,q)|\,\mathrm{d}p = \|k_A(\cdot\,,q_0)\|_{L^1(M)} = \|\kappa_A\|_{L^1(M)} < \infty.$$

Consequently, A also satisfies (3.9) and is a bounded equivariant linear operator per Lemma 3.1.

"$\Leftarrow$"
Assuming we have an equivariant linear operator A with kernel $k_A \in C(M\times N)$ then we pick a fixed $q_0 \in N$ and define $\kappa_A \in C(M)$

$$\kappa_A(p) := k_A(p,q_0).$$

This reduced kernel κ_A satisfies the compatibility condition (3.12) since if $h \in G_{q_0}$ then

$$\begin{aligned}
(h\cdot\kappa_A)(p) &= k_A(h^{-1}\cdot p, q_0) = k_A(h^{-1}\cdot p, h^{-1}\cdot q_0)\\
&= \chi_M(h)\, k_A(p,q_0) = \chi_M(h)\,\kappa_A(p).
\end{aligned}$$

Since we required $k_A(\cdot\,,q) \in L^1(M)$ for some $q \in N$, we apply Lemma 3.2 to find

$$\|\kappa_A\|_{L^1(M)} = \|k_A(\cdot\,,q_0)\|_{L^1(M)} = \|k_A(\cdot\,,q)\|_{L^1(M)} < \infty.$$

□

GROUP CONVOLUTION. In the setting where both input and output homogeneous space are the principal homogeneous spaces, i.e. $G = M = N$, Theorem 3.3 simplifies greatly. Indeed $M = G$ admits a Haar integral so $\chi_M \equiv 1$ and the isotropy group of any group element is trivially $\{e\}$. Consequently, the compatibility condition (3.12) is trivially satisfied and so any $\kappa_A \in C(G) \cap L^1(G)$ defines an equivariant linear operator $A : \mathrm{BC}(G) \to \mathrm{BC}(G)$ by

$$(Af)(h) = \int_G (h \cdot \kappa_A)(g)\, f(g)\, \mathrm{d}g = \int_G \kappa_A(h^{-1}g)\, f(g)\, \mathrm{d}g.$$

We also call this operation *group cross-correlation* and denote it as

$$(\kappa \star_G f)(h) := \int_G \kappa(h^{-1}g)\, f(g)\ \mathrm{d}g.$$

As in the familiar $\mathbb{R}^n$ setting, group cross-correlation is closely related to *group convolution*, which is defined as

$$(\check{\kappa} *_G f)(h) := \int_G \check{\kappa}(g^{-1}h)\, f(g)\ \mathrm{d}g.$$

Group cross-correlation and group convolution are the same up to inversion of the kernel, i.e. $\kappa \star_G f = \check{\kappa} *_G f$ if $\kappa(g^{-1}) = \check{\kappa}(g)$ for all $g \in G$. If it is clear from the context that we are operating in a group we drop the subscripts from the $\star$ and $*$ symbols.

4

THE LIE GROUP SE(2)

The Lie group of primary interest in our research is the *special Euclidean group* in two dimensions, also known as the group of two-dimensional rigid body motions since it consists of the translations and rotations of two-dimensional objects. This Lie group will allow us to make image processing algorithms that are equivariant to both translation and rotation of the input image, which is a desirable property for various image processing tasks.

4.1 DEFINITION & ESSENTIAL PROPERTIES

The Lie group SE(2) is defined as $\mathrm{SE}(2) := (\mathbb{R}^2, +) \rtimes \mathrm{SO}(2)$, where $(\mathbb{R}^2, +)$ is the usual $\mathbb{R}^2$ as an additive group and $\mathrm{SO}(2) \subset \mathrm{GL}(2)$ is the one-dimensional group of planar rotations represented as orthogonal 2×2 matrices with determinant 1. The semi-direct group product of two elements $(x_1, R_1), (x_2, R_2) \in \mathrm{SE}(2)$ is given by

$$(x_2, R_2)(x_1, R_1) := (x_2 + R_2 x_1, R_2 R_1). \tag{4.1}$$

We use the semi-direct product symbol $\rtimes$ here to emphasize that SE(2) *is not the direct group product but that the right factor enters the computation of the left factor.*

The unit element of SE(2) is then given by $e = (0, I)$ and inversion by $(x, R)^{-1} = (-R^{-1}x, R^{-1})$. Clearly, SE(2) is not commutative.

We often parametrize the matrices of SO(2) by their angle of rotation $\theta \in \mathbb{R}/(2\pi\mathbb{Z})$ as

$$R(\theta) := \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

with the group product given by $R(\theta_1)R(\theta_2) = R(\theta_1 + \theta_2)$. Similarly, it can be convenient to use the so called *small angle identifica-*

tion $\mathrm{SE}(2) \equiv \mathbb{R}^2 \times [-\pi, \pi)$, in which case the group product can be written as

$$(x_2, \theta_2)(x_1, \theta_1) = (x_2 + R(\theta_2)x_1, (\theta_1 + \theta_2 + \pi \bmod 2\pi) - \pi)$$

The map $\mathrm{SE}(2) \to \mathbb{R}^2 \times [-\pi, \pi)$ *is not a chart since the image is not an open subset of* $\mathbb{R}^3$*, though as long as we are careful around the periodic boundary it serves as a convenient coordinate system.*

for all $(x_1, R_1), (x_2, R_2) \in \mathbb{R}^2 \times [-\pi, \pi)$. The unit element then becomes $e = (0, 0)$ and inversion is given by $(x, \theta)^{-1} = (-R(-\theta)x, -\theta)$ if $\theta \in (-\pi, \pi)$ and $(x, -\pi)^{-1} = (x, -\pi)$ if not.

HAAR INTEGRAL & MEASURE. The usual Lebesgue integral on $\mathbb{R}^2 \times [-\pi, \pi) \subset \mathbb{R}^3$ is also the Haar integral on SE(2). Consider $(y, \beta) \in \mathrm{SE}(2)$ acting on some integral function $f : \mathrm{SE}(2) \to \mathbb{R}$. Then

$$\begin{aligned}
&\int_{\mathbb{R}^2} \int_{-\pi}^{\pi} ((y, \beta) \cdot f)(x, \theta) \, \mathrm{d}\theta \, \mathrm{d}x \\
&= \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} f\left((y, \beta)^{-1}(x, \theta)\right) \mathrm{d}\theta \, \mathrm{d}x \\
&= \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} f(z, \phi) \begin{vmatrix} \cos(-\beta) & -\sin(-\beta) & 0 \\ \sin(-\beta) & \cos(-\beta) & 0 \\ 0 & 0 & 1 \end{vmatrix} \mathrm{d}\phi \, \mathrm{d}z \\
&= \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} f(z, \phi) \, \mathrm{d}\phi \, \mathrm{d}z.
\end{aligned}$$

Hence the integral is invariant under left translation of the function f. The left Haar integral implies the left Haar measure μ through

$$\mu(S) = \int_S \mathrm{d}g$$

for any measurable $S \subset \mathrm{SE}(2)$.

LIE ALGEBRA BASIS. The Lie algebra of SE(2) is denoted by se(2). Recall that the Lie algebra of a Lie group G is its tangent space at the unit $T_e G$ and is isomorphic to the set of its left invariant vector fields $\mathfrak{X}(G)$. The identification of $\mathrm{SE}(2) \equiv \mathbb{R}^2 \times [-\pi, \pi)$ gives us a natural basis of $\mathfrak{se}(2) = T_e \mathrm{SE}(2)$, namely

$$A_1 := (\partial_{x^1})_e \,, \quad A_2 := (\partial_{x^2})_e \,, \quad A_3 := (\partial_\theta)_e \,. \tag{4.2}$$

This basis of se(2) extends naturally to a basis $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ for $\mathfrak{X}(\mathrm{SE}(2))$ through the pushforward of the left multiplication L_g for all $g \in \mathrm{SE}(2)$:

$$(\mathcal{A}_1)_g := (L_g)_* A_1, \quad (\mathcal{A}_2)_g := (L_g)_* A_2, \quad (\mathcal{A}_3)_g := (L_g)_* A_3.$$

Working out these expressions leads to the following more tangible formulas where we let $g \equiv (x, \theta) \equiv (x^1, x^2, \theta)$:

$$\begin{aligned} (\mathcal{A}_1)_{(x,\theta)} &= \cos\theta\,(\partial_{x^1})_{(x,\theta)} + \sin\theta\,(\partial_{x^2})_{(x,\theta)}, \\ (\mathcal{A}_2)_{(x,\theta)} &= -\sin\theta\,(\partial_{x^1})_{(x,\theta)} + \cos\theta\,(\partial_{x^2})_{(x,\theta)}, \\ (\mathcal{A}_3)_{(x,\theta)} &= (\partial_\theta)_{(x,\theta)}. \end{aligned} \tag{4.3}$$

These three vector fields are illustrated in Figure 4.1.

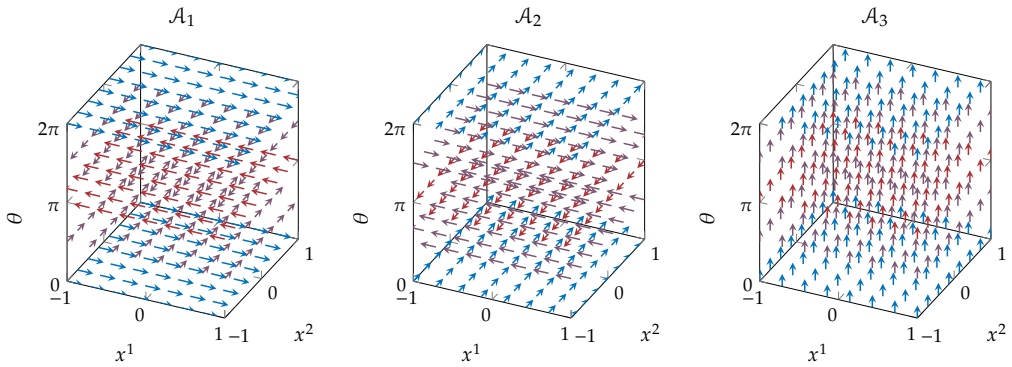


FIGURE 4.1: Our preferred basis for left invariant vector fields on SE(2) consists of the vector fields $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$ given by (4.3).

Naturally, left invariant vector fields are group equivariant operators on smooth functions under the action of left multiplication. Denote the induced action of left multiplication on a function $f \in C^\infty(\mathrm{SE}(2))$ with $g \cdot f := f \circ L_{g^{-1}}$, then

$$(\mathcal{A}_i)_h(g \cdot f) = (\mathcal{A}_i)_h(f \circ L_{g^{-1}}) = \Big((L_{g^{-1}})_*(\mathcal{A}_i)_h\Big) f = (\mathcal{A}_i)_{g^{-1}h} f,$$

for all $h \in \mathrm{SE}(2)$. At the same time we have

$$h \mapsto g \cdot ((\mathcal{A}_i)_h f) = (\mathcal{A}_i)_{g^{-1}h} f,$$

so we may conclude that $\mathcal{A}_i(g \cdot f) = g \cdot (\mathcal{A}_i f)$ for all $g \in G$, $f \in C^\infty(\mathrm{SE}(2))$ (or at least differentiable) and $i = 1, 2, 3$. Hence, left invariant vector fields form an important class of equivariant operators on a Lie group.

DUAL BASIS. The dual basis $(\omega^1, \omega^2, \omega^3) \subset \mathfrak{X}(\mathrm{SE}(2))^*$ for the left invariant covector fields is defined by $\omega^i(\mathcal{A}_j) = \delta^i_j$ and can be expressed in terms of the usual duals $(\mathrm{d}x^1, \mathrm{d}x^2, \mathrm{d}\theta)$ as

$$\begin{aligned}(\omega^1)_{(x,\theta)} &= \cos\theta\,(\mathrm{d}x^1)_{(x,\theta)} + \sin\theta\,(\mathrm{d}x^2)_{(x,\theta)},\\ (\omega^2)_{(x,\theta)} &= -\sin\theta\,(\mathrm{d}x^1)_{(x,\theta)} + \cos\theta\,(\mathrm{d}x^2)_{(x,\theta)},\\ (\omega^3)_{(x,\theta)} &= (\mathrm{d}\theta)_{(x,\theta)}.\end{aligned} \tag{4.4}$$

Naturally, this dual basis is also left invariant in the sense that $(L_g)^*\omega^i = \omega^i$ for all $g \in \mathrm{SE}(2)$ and $i = 1, 2, 3$.

STRUCTURE CONSTANTS. The structure constants of the left invariant frame $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ follow from straightforward computation with the expressions in (4.3) that yield

$$[\mathcal{A}_1, \mathcal{A}_2] = 0, \quad [\mathcal{A}_1, \mathcal{A}_3] = -\mathcal{A}_2, \quad [\mathcal{A}_2, \mathcal{A}_3] = \mathcal{A}_1. \tag{4.5}$$

Consequently, the only non-zero structure constants are

$$c^2_{13} = -1, \quad c^2_{31} = 1, \quad c^1_{23} = 1, \quad c^1_{32} = -1.$$

Being noncommutative the vector fields $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ are not the coordinate basis tangent vector of some chart.

4.2 INVARIANT RIEMANNIAN METRIC

On SE(2) a Riemannian metric $\mathcal{G} \in \Gamma(T^*\mathrm{SE}(2) \otimes T^*\mathrm{SE}(2))$ can be expressed in the above dual basis as

$$\mathcal{G} = \sum_{i,j=1}^{3} \mathcal{G}_{ij}\, \omega^i \otimes \omega^j \tag{4.6}$$

where each $\mathcal{G}_{ij} \in C^\infty(\mathrm{SE}(2))$ so that $[\mathcal{G}_{ij}(g)]$ is a symmetric positive definite matrix at each $g \in \mathrm{SE}(2)$.

We generally want the Riemannian metric to be *left invariant*, i.e. $(L_g)^* \mathcal{G} = \mathcal{G}$ for all $g \in G$. This requires that

$$\begin{aligned}\sum_{i,j=1}^{3} \mathcal{G}_{ij}\, \omega^i \otimes \omega^j = \mathcal{G} &= (L_g)^* \mathcal{G} \\ &= \sum_{i,j=1}^{3} \left(\mathcal{G}_{ij} \circ L_g\right) \left((L_g)^* \omega^i\right) \otimes \left((L_g)^* \omega^j\right) \\ &= \sum_{i,j=1}^{3} \left(\mathcal{G}_{ij} \circ L_g\right)\, \omega^i \otimes \omega^j\end{aligned}$$

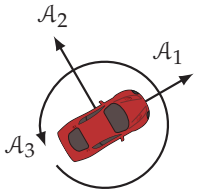
which requires that for all $i, j = 1, 2, 3$ and $g \in \mathrm{SE}(2)$ that $\mathcal{G}_{ij} \circ L_g = \mathcal{G}_{ij}$. Since for any $h_1, h_2 \in \mathrm{SE}(2)$ we can always find a $g \in \mathrm{SE}(2)$ so that $h_2 = g h_1$, the above equality can only hold if all the functions $\mathcal{G}_{ij}$ are constant. From which we conclude that all (left) invariant Riemannian metrics on SE(2) are of the form (4.6) with constants $\mathcal{G}_{ij} \in \mathbb{R}$ that form a symmetric positive definite matrix $[\mathcal{G}_{ij}] \in \mathbb{R}^{3\times 3}$. Hence, all invariant Riemannian metrics on SE(2) have only 6 degrees of freedom.

DIAGONAL METRIC. Most of the time we give a canonical interpretation to the left invariant vector fields $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$. We interpret $\mathcal{A}_1$ as being the *main direction* and interpret moving along it as *going forward*, similarly we think of $\mathcal{A}_2$ as pointing *sideways* and moving along or against it as moving *laterally* to the left or right. Finally we think of moving along or against $\mathcal{A}_3$ as rotating counterclockwise or clockwise.

Under this interpretation it makes sense to consider Riemannian metrics that are *diagonal* with respect to this basis, that is, the Riemannian metrics that are given by

$$\mathcal{G} = \sum_{i=1}^{3} \mathcal{G}_{ii}\, \omega^i \otimes \omega^i \tag{4.7}$$

where $\mathcal{G}_{ii} \in \mathbb{R}_{>0}$ for $i = 1, 2, 3$ are the 3 degrees of freedom of this family of Riemannian metrics. The parameters $\mathcal{G}_{11}, \mathcal{G}_{22}$ and $\mathcal{G}_{33}$ can be thought of as the cost of moving longitudinally, the cost of moving laterally and the cost of rotating respectively. We will use this type of Riemannian metric in several applications.



Interpretation of the standard bases of $\mathfrak{X}(\mathrm{SE}(2))$ *and* $\mathfrak{se}(2)$ *in terms of the movements of a car: forward and backward along* $\mathcal{A}_1$*, lateral slip along* $\mathcal{A}_2$ *and rotation along* $\mathcal{A}_3$*.*

4.3 EXPONENTIAL AND LOGARITHMIC MAP

The exponential map $\exp_{\mathrm{SE}(2)} : \mathfrak{se}(2) \to \mathrm{SE}(2)$ is given in terms of the basis A_1, A_2, A_3 and the small angle identification $(x, \theta) \in \mathbb{R}^2 \times [-\pi, \pi)$ by

$$\exp_{\mathrm{SE}(2)}(c^1 A_1 + c^2 A_2 + c^3 A_3) = \begin{bmatrix} \left(c^1 \cos\frac{c^3}{2} - c^2 \sin\frac{c^3}{2}\right) \operatorname{sinc}\frac{c^3}{2} \\ \left(c^1 \sin\frac{c^3}{2} + c^2 \cos\frac{c^3}{2}\right) \operatorname{sinc}\frac{c^3}{2} \\ c^3 \bmod 2\pi \end{bmatrix}, \tag{4.8}$$

Using the small angle identification implies $\theta \bmod 2\pi \in [-\pi, \pi)$ for the purpose of numerical computations.

for any $c^1, c^2, c^3 \in \mathbb{R}$, where the $\operatorname{sinc} : \mathbb{R} \to \mathbb{R}$ function is given by

$$\operatorname{sinc}(x) := \begin{cases} \frac{\sin x}{x} & \text{if } x \neq 0, \\ 1 & \text{if } x = 0. \end{cases}$$

See also Bekkers (2017, §2.3.3.). Clearly, (4.8) is not injective. It is surjective however, as can be seen from the logarithmic map $\log_{\mathrm{SE}(2)} : \mathrm{SE}(2) \to \mathfrak{se}(2)$ given in terms of the same basis and *small angle coordinates* as

$$\begin{aligned} \log_{\mathrm{SE}(2)}(x^1, x^2, \theta) = & \frac{x^1 \cos\frac{\theta}{2} + x^2 \sin\frac{\theta}{2}}{\operatorname{sinc}\frac{\theta}{2}} A_1 \\ & + \frac{-x^1 \sin\frac{\theta}{2} + x^2 \cos\frac{\theta}{2}}{\operatorname{sinc}\frac{\theta}{2}} A_2 \\ & + \theta A_3. \end{aligned} \tag{4.9}$$

Computation verifies that the logarithmic map is a right inverse of the exponential map, i.e. $\exp_{\mathrm{SE}(2)} \circ \log_{\mathrm{SE}(2)} = \mathrm{id}_{\mathrm{SE}(2)}$, but it is not a left inverse. Technically we choose a *branch* of the logarithmic map here since we can replace (4.9) by

$$\begin{aligned} & \frac{\theta + 2\pi n}{\theta} \cdot \frac{x^1 \cos\frac{\theta}{2} + x^2 \sin\frac{\theta}{2}}{\operatorname{sinc}\frac{\theta}{2}} A_1 \\ & + \frac{\theta + 2\pi n}{\theta} \cdot \frac{-x^1 \sin\frac{\theta}{2} + x^2 \cos\frac{\theta}{2}}{\operatorname{sinc}\frac{\theta}{2}} A_2 \\ & + (\theta + 2\pi n) A_3 \end{aligned}$$

for any $n \in \mathbb{Z}$ and still get a right inverse of the exponential map but one that maps to a different subset of se(2).

EXPONENTIAL COORDINATES OF THE FIRST KIND. The coefficients from (4.9) can also serve as a coordinate system when taken on their own:

$$\begin{aligned} c^1(x^1, x^2, \theta) &:= \frac{x^1 \cos\frac{\theta}{2} + x^2 \sin\frac{\theta}{2}}{\operatorname{sinc}\frac{\theta}{2}} \\ c^2(x^1, x^2, \theta) &:= \frac{-x^1 \sin\frac{\theta}{2} + x^2 \cos\frac{\theta}{2}}{\operatorname{sinc}\frac{\theta}{2}} \\ c^3(x^1, x^2, \theta) &:= \theta \end{aligned} \tag{4.10}$$

which are called the *exponential coordinates of the first kind*.

4.4 THE SPACE OF POSITION AND ORIENTATIONS $\mathbb{M}_2$

The space of positions and orientations in d dimensions is defined as the smooth manifold $\mathbb{M}_d := \mathbb{R}^d \times S^{d-1}$. For the case $d = 2$ we have $\mathbb{M}_2 = \mathbb{R}^2 \times S^1 \equiv \mathrm{SE}(2)$, though for $d > 2$ it is not the case that $\mathbb{M}_d$ is equivalent to $\mathrm{SE}(d) \equiv \mathbb{R}^d \rtimes \mathrm{SO}(d)$ since the former has $2d - 1$ dimensions and the latter has $1/2 d(d-1)$ dimensions. Since $\mathbb{M}_2 \equiv \mathrm{SE}(2)$ and each Lie group is a homogeneous space of itself, $\mathbb{M}_2$ is a homogeneous space of SE(2).

It may seem redundant to assign a different name and symbol to what is essentially the same object but the two different names emphasize the two roles a Lie group can play: first as a space in which we do geometry and let functions live (the homogeneous space) and second as a set of transformations that can act on that space and encodes its symmetries (the Lie group). We will use both the designations $\mathbb{M}_2$ and SE(2) when we want to emphasize those different roles.

4.5 $\mathbb{R}^2$ AS A HOMOGENEOUS SPACE

ACTIONS. The standard way by which SE(2) acts on $\mathbb{R}^2$ is by translating and rotating the plane, or more formally through the (left) action $\rho : \mathrm{SE}(2) \times \mathbb{R}^2 \to \mathbb{R}^2$ given by

$$g \cdot y = (x, \theta) \cdot y = \rho_g(y) = \rho(g, y) := x + R(\theta)y \tag{4.11}$$

for any $g \equiv (x, \theta) \in \mathrm{SE}(2)$ and $y \in \mathbb{R}^2$. We choose any of the equivalent notations depending on what is convenient in a given setting. This action naturally induces an action on functions on $\mathbb{R}^2$ through $(g \cdot f)(y) := f(g^{-1} \cdot y)$ for any $f : \mathbb{R}^2 \to \mathbb{R}$.

The standard frame on $\mathbb{R}^2$ is given by the vector fields ∂_{y^1} and ∂_{y^2} whose duals we denote with $\mathrm{d}y^1$ and $\mathrm{d}y^2$. In terms of the frame $(\partial_{y^1}, \partial_{y^2})$, the pushforward of the action with a group element $g \in \mathrm{SE}(2)$ on a vector field is determined by

$$\begin{aligned} (\rho_g)_* \partial_{y^1} = (\rho_{(x,\theta)})_* \partial_{y^1} &= \quad \cos\theta\, \partial_{y^1} + \sin\theta\, \partial_{y^2}, \\ (\rho_g)_* \partial_{y^2} = (\rho_{(x,\theta)})_* \partial_{y^2} &= -\sin\theta\, \partial_{y^1} + \cos\theta\, \partial_{y^2}. \end{aligned} \tag{4.12}$$

Similarly for the coframe $(\mathrm{d}y^1, \mathrm{d}y^2)$ we have

$$\begin{aligned} (\rho_g)^* \,\mathrm{d}y^1 = (\rho_{(x,\theta)})^* \,\mathrm{d}y^1 &= \quad \cos\theta\ \mathrm{d}y^1 + \sin\theta\ \mathrm{d}y^2, \\ (\rho_g)^* \,\mathrm{d}y^2 = (\rho_{(x,\theta)})^* \,\mathrm{d}y^2 &= -\sin\theta\ \mathrm{d}y^1 + \cos\theta\ \mathrm{d}y^2, \end{aligned} \tag{4.13}$$

so that

$$\left((\rho_g)^* \,\mathrm{d}y^i\right)\left((\rho_g)_* \partial_{y^j}\right) = \delta^i_j$$

for all $g \in \mathrm{SE}(2)$. Note how the previous equations only depend on angular coordinate θ and not on the translation coordinates $x = (x^1, x^2)$, this is expected since the vector fields $\partial_{y^1}, \partial_{y^2}$ and covector fields $\mathrm{d}y^1, \mathrm{d}y^2$ are translation invariant but not rotation invariant.

INVARIANT INTEGRAL. The Lebesgue integral on $\mathbb{R}^2$ is invariant under action by SE(2) since the Lebesgue measure is rotation and translation invariant, i.e. rotating a translating a shape does not change its Lebesgue measure/surface area. Hence we have

$$\int_{\mathbb{R}^2} (g \cdot f)(y)\, \mathrm{d}y = \int_{\mathbb{R}^2} f(y)\, \mathrm{d}y$$

for all $g \in \mathrm{SE}(2)$ and (integrable) $f : \mathbb{R}^2 \to \mathbb{R}$.

INVARIANT RIEMANNIAN METRIC. Riemannian metrics on $\mathbb{R}^2$ that are invariant under action by SE(2) are very limited. In fact, only isotropic metrics are invariant under SE(2) , that is, the metric has to be a scalar multiple of the standard Euclidean metric $\mathcal{G}^{\text{euclid}} := dy^1 \otimes dy^1 + dy^2 \otimes dy^2$.

Part II

DEEP LEARNING

The second part of the thesis details our research into neural networks. Our chief effort here consists of developing a neural network architecture for image processing and related tasks that is based on *partial differential equations*. We call this new type of neural network a *PDE-based group convolutional neural network* or *PDE-G-CNN* for short. Such an architecture adds a strong geometric prior to the network that is beneficial in low-data applications that require strong equivariance. The PDE based approach complements the group equivariance approach by giving us a set of equivariant operators for any choice of group or homogeneous space. We first summarize *group equivariant convolutional neural networks* in Chapter 5 before we develop PDE-based networks on top of them in Chapter 6.

5

GEOMETRIC EQUIVARIANCES IN NEURAL NETWORKS

In this chapter we cover the design of *group equivariant convolutional networks* for imaging applications. In many image processing tasks it is desirable for a system to exhibit equivariance with respect to typical geometric transforms such as translation, rotation and scaling. Group equivariant convolutional networks (or G-CNNs) as put forward by Cohen (2021), Cohen, Geiger, and Weiler (2020), and Cohen and Welling (2016) provide a framework for designing neural networks that includes these symmetries as a structural part of the network. Their work builds on earlier *rigid-motion scattering* networks by Sifre (2014) and Sifre and Mallat (2013, 2014) that also take equivariant processing as per Duits (2005, Thm. 21) and Duits and Burgeth (2007) as their starting point.

Saying a model is group equivariant *and* convolutional *is technically redundant since the group equivariance requirement will enforce a convolutional structure on its linear operators.*

The G-CNN framework has yielded many follow-on works that can be roughly categorized as

- discrete G-CNNs (Cohen and Welling, 2016; Dieleman, De Fauw, and Kavukcuoglu, 2016; Dieleman, Willett, and Dambre, 2015; Winkels and Cohen, 2018; Worrall and Brostow, 2018),
- regular continuous G-CNNs (Bekkers, 2019; Bekkers, Lafarge, et al., 2018; Finzi, Stanton, Izmailov, and Wilson, 2020; Lafarge, Bekkers, Pluim, Duits, and Veta, 2021; Oyallon and Mallat, 2015),
- and steerable continuous G-CNNs (Esteves, Allen-Blanchette, Makadia, and Daniilidis, 2018; Kondor and Trivedi, 2018; Weiler and Cesa, 2019; Weiler, Hamprecht, and Storath,

2018; Worrall, Garbin, Turmukhambetov, and Brostow, 2017) that rely on Fourier transforms on homogeneous spaces (Chirikjian and Kyatkin, 2000; Duits, Bekkers, and Mashtakov, 2019).

Both regular and steerable G-CNNs arise naturally from linear mappings between functions on homogeneous spaces that have equivariance constraints placed on them. Regular G-CNNs have to extend the domain and lift feature maps to a higher homogeneous space to obtain equivariance, as we will see in this chapter.

Steerable CNNs on the other hand extend the codomain by generating fiber bundles in which a steerable feature vector is assigned to each position in the base domain. Although steerable operators have benefits in terms of computational efficiency (Franken, van Almsick, Rongen, Florack, and ter Haar Romeny, 2006; Reisert, 2008), working with steerable representations puts constraints on activations within the network which limits its representation power. Hence, we take regular continuous G-CNNs as our point of departure.

CHAPTER OUTLINE. After a brief re-examination of traditional convolutional neural networks (CNNs) from the viewpoint of equivariance we will develop the general architecture of G-CNNs starting from the theory in Chapter 3 and then work out the details for the rotation-translation setting where we use the Lie group SE(2).

5.1 CONVOLUTIONAL NEURAL NETWORKS

LeCun et al. (1989) is also known as the LeNet paper and Krizhevsky, Sutskever, and Hinton (2012) as the AlexNet paper.

The early convolutional neural networks from LeCun et al. (1989) and later Krizhevsky, Sutskever, and Hinton (2012) are the original group equivariant neural network *avant la lettre*. Convolutional neural networks are commonly abbreviated to CNNs or ConvNets. These networks have three main ingredients:

1. convolution layers,
2. pointwise activation functions,
3. pooling operations.

We will discuss the pooling operations separately in § 5.3. In this section we cover the first two ingredients.

CNN FROM TRANSLATION EQUIVARIANCE. We can in retrospect derive the CNN architecture from first principles using translation equivariance. Take the n-dimensional translation group $T_n = (\mathbb{R}^n, +)$ that acts on the homogeneous space $\mathbb{R}^n$ via the action $x \cdot y := x + y$ for every $x \in T_n$ and $y \in \mathbb{R}^n$. Then by applying Theorem 3.3 we find that an equivariant linear integral operator between bounded and continuous functions $A : \mathrm{BC}(\mathbb{R}^n) \to \mathrm{BC}(\mathbb{R}^n)$ has to be of the form

$$\begin{aligned}(Af)(x) &= \int_{\mathbb{R}^n} (x \cdot \kappa_A)(y)\, f(y)\, \mathrm{d}y \\ &= \int_{\mathbb{R}^n} \kappa_A(x^{-1} \cdot y)\, f(y)\, \mathrm{d}y \\ &= \int_{\mathbb{R}^n} \kappa_A(y - x)\, f(y)\, \mathrm{d}y,\end{aligned}$$

where we use the Lebesgue measure (which is translation invariant) and where the compatibility condition on $\kappa_A \in C(\mathbb{R}^n) \cap L^1(\mathbb{R}^n)$ is trivial since we are working in the principal homogeneous space of T_n. This last equation is nothing more than the usual cross-correlation equation or convolution equation if we invert κ_A.

DISCRETIZATION. To implement a CNN one naturally needs to discretize the continuous convolution/cross-correlation operation, usually on a regular finite grid. Our continuous notion of equivariance makes little sense in the discrete world. So when we say a discrete implemented network is equivariant we mean that it would be equivariant in the continuous setting. We could call this *equivariance up to discretization*. For practical applications we count on being able to discretize finely enough to have good enough discrete equivariance for a given application's requirements.

CNN LAYERS. We conclude that for a traditional neural network that operates on spatial or temporal data (such as images and audio) to be translation equivariant it must be based around convolution. In addition to convolution any type of pointwise computation with data is of course also equivariant, which we will need for the combining of channels and the activation function. Hence, given a set of input functions $(u_i^{\mathrm{in}})_{i=1}^{C_{\mathrm{in}}} \subset \mathrm{BC}(\mathbb{R}^n)$ we can construct a *con-*

'Traditional' neural network should be understood as having trainable linear operators and pointwise activation functions as in BENGIO, GOODFELLOW, *and* COURVILLE *(2017, Ch. 6).*

volutional layer that outputs a set of functions $(u_j^{\text{out}})_{j=1}^{C_{\text{out}}} \subset \mathrm{BC}(\mathbb{R}^n)$ as follows:

$$u_j^{\text{out}} = \sigma\left(\sum_{i=1}^{C_{\text{in}}} a_{ij}\,(k_i * u_i^{\text{in}}) + b_j\right) \tag{5.1}$$

where each $k_i : \mathbb{R}^n \to \mathbb{R}$ is a trainable kernel and each $a_{ij} \in \mathbb{R}$ is a trainable coefficient, the bias term $b_j \in \mathbb{R}$ is optional and $\sigma : \mathbb{R} \to \mathbb{R}$ is some choice of activation function that is applied pointwise. As is conventional in deep learning literature we will also refer to these u functions as *feature maps*. When not referring to a specific instance of a feature map we also refer to them as *channels*, so we would say about the above operator $(u_i^{\text{in}})_{i=1}^{C_{\text{in}}} \mapsto (u_j^{\text{out}})_{j=1}^{C_{\text{out}}}$ as having C_{in} *input channels* and C_{out} *output channels*. The layer in (5.1) is called a *single channel convolution layer* or *depthwise convolution layer*. Figure 5.1 illustrates this type of layer in the 2-dimensional discretized setting.

The activation functions can be considered as part of the convolution layer as we do here or as a separate layer.

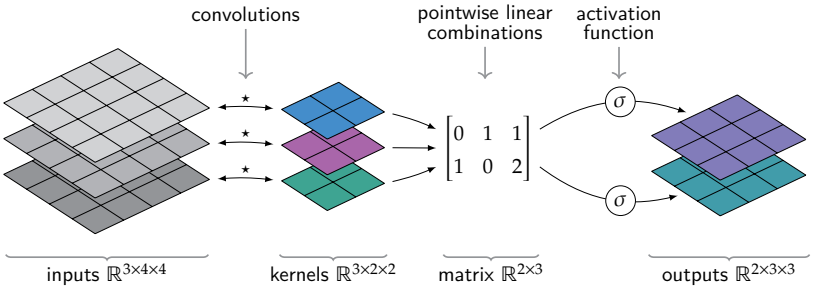


FIGURE 5.1: In a *single channel convolution layer*, also called *depthwise convolution layer*, each input channel is assigned a single kernel. After doing a cross-correlation/convolution on each input channel we take pointwise linear combinations of the resulting maps to generate the desired number of output channels. The biases (b in (5.1)) are omitted.

The alternative to the single channel convolution layer is the *multi channel convolution layer*, which is given by the formula

$$u_j^{\text{out}} = \sigma\left(\sum_{i=1}^{C_{\text{in}}} k_{ij} * u_i^{\text{in}} + b_j\right) \tag{5.2}$$

where each for each $i = 1 \dots C_{\text{in}}$ and $j = 1 \dots C_{\text{out}}$, $k_{ij} : \mathbb{R}^n \to \mathbb{R}$ is a trainable kernel. Hence, in this setup each combination of input

and output channel is assigned its own kernel. This type of layer is illustrated in Figure 5.2.

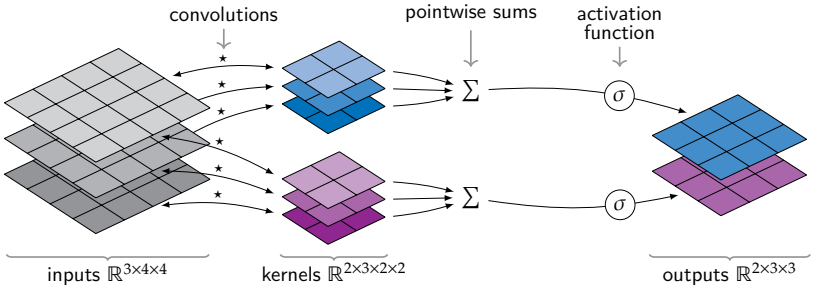
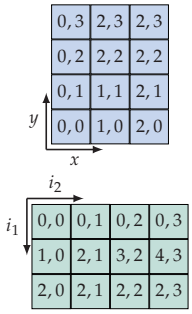


FIGURE 5.2: In a *multi channel convolution layer* each output channel is assigned a stack of kernels, where each stack has a kernel for each input channel. This results in $C_{\text{out}} \cdot C_{\text{in}}$ kernels. The resulting outputs of the $C_{\text{out}} \cdot C_{\text{in}}$ cross-correlations/convolutions are then summed up pointwise for each output channel resulting in C_{out} output channels. The biases (b in (5.2)) are omitted.

The single and multi-channel convolution designs are equivalent in the sense that for every network using one type of layer an equivalent network can be constructed using the other type (possibly with more layers when going from multi-channel to single-channel convolution layers). The benefit of the single channel convolution layer is that it logically separates *intra channel* processing (the convolutions) and *inter channel* processing (the linear combination of channels) while these are intermingled in a multi channel convolution layer.



Several image coordinate conventions exist. The Cartesian index convention (top) is perhaps more natural mathematically but the array index convention (bottom) is ubiquitous in implementations and used in Figure 5.3 and 5.4.

DISCRETIZING. To be implemented (5.1) and (5.2) need to be discretized, we will briefly summarize how this is normally done. For imaging and other temporal/spatial modalities, discretizing is usually done using a uniform rectangular grid. In this case both the input and kernel are samples on a finite rectangular grid and the kernel slides over the input grid. The finite pointwise multiplication and sum of the kernel elements with their corresponding input elements yields an output element. This is illustrated in Figure 5.3.

Since the input grid is in practice finite and the kernel consists of more than one element, the output size of the discrete convolution

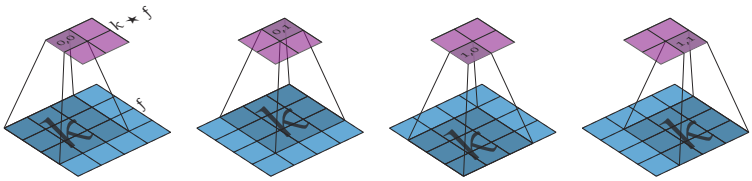


FIGURE 5.3: An illustration of discrete convolution (or cross-correlation) in 2D. Here the input $f \in \mathbb{R}^{4\times4}$ (colored blue) is convolved with the kernel $k \in \mathbb{R}^{3\times3}$, which yields an output $(k \star f) \in \mathbb{R}^{2\times2}$ (colored purple). There is no padding applied to the input and the kernel moves strides of 1 in each direction which shrinks the output.

does not need to match the size of the input as is the case in Figure 5.3. When shrinking of the feature maps is not desirable we add *padding* as in Figure 5.4 to be able to preserve the feature map dimensions.

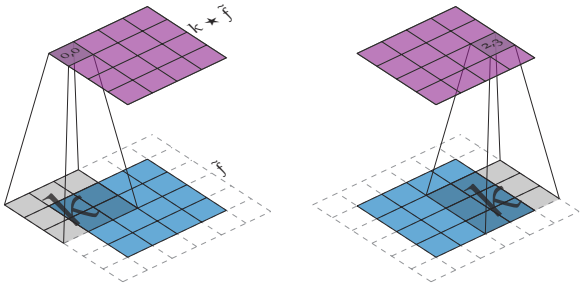


FIGURE 5.4: Adding padding to the input $f \in \mathbb{R}^{4\times4}$ creates a larger effective input $\tilde{f} \in \mathbb{R}^{6\times6}$ that allows the output of the convolution operation to retain the size of the original input. The most common type of padding is zero-padding, where each out-of-bounds value is assumed to be zero.

In the examples from Figure 5.3 and Figure 5.4 the kernel is translated over the input in single element increments. We can also move the kernel multiple steps each time. How many steps the kernel is moved each time in any given direction is called the *stride*. An example of convolution with strides larger than 1 is shown in Figure 5.5.

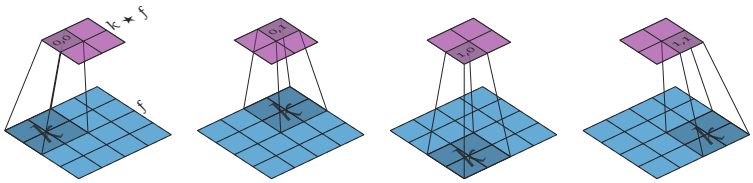


FIGURE 5.5: In discrete convolution (or cross-correlation) the kernel can stride over the input multiple elements at a time. Here the kernel has a stride of 2 in each dimension.

ADDING EQUIVARIANCES TO A CNN. Now suppose we wish to add an additional equivariance to a 2-dimensional CNN, say with respect to rotation or scaling, both natural requirements in many image processing applications.

Starting with adding rotation equivariance we would upgrade to the group SE(2) and take the setting from Chapter 4. The Lebesgue measure on $\mathbb{R}^2$ is rotation and translation invariant, so we have an invariant integral. But this time, the homogeneous space $\mathbb{R}^2$ (that we use both on the input and output side) is not the principal homogeneous space of SE(2) and so as per Theorem 3.3 we have a non-trivial compatibility condition on our kernels. Specifically, let the origin $0 = (0, 0) \in \mathbb{R}^2$ be the reference element of the homogeneous space. Then its stabilizer is $G_0 = \{0\} \times \mathrm{SO}(2) \subset \mathrm{SE}(2)$, i.e. just the roto-translations mapping the origin to itself. The compatibility condition on any kernel $k : \mathbb{R}^2 \to \mathbb{R}$ we use is then $h \cdot k = k$ for all $h \in G_0$, or more concretely

$$k(R(\theta)x) = k(x)$$

for all $\theta \in [0, 2\pi)$ and $x \in \mathbb{R}^2$, meaning that the kernels need to be isotropic. This is a severely restrictive condition on the kernels, one that would cripple a network since it precludes any detection of directional features.

Thing will get even worse if we want to add scaling equivariance on top of rotation equivariance. To be equivariant with respect to the SIM(2) group the kernel would need to be rotation and scale invariant, i.e.

$$k(aR(\theta)x) = k(x)$$

for all $a > 0$, $\theta \in [0, 2\pi)$ and $x \in \mathbb{R}^2$. This only holds if the kernel is constant, which would preclude any meaningful processing. Clearly, to design a neural network architecture for image processing (and other spatial/temporal modalities) that has these equivariances another approach is called for. The answer is the G-CNN that we will explore next.

5.2 GROUP EQUIVARIANT CONVOLUTIONAL NEURAL NETWORK

Say we need a neural network layer that takes in a set of scalar functions on a space M and outputs a set of scalar functions on a space N. We will restrict ourselves to continuous and bounded functions as in Chapter 3. We call the input function space $X = \mathrm{BC}(M)$ and the output function space $Y = \mathrm{BC}(N)$. We assume M and N are both homogeneous spaces with respect to some Lie group G so that we can talk about equivariances in the sense of Chapter 3.

From Theorem 3.3 we know how to construct an equivariant linear operator from X to Y. The symmetry of the kernel of the operator is dictated by the compatibility condition (3.12) which depends only on the choice of output homogeneous space N. Indeed, the larger dimensional N is, the easier the compatibility condition. When N is the principal homogeneous space the condition is trivial with no extra symmetry requirement placed on the kernel. This is then how we will escape the overly restrictive kernel symmetries we stumbled over in the previous section: by expanding the output domain of our operators.

LIFTING TO $\mathrm{SE}(2)$. If we apply Theorem 3.3 for $M = \mathbb{R}^2$ and $N = \mathbb{M}_2 = \mathrm{SE}(2)$ we obtain an equivariant linear operator $A : \mathrm{BC}(M) \to \mathrm{BC}(N)$ of the form

$$(Af)(x, \theta) = \int_{R^2} ((x, \theta) \cdot k)(y)\, f(y)\, \mathrm{d}y. \tag{5.3}$$

where $f, k : \mathbb{R}^2 \to \mathbb{R}$ and k has no further symmetry requirements for A to be equivariant. We can rework this equation by noting that

$$\begin{aligned}((x,\theta)\cdot k)(y) &= ((x,0)\cdot(0,\theta)\cdot k)(y)\\ &= ((0,\theta)\cdot k)((x,0)^{-1}\cdot y)\\ &= ((0,\theta)\cdot k)(y-x),\end{aligned}$$

and so we can rewrite the expression for A as

$$(Af)(x,\theta) = \int_{\mathbb{R}^2} ((0,\theta)\cdot k)(y-x)\, f(y)\,\mathrm{d}y = \big(((0,\theta)\cdot k)\star f\big)(y). \tag{5.4}$$

This means that we can express the operator A as a convolution with various rotated versions of the kernel k given by $(0,\theta)\cdot k = x \mapsto k(R(-\theta)x)$ for each $\theta \in [0, 2\pi)$. This process is called *lifting* to SE(2). This operation can readily be discretized as is illustrated in Figure 5.6. This is the approach taken by Bekkers, Lafarge, et al. (2018) and Lafarge, Bekkers, Pluim, Duits, and Veta (2021).

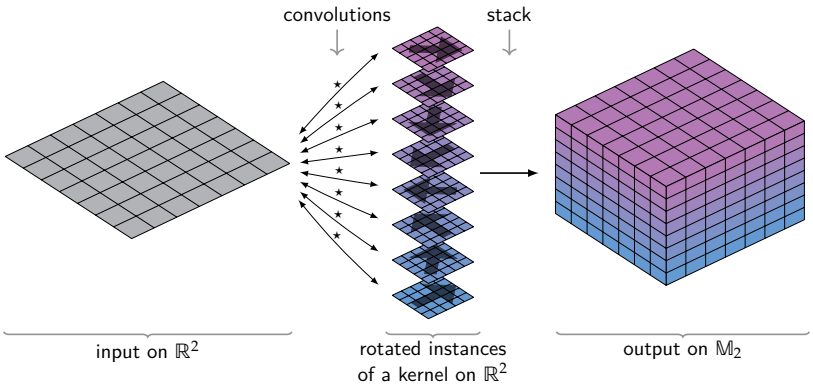


FIGURE 5.6: An illustration of how (discrete) feature maps on $\mathbb{R}^2$ are *lifted* to $\mathbb{M}_2$ (or SE(2)) in an equivariant manner. A set of rotated instances of the same $\mathbb{R}^2$ kernel is made and a convolution is performed on the input map with each instance of the kernel. Not shown is the padding needed to maintain the size of the spatial dimensions, the colors are strictly illustrative.

The lifting of images or other spatial/temporal data to higher dimensional homogeneous spaces is a very general technique not

confined to SE(2). In Figure 5.7 the same idea is illustrated for the scaling and translation group ST(2) (recall § 2.3) where the input maps are convolved with scaled versions of the same kernel. Lifting to SIM(2) would work the same way by making a set of rotated and scaled versions of a kernel.

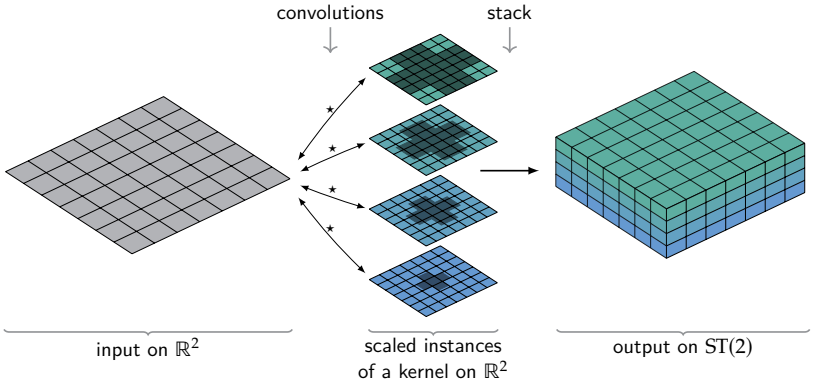


FIGURE 5.7: An illustration of how (discrete) feature maps on $\mathbb{R}^2$ are *lifted* to ST(2) in an equivariant manner. A set of scaled instances of the same $\mathbb{R}^2$ kernel is made and a convolution is performed on the input map with each instance. Not shown is the padding needed to maintain the size of the spatial dimensions, the colors are strictly illustrative.

Lifting has a rich mathematical history outside of its recent use in G-CNNs. We will cover this topic more in Chapter 9 when we use lifting outside the confines of neural networks.

LIFTING LAYER. We can incorporate the lifting operator by replacing the convolution operations in the first layer of a CNN. Say we have a set of feature maps $(u_i^{\text{in}})_{i=1}^{C_{\text{in}}} \subset \text{BC}(\mathbb{R}^2)$ and wish to generate a set of lifted feature maps $(u_i^{\text{out}})_{i=1}^{C_{\text{out}}} \subset \text{BC}(\mathbb{M}_2)$. We can do this by modifying (5.2) to use the lifting expression from (5.4) as follows:

$$u_j^{\text{out}}(x,\theta) = \sigma\left(\sum_{i=1}^{C_{\text{in}}} \left(\left((0,\theta)\cdot k_{ij}\right) \star u_i^{\text{in}}\right)(x) + b_j\right) \tag{5.5}$$

for $j = 1 \dots C_{\text{out}}$, where the kernels $k_{ij} \in C(\mathbb{R}^2) \cap L^2(\mathbb{R}^2)$ and biases $b_j \in \mathbb{R}$ are the trainable parts of the layer and $\sigma : \mathbb{R} \to \mathbb{R}$ is a scalar

activation function that is applied pointwise. Discretizing (5.5) yields the construction in Figure 5.8 for a 3-to-2 channel lifting layer.

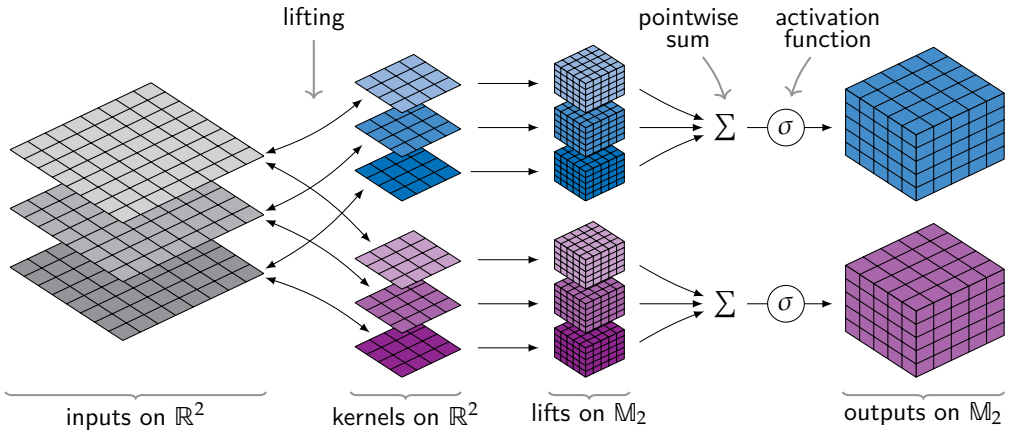


FIGURE 5.8: The architecture of a $\mathbb{R}^2$ (3 channel) to $\mathbb{M}_2$ (2 channel) *lifting layer* is based on the multi channel convolution layer from Figure 5.2 but instead of convolving the kernels with the inputs we perform a lift as per (5.3) and Figure 5.6. The 3-dimensional feature maps on $\mathbb{M}_2$ produced by this are summed up for each output channel and passed through an activation function to yield the output maps.

GROUP CONVOLUTION LAYER. Once the lifting layer has produced feature maps that live on $\mathbb{M}_2$ we can once again apply Theorem 3.3 to build an equivariant operator that works on feature maps on $\mathbb{M}_2$. Let the operator $A : \mathrm{BC}(\mathbb{M}_2) \to \mathrm{BC}(\mathbb{M}_2)$ be given by

$$(Af)(x,\theta) = \int_{\mathbb{R}^2} \int_0^{2\pi} ((x,\theta) \cdot k)(y,\beta) f(y,\beta) \, \mathrm{d}\beta \, \mathrm{d}y$$

for every $f \in \mathrm{BC}(\mathbb{M}_2)$, $(x,\theta) \in \mathbb{M}_2$ and some kernel $k \in C(\mathbb{M}_2) \cap B(\mathbb{M}_2)$. Since this operation works with functions on the group (i.e. the principal homogeneous space) instead of lower-dimensional

homogeneous spaces it is also referred to as SE(2) *group cross correlation* and denoted by

$$\left(k \star_{\mathrm{SE}(2)} f\right)(x,\theta) := \int_{\mathbb{R}^2}\int_0^{2\pi} k\left((x,\theta)^{-1}(y,\beta)\right) f(y,\beta)\,\mathrm{d}\beta\,\mathrm{d}y \tag{5.6}$$

or by inverting the kernel referred to as SE(2) *group convolution*:

$$\left(k *_{\mathrm{SE}(2)} f\right)(x,\theta) := \int_{\mathbb{R}^2}\int_0^{2\pi} k\left((y,\beta)^{-1}(x,\theta)\right) f(y,\beta)\,\mathrm{d}\beta\,\mathrm{d}y. \tag{5.7}$$

When the group we are working in is clear we drop the subscripts on the $\star$ and $*$ symbols.

In deep learning it is conventional to refer to both cross-correlation and convolution as simply convolution. Indeed, convolutional layers in neural networks generally implement the cross-correlation operation. We will distinguish the two when talking about the operation being performed, but in keeping with convention we will still refer to the networks and layers as *convolutional* regardless of whether the underlying implementation uses cross-correlation as in (5.6) or convolution as in (5.7).

We can now insert this operation into the single channel convolution layer from (5.1) to obtain a *single channel group convolution layer*. Let $C_{\text{in}}, C_{\text{out}} \in \mathbb{N}$ be the number of input respectively output channels of the layer and let $\left(u_i^{\text{in}}\right)_{i=1}^{C_{\text{in}}} \subset \mathrm{BC}(\mathbb{M}_2)$ be the input feature maps, then the output feature maps $\left(u_j^{\text{in}}\right)_{j=1}^{C_{\text{out}}} \subset \mathrm{BC}(\mathbb{M}_2)$ are given by

$$u_j^{\text{out}} = \sigma\left(\sum_{i=1}^{C_{\text{in}}} a_{ij}\left(k_i \star_{\mathrm{SE}(2)} u_i^{\text{in}}\right) + b_j\right) \tag{5.8}$$

for $j = 1 \dots C_{\text{out}}$, where $k_i \in L^1(\mathbb{M}_2) \cap C(\mathbb{M}_2)$ are the trainable kernels and $a_{ij}, b_j \in \mathbb{R}$ the trainable coefficients and biases for $i = 1 \dots C_{\text{in}}$ and $j = 1 \dots C_{\text{out}}$. Once we discretize (5.8) we get the layer architecture illustrated in Figure 5.9.

PROJECTING BACK DOWN. Generally we are not interested in higher-dimensional feature maps but in an output that is a processed image or some form of categorical classification. In other

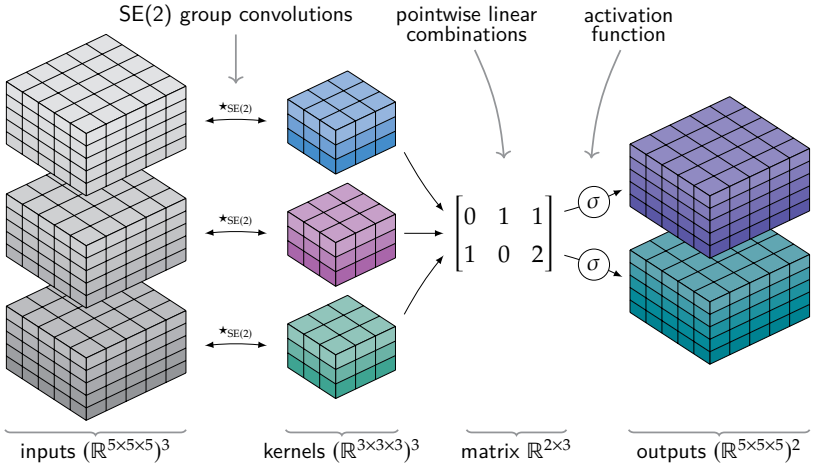


FIGURE 5.9: The architecture of an SE(2) equivariant $\mathbb{M}_2$ to $\mathbb{M}_2$ *group convolution layer* is based on the single channel convolution layer from Figure 5.1 except that the feature maps and kernels live on $\mathbb{M}_2$ and the conventional spatial convolution is replaced by the SE(2) group convolution.

words, we need to go back down to a lower-dimensional homogeneous space, either the original $\mathbb{R}^2$ in case we desire an image output or the trivial homogeneous space $\{0\}$ in the case of classification tasks. Going from higher to lower-dimensional homogeneous spaces is referred to as *projection*.

We can apply Theorem 3.3 once again to obtain an operator with trainable kernel. In practice, trainable projection operators are rarely used for two reasons. First, they are not necessary since if more network expressivity is required another group convolution layer can be inserted before the projection. Second, while lifting and group convolution layers have no restrictions on the kernel, in the case of projection the kernel compatibility condition (3.12) will not be trivial and so the kernel will require certain symmetries. This presents an added complication for implementation that can be avoided by fixing the projection operation.

Two variants are common for projecting a feature map on $\mathbb{M}_2$ to one on $\mathbb{R}^2$. The first is the linear projection that integrates over the orientations per

$$(P_{\text{lin}} f)(x) := \int_0^{2\pi} f(x, \theta) \, \mathrm{d}\theta \tag{5.9}$$

for any $f : \mathbb{M}_2 \to \mathbb{R}$ and all $x \in \mathbb{R}^2$. The second is the maximum projection that takes the maximum over the orientations per

$$(P_{\max} f)(x) := \max_{\theta \in [0, 2\pi)} f(x, \theta) \tag{5.10}$$

for any $f : \mathbb{M}_2 \to \mathbb{R}$ and all $x \in \mathbb{R}^2$. These projections are naturally SE(2) equivariant. Once we have conventional spatial feature maps once again we can use more conventional neural network operations to obtain the desired final output.

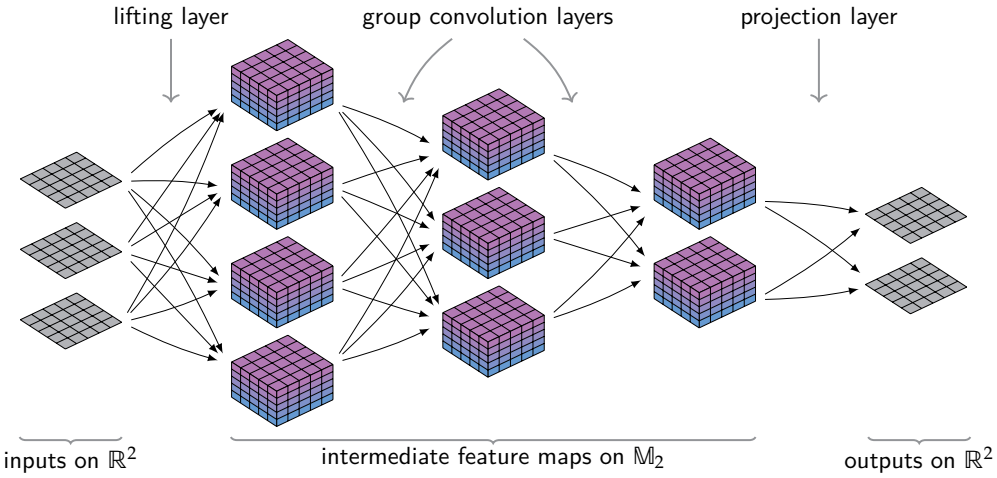


FIGURE 5.10: An example of an SE(2) equivariant CNN that takes in 3 two-dimensional feature maps on the left and outputs 2 two-dimensional feature maps on the right (say for segmenting an RGB image into two classes). The kernels of the lifting and group convolution layers have no symmetry requirements.

OVERALL ARCHITECTURE. The three layers we presented, the lifting, group convolution and projection layers, constitute the

core ingredients of a group equivariant convolutional neural network (G-CNN). Since all these layers perform only equivariant operations, a network built with them will be equivariant as a whole. This overall *lift-convolve-project* architecture is illustrated in Figure 5.10.

NON TRAINABLE LIFTING. Having a trainable lifting layer as in (5.5) and Figure 5.8 is not the only option for lifting the input of the network to a higher homogeneous space. Indeed the *orientation score transform* that we use in Chapter 9 can be used instead of a lifting layer to transform the input image to a set of feature maps on $\mathbb{M}_2$ directly. The actual neural network can then start straight away with group convolution layers. This approach was shown to work in BELLAARD, PAI, BESCOS, and DUITS (2023). In our work on PDE-based CNNs in the the following chapters we restrict ourselves to the trainable lifting layer.

5.3 POOLING

Another common operation in CNNs is *pooling*, which is used to coarsen the spatial discretization of feature maps going through the network. Pooling is conceptually similar to convolution in that a *window* slides over the input much like a kernel does in convolution. The difference is that instead of taking a weighted sum of the kernel elements with their corresponding input elements a fixed collating function is applied to the input elements in each window. This collation function typically consists of taking the maximum, the minimum or the average of the elements in the window. This type of operation is illustrated in Figure 5.11 for a conventional CNN.

POOLING IN THE CONTINUOUS SETTING. Pooling operations are rarely discussed outside of the discrete setting but they can be formulated in the continous setting as well. We can think of the conventional discrete pooling operation as actually doing two separate things:

1. replacing each element in the grid with a collation of elements in its window and

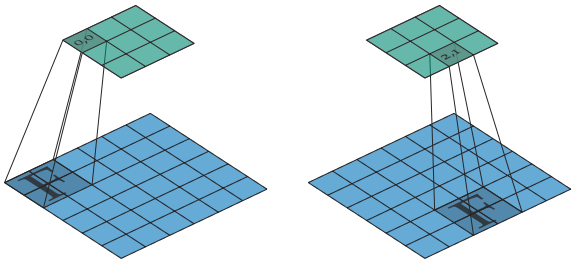


FIGURE 5.11: Pooling operations slide a window over the input feature maps and collate the elements inside the window through a choice of $F : \mathbb{R}^{\#\text{window elements}} \to \mathbb{R}$. Usually F is taking the maximum, minimum or average and the stride of the window is chosen to equal the size of the window.

2. subsampling the grid.

The second operation is one that only makes sense in the discrete setting, but the first can also be formulated in the continous setting and captures the essence of the pooling operation.

This logical splitting of 'strided' pooling into a non-strided pooling step and a subsampling step was also proposed in COHEN (2021, § 2.4.3).

Staying in the $\mathbb{R}^n$ setting, let us define a *window* as a compact subset $W \subset \mathbb{R}^n$, possibly but not necessarily centered around the origin and having a single component. A *maximum* and *minimum pooling* operation with window W on a function $f : \mathbb{R}^n \to \mathbb{R}$ can then be formulated as

$$\begin{aligned} \text{MaxPool}(f, W)(x) &:= \sup_{y \in W} f(x + y), \\ \text{MinPool}(f, W)(x) &:= \inf_{y \in W} f(x + y). \end{aligned} \tag{5.11}$$

An *average pooling* operation can be defined as

$$\text{AvgPool}(f, W)(x) := \frac{1}{\text{vol}(W)} \int_W f(x + y) \, \mathrm{d}y. \tag{5.12}$$

If we let the window W be some hypercube around the origin, say $W = [-D, D]^n$ for some $D > 0$, then we have continuous operations that will match the familiar pooling operations from Figure 5.11 once discretized and combined with downsampling of the output.

Observe how these pooling operations are translation equivariant, i.e. for all $z \in \mathbb{R}^n$

$$\begin{aligned}\text{MaxPool}(p \mapsto f(p+z), W)(x) &= \sup_{y \in W} f(x+y+z) \\ &= \text{MaxPool}(f, W)(x+z)\end{aligned}$$

and similarly for the other pooling variants.

We can now wonder if we can formulate these type of pooling operations on homogeneous spaces in a way that makes them equivariant. This was done by COHEN (2021, §2.4.3) for principal homogeneous spaces, i.e. the groups themselves, but not for homogeneous spaces in general.

EQUIVARIANT MAX POOLING. We first generalize maximum and minimum pooling to homogeneous spaces. Let M be a homogeneous space of a Lie group G with reference element p_0. Let the window of the pooling operator W now be a compact subset of the group, potentially but not necessarily consisting of a single component containing the unit element.

If M was the principal homogeneous space, i.e. $M = G$, then we could proceed as per COHEN and WELLING (2016, §6.3) and COHEN (2021, §2.4.3) and define

$$\text{MaxPool}(f, W)(p) := \sup_{g \in pW} f(g) \tag{5.13}$$

for any $f : M \to \mathbb{R}$. This construction does not directly translate to non-principal homogeneous spaces and will require some work in the same vein as we saw for Theorem 3.3.

We start by modifying (5.13) by picking a $g_p \in G$ that satisfies $g_p \cdot p_0 = p$ for every $p \in M$, i.e. $g_p \in G_{p_0 \to p}$. Then we could write the maximum pooling of a function $f : M \to \mathbb{R}$ with window $W \subset G$ as

Recall that $G_{p_0 \to p}$ is the set of group elements that map p_0 to p as per (2.6).

$$\text{MaxPool}(f, W)(p) := \sup_{g \in g_p W} f(g \cdot p_0). \tag{5.14}$$

This properly generalizes (5.13) since if $M = G$ and we let $p_0 = e$ then $g_p = p$ and $g \cdot p_0 = g$. The problem here is well-posedness, the result should not depend on our choice of $g_p \in G_{p_0 \to p}$. Take two group elements $g_p, g_p' \in G_{p_0 \to p}$ then there exists an $h \in G_{p_0}$

Recall that G_{p_0} is the isotropy group of p_0 as per (2.5).

so that $g'_p = g_p h$, consequently

$$\sup_{g \in g'_p W} f(g \cdot p_0) = \sup_{g \in g_p h W} f(g \cdot p_0) \stackrel{?}{=} \sup_{g \in g_p W} f(g \cdot p_0)$$

should hold, but the last equality only holds if $hW = W$. Thus, in general, (5.14) is only well-posed if

$$\forall h \in G_{p_0} : hW = W, \tag{5.15}$$

or in words: the window should be invariant under the action of the stabilizer of the reference element p_0. It is no accident that this condition on the window is essentially the same as the condition on the kernel (3.12) that we found was required for equivariant linear operators to be well-posed in Theorem 3.3.

We can now check with $W \subset G$ satisfying (5.15) that (5.14) is indeed equivariant. Let $h \in G$ and $f : M \to \mathbb{R}$ then

$$\begin{aligned}
\operatorname{MaxPool}(h \cdot f, W)(p) &= \sup_{g \in g_p W} f(h^{-1} g \cdot p_0) \\
&= \sup_{h g' \in g_p W} f(g' \cdot p_0) \\
&= \sup_{g' \in h^{-1} g_p W} f(g' \cdot p_0) \\
&= \sup_{g' \in g_{h^{-1}p} W} f(g' \cdot p_0) \\
&= (h \cdot \operatorname{MaxPool}(f, W))(p),
\end{aligned}$$

from which we conclude that $\operatorname{MaxPool}(\cdot, W) : B(M) \to B(M)$ is indeed an equivariant operator. Naturally, equivariant minimum pooling works the same way except we take the infimum instead of the supremum.

MORPHOLOGICAL CONVOLUTION. Observe that (5.14) can also be written as

$$\begin{aligned}
\operatorname{MaxPool}(f, W)(p) &= \sup_{q \in g_p W \cdot p_0} f(q) \\
&= \sup_{q \in M} f(q) + \iota_{W \cdot p_0}(g_p^{-1} \cdot q)
\end{aligned} \tag{5.16}$$

where we define the *morphological indicator function* $\boldsymbol{\iota}_S : M \to \{0, \infty\}$ as

$$\boldsymbol{\iota}_S(p) := \begin{cases} 0 & \text{if } p \in S, \\ -\infty & \text{else.} \end{cases} \tag{5.17}$$

We used the morphological indicator function $\boldsymbol{\iota}$ here in the same way we use the conventional indicator function $\mathbb{1} : M \to \{0, 1\}$ to say something like

$$\int_S f(p)\,\mathrm{d}p = \int_M f(p)\,\mathbb{1}_S(p)\,\mathrm{d}p$$

for some $S \subset M$.

The benefit of writing maximum (or minimum) pooling as in (5.16) is that is reveals a convolutional structure similar to Theorem 3.3 if one replaces the integral by the supremum and replaces multiplying the function with the kernel by adding them to each other. Indeed if one defines *morphological convolution* of a kernel $k : M \to \mathbb{R} \cup \{\infty\}$ and function $f : M \to \mathbb{R}$ by

$$(k \square_M f)(p) := \inf_{q \in M} (g_p \cdot k)(q) + f(q)$$

(assuming appropriate symmetries on k analogous to (5.15) for well-posedness), then one can also define maximum and minimum pooling by

$$\begin{aligned} \operatorname{MaxPool}(f, W) &:= -\left(\boldsymbol{\iota}_{W \cdot p_0} \square_M (-f)\right) \\ \operatorname{MinPool}(f, W) &:= \boldsymbol{\iota}_{W \cdot p_0} \square_M f. \end{aligned}$$

We will come back to the topic of morphological convolutions in the next chapter.

AVERAGE POOLING. Average pooling can rely on our previous results from Theorem 3.3 since it is a linear operator. Let M admit an invariant integral and measure μ. We define the kernel we will use as the normalized indicator function of $W \cdot p_0$:

$$\kappa_W(p) := \frac{1}{\mu(W \cdot p_0)} \mathbb{1}_{W \cdot p_0}(p),$$

which is in $L^1(M)$ if W is compact. Then we define the average pooling operator $\mathrm{AvgPool}(\cdot, W) : \mathrm{BC}(M) \to \mathrm{BC}(M)$ as

$$\begin{aligned}\mathrm{AvgPool}(f, W)(p) &:= \frac{1}{\mu(W \cdot p_0)} \int_M \mathbb{1}_{W \cdot p_0}(g_p^{-1} \cdot q)\, f(q)\, \mathrm{d}\mu(q) \\ &= \frac{1}{\mu(W \cdot p_0)} \int_M \mathbb{1}_{g_p W \cdot p_0}(q)\, f(q)\, \mathrm{d}\mu(q) \\ &= \frac{1}{\mu(W \cdot p_0)} \int_{g_p W \cdot p_0} f(q)\, \mathrm{d}\mu(q).\end{aligned} \tag{5.18}$$

The kernel κ_W is not continuous so we do not strictly satisfy the conditions of Theorem 3.3 but as long as W is compact $\mathrm{AvgPool}(\cdot, W)$ is still a bounded linear operator on $\mathrm{BC}(M)$. We verify equivariance since we cannot technically apply Theorem 3.3. Let $g \in G$ then

$$\begin{aligned}&\mathrm{AvgPool}(g \cdot f, W)(p) \\ &\quad= \frac{1}{\mu(W \cdot p_0)} \int_M \mathbb{1}_{g_p W \cdot p_0}(q)\, f(g^{-1} \cdot q)\, \mathrm{d}\mu(q) \\ &\quad= \frac{1}{\mu(W \cdot p_0)} \int_M \mathbb{1}_{g^{-1} g_p W \cdot p_0}(g^{-1} \cdot q)\, f(g^{-1} \cdot q)\, \mathrm{d}\mu(q) \\ &\quad= \frac{1}{\mu(W \cdot p_0)} \int_M \mathbb{1}_{g^{-1} g_p W \cdot p_0}(q)\, f(q)\, \mathrm{d}\mu(q) \\ &\quad= \frac{1}{\mu(W \cdot p_0)} \int_M \mathbb{1}_{g_{g^{-1}.p} W \cdot p_0}(q)\, f(q)\, \mathrm{d}\mu(q) \\ &\quad= \frac{1}{\mu(W \cdot p_0)} \int_M \mathbb{1}_{g_{g^{-1}.p} W \cdot p_0}(q)\, f(q)\, \mathrm{d}\mu(q) \\ &\quad= \big(g \cdot \mathrm{AvgPool}(f, W)\big)(p),\end{aligned}$$

since f and g were arbitrary we conclude that $\mathrm{AvgPool}(\cdot, W)$ is indeed equivariant.

We have now generalized all traditional CNN operators to equivariant counterparts in the continous setting. These generalized operators are the first step towards developing PDE-based equivariant neural networks.

6

PDE-BASED CONVOLUTIONAL NEURAL NETWORKS

In this chapter we introduce *PDE-based group CNNs* or *PDE-G-CNNs* for short. Built on the group CNNs from the previous Chapter 5 we present a PDE-based framework that replaces the usual CNN trifecta of convolution, pooling and activation function with a Hamilton-Jacobi type evolution PDE, or rather a numerical solver for such a PDE.

Formulating these PDEs on homogeneous spaces allows us to retain the equivariance properties of G-CNNs while adding an additional prior in the form of PDE-based data processing. We will work out the specifics of this approach for our primary case of interest: roto-translation equivariance.

This chapter is based on the initial work in Smets, Portegies, Bekkers, and Duits (2022) which was followed up on by Bellaard, Bon, Pai, Smets, and Duits (2023), Pai, Bellaard, Smets, and Duits (2023) and Bellaard, Pai, Bescos, and Duits (2023).

6.1 INSPIRATION

KERNELS AS GREEN'S FUNCTIONS. Convolutional neural network already carry a certain resemblance to numerical PDE solvers. Consider how many inhomogeneous linear differential equations are solvable by convolving the initial value with a Green's function. For example, in the $\mathbb{R}^n$ setting the diffusion equation

$$\begin{cases} \partial_t u(\cdot, t) = \Delta u(\cdot, t) \\ u(\cdot, 0) = u_0 \end{cases}$$

is solved by convolution with the Gaussian kernel G_t as $u(\cdot, t) = G_t * u_0$.

This means that convolutional layers in neural network as in Figure 5.1 and 5.9 are already closely related to numerical solvers of linear PDEs. The 'solvers' learned are restricted in the sense that they are necessarily linear and limited by the generally small discrete size of the kernels used in neural network. On the other hand these 'solvers' are unrestricted in the sense that there is no structure imposed on the kernel outside it discrete size and so no particular control over what type of PDEs (if any) can be learned.

POOLING AS A MORPHOLOGICAL PDE. Another analogue can be found between maximum and minimum pooling in neural network and morphological PDEs. In mathematical morphology one is concerned with the analysis of shapes, for an overview see SERRA and VINCENT (1992). The two fundamental operations in mathematical morphology are *dilation* and *erosion*. They roughly correspond to maximum respectively minimum pooling with a variable sized window. Dilation and erosion relate to the Hamilton-Jacobi initial value problem on $\mathbb{R}^n$:

$$\begin{cases} \partial_t u(\cdot, t) = \pm \mathcal{H}(\mathrm{d}u(\cdot, t)), \\ u(\cdot, 0) = u_0(\cdot), \end{cases}$$

corresponding to dilation with $(+)$ and erosion with $(-)$. Under appropriate conditions on the Hamiltonian $\mathcal{H}|_x : T_x^*\mathbb{R}^n \to \mathbb{R}$ (EVANS, 2022, §3.3.2) the solution is given by the *Hopf-Lax formula*

$$u(x, t) = \inf_{y \in \mathbb{R}^n} \left\{ u_0(y) + t\mathcal{L}\left(\frac{x - y}{t}\right) \right\}, \tag{6.1}$$

where $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ is the Lagrangian associated to the Hamiltonian $\mathcal{H}$. We can write minimum pooling (5.11) in a similar manner as

$$\begin{aligned} \mathrm{MinPool}(f, W)(x) &= \inf_{y \in W} f(x + y) \\ &= \inf_{y \in \mathbb{R}^n} \left(f(x + y) + k_W(y) \right) \\ &= \inf_{y \in \mathbb{R}^n} \left(f(y) + k_W(y - x) \right) \end{aligned}$$

where we let

$$k_W(y) := \begin{cases} 0 & \text{if } y \in W, \\ \infty & \text{else.} \end{cases}$$

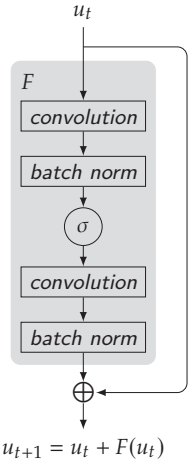
These types of PDEs have a rich history in image processing (VAN DEN BOOMGAARD and SMEULDERS, 1994; BREUß and WEICKERT, 2009; MARAGOS, 2005; SOILLE et al., 1999) and their appearance in disguise in classical CNNs is encouraging towards our approach.

RESIDUAL CONNECTIONS AS EULER STEPS. Additive *residual connections*, where the input of a neural network's layer (or set of layers) is added to its output have become ubiquitous (LIU et al., 2022; WOO et al., 2023) in CNNs since their introduction by HE, ZHANG, REN, and SUN (2016). See the margin figure for an illustration. An additive residual is reminiscent of solving an evolution equation $\partial_t u = F(u)$ using the forward Euler method where one iteratively computes $u(t + \Delta t) = u(t) + \Delta t\, F(u(t))$. This connection is is further explored by linking neural networks with *ordinary differential equations* (CHEN, RUBANOVA, BETTENCOURT, and DUVENAUD, 2018; KIDGER, 2022; SANDER, ABLIN, and PEYRÉ, 2022).

6.2 RELATED WORK

DOMAIN EXTENSION. With G-CNNs being the starting point for developing PDE-G-CNNs we take a domain extension viewpoint where input feature maps are lifted to higher homogeneous spaces as in Figure 5.6 or 5.7. This idea previously originated from coherent state theory (ALI, ANTOINE, GAZEAU, et al., 2000), orientation score theory (DUITS, 2005; DUITS, DUITS, VAN ALMSICK, and FLORACK, 2004), cortical perception models (CITTI and SARTI, 2006) and rigid-body motion scattering (SIFRE, 2014; SIFRE and MALLAT, 2014).

We will mainly be concerned with the Lie group SE(2) and its principal homogeneous space $\mathbb{M}_2$ but other Lie groups/homogeneous spaces have been used as extended domains such as the Heisenberg groups (BARBIERI, CITTI, COCCI, and SARTI, 2014; DUITS, FÜHR, et al., 2013), SIM(2) (BASPINAR, 2018) and SE(3)/SO(2) (BOLELLI, 2023; PORTEGIES, 2018) for example.



Residual connections (HE, ZHANG, REN, and SUN, 2016) (above) are similar to a step of the forward Euler method.

PROBABILISTIC-CNNS. Our geometric PDEs relate to α-stable Lévy processes (DUITS, BEKKERS, and MASHTAKOV, 2019) and cost-processes akin to (AKIAN, QUADRAT, and VIOT, 1994), but then on the homogeneous spaces like $\mathbb{M}_d$ rather than the usual $\mathbb{R}^d$. This relates to probabilistic equivariant numerical neural networks (FINZI, BONDESAN, and WELLING, 2020) that use anisotropic convection and diffusion on $\mathbb{R}^d$.

In contrast to these networks, the PDE-G-CNNs that we propose allow for *simultaneous* spatial and angular diffusion on $\mathbb{M}_d$. Furthermore we include nonlinear Bellman processes (AKIAN, QUADRAT, and VIOT, 1994) for maximum and minimum pooling over Riemannian balls on $\mathbb{M}_2$.

KERCNNS. An approach to introducing horizontal connectivity in CNNs that does not require a Lie group structure was proposed by MONTOBBIO, BONNASSE-GAHOT, CITTI, and SARTI (2019) and MONTOBBIO, SARTI, and CITTI (2020) in the form of KerCNNs. In this biologically inspired metric model a diffusion process is used to achieve intra-layer connectivity.

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

DIFFERENTIAL EQUATIONS AND NEURAL NETWORKS. The connection between neural networks and differential equations became widely known in 2017, when WEINAN (2017) explicitly explained the connection between neural networks and dynamical systems especially in the context of very deep ResNets (HE, ZHANG, REN, and SUN, 2016). This point of view was further expanded by LU, ZHONG, LI, and DONG (2018), showing how many very deep neural networks can be viewed as discretizations of ordinary differential equations.

The somewhat opposite point of view was taken by CHEN, RUBANOVA, BETTENCOURT, and DUVENAUD (2018), who introduced a new type of neural network which no longer has discrete layers, replacing them by a field parameterized by a continuous time variable. Weinan E also indicated a relationship between CNNs and PDEs, or rather with evolution equations involving a nonlocal operator. Implicitly, the connection between neural networks and

differential equations was also explored by the early works of Chen, Yu, and Pock (2015) who learn parameters in a reaction-diffusion equation. This connection between neural networks and PDEs was then made more explicit by Long, Lu, Ma, and Dong (2018) who made it possible to learn a much wider class of PDEs with their PDE-Net. More recent work in PDE inspired neural networks includes Ruthotto and Haber (2020) and Shen, He, Lin, and Ma (2020).

Basing neural network computations on PDEs formulated on manifolds also makes the processing independent with respect to the choice of coordinates on the manifold in the fashion of Weiler, Forré, Verlinde, and Welling (2021).

More recent work in this direction includes integrating equivariant partial differential operators in steerable CNNs (Jenner and Weiler, 2021), drawing a strong analogy between deep learning and physics.

A useful aspect of the connection between neural networks and differential equations is the observation that the stability of the differential equation can give insight into the stability and generalization ability of the neural network (Haber and Ruthotto, 2017). Moreover, there are intriguing analogies with numerical PDE approximations and specific network architectures (e.g. ResNets), as can be seen in the comprehensive overview article by Alt, Peter, Weickert, and Schrader (2021).

6.3 THE PDE IN QUESTION

OUR APPROACH. Given the similarities noted in §6.1 we propose to make the link between CNNs and PDEs more explicit by choosing a PDE as a *starting point* for building a neural network architecture. The PDE we propose to use for this purpose comes from the geometric image analysis world (Burger, Sawatzky, and Steidl, 2016; Dubrovina-Karni, Rosman, and Kimmel, 2014; Duits and Burgeth, 2007; Duran, Moeller, Sbert, and Cremers, 2016; Morel and Solimini, 2012; Peyré, Péchaud, Keriven, Cohen, et al., 2010; Sapiro, 2006; Sethian et al., 1999; Weickert, 1996; Weickert, Grewenig, Schroers, and Bruhn, 2016; Welk and Weickert, 2019). It was chosen based on the following criteria:

1. it exhibits similar behaviour to a CNN's existing convolution, pooling and activation operations,
2. it can be formulated on general homogeneous spaces and not just $\mathbb{R}^n$,
3. a numerical solver for it can be efficiently implemented on modern parallel hardware and integrated in existing deep learning frameworks.

Let M be a homogeneous space of some Lie group G, then we propose the following evolution equation:

$$\begin{cases} \dfrac{\partial u}{\partial t}(p,t) = & -\boldsymbol{c}_p\, u(\cdot,t) & \text{(convection)} \\ & -(-\Delta_{\mathcal{G}_1})^{\nu}_p\, u(\cdot,t) & \text{(diffusion)} \\ & +\frac{1}{\alpha}\left\| \left(\nabla_{\mathcal{G}_2^+} u(\cdot,t)\right)_p \right\|^{\alpha}_{\mathcal{G}_2^+} & \text{(dilation)} \\ & -\frac{1}{\alpha}\left\| \left(\nabla_{\mathcal{G}_2^-} u(\cdot,t)\right)_p \right\|^{\alpha}_{\mathcal{G}_2^-} & \text{(erosion)} \\ & & \text{for all } p \in M,\ t \geq 0, \\ u(p,0) = u_0(p) & & \text{for all } p \in M. \end{cases} \tag{6.2}$$

where $\boldsymbol{c} \in \mathfrak{X}(M)$ is a smooth vector field, $\mathcal{G}_1, \mathcal{G}_2^+, \mathcal{G}_2^- \in \Gamma(T^{(0,2)}M)$ are metric tensor fields, $0 < \nu \leq 1$, $\alpha > 1$ are smoothness parameters[1] and u is the viscosity solution (Evans, 2022, §10.1; Strömberg, 2002). The smoothness parameters we choose as some constants, we will come back to them later. That leaves $\{\boldsymbol{c}, \mathcal{G}_1, \mathcal{G}_2^+, \mathcal{G}_2^-\}$ as the actual trainable parts of this PDE, each of which controls one of the four terms of the generator. To 'train' these tensor fields we will need to parametrize them in some fashion so that we can train the actual scalar parameters. Demanding equivariance will severely restrict the degrees of freedom available to us and and

The fractional diffusion parameter ν can in principle be given a value larger than 1 but then the associated heat-kernel is no longer guaranteed to be non-negative, see Grigor'yan, Hu, and Lau (2014, §1.2.2).

1 In Smets, Portegies, Bekkers, and Duits (2022) we adopted a different convention for the smoothness parameters. Here we separate the smoothness parameters for diffusion and dilation/erosion and use the $1/\alpha$ factor and α exponent from Bellaard, Bon, Pai, Smets, and Duits (2023, §3) instead of the less convenient 2α exponent.

yield natural parametrizations with a limited number of parameters, we will detail this later in the chapter. As labeled in (6.2) these four terms are called the *convection*, *diffusion*, *dilation* and *erosion* terms. Convection can be understood as *transporting* data, diffusion as *regularization* and dilation and erosion as a 'soft' form of *max* respectively *min pooling*.

EQUIVARIANT GENERATOR. Naturally we would like the evolution caused by (6.2) to be equivariant in the sense that if we have two solutions to the PDE $u_1, u_2 \in M \times \mathbb{R}_{\geq 0}$ with initial conditions that are related as per $u_2(\cdot, 0) = g \cdot u_1(\cdot, 0)$ for some $g \in G$ then we would like $u_2(\cdot, t) = g \cdot u_1(\cdot, t)$ for all $t \geq 0$. Or in words: applying a group transform on the initial conditions and then evolving yields the same result as first evolving and then applying the group transform.

For the convection generator we want that

$$-\boldsymbol{c}_p(g \cdot f) = -\left(g \cdot (\boldsymbol{c} f)\right)(p),$$

or stated equivalently by labeling the action of G on M by $\rho_g(p) = g \cdot p$:

$$\boldsymbol{c}_p(f \circ \rho_{g^{-1}}) = \boldsymbol{c}_{g^{-1} \cdot p} f$$

for all $f \in C^1(M)$ and $g \in G$. This only holds if

$$\left(\rho_g\right)_* \boldsymbol{c} = \boldsymbol{c},$$

for all $g \in G$, i.e. $\boldsymbol{c}$ needs to be a G-invariant vector field. In the principal case $M = G$ these are the left-invariant vector fields which form a vector space with the same number of dimensions as M, in the more general case the space of G-invariant vector fields will have fewer or equal number of dimensions than M.

The equivariance of the three remaining terms depends on their associated metrics $\mathcal{G}_1$, $\mathcal{G}_2^+$ and $\mathcal{G}_2^-$, which will need to be G-invariant as well for the generator to be equivariant. This again restricts the degrees of freedom of the metrics to a finite number. In the principal case $M = G$ where M is n-dimensional we have $\frac{1}{2}n(n+1)$ free parameters. In the more general case of lower dimensional homogeneous spaces we have less than that. In fact, recalling §2.4, such a Riemannian metric need not always exists.

We could for example use the unconstrained parameterization of symmetric positive definite matrices by PINHEIRO *and* BATES *(1996).*

OPERATOR SPLITTING. Our PDE solver takes an *operator splitting* approach. We split numeric timesteps into substeps, each of which only evolves the data according to a single term at a time. Let $u_0 : \mathbb{R}^2 \to \mathbb{R}$ be the initial condition (of appropriate regularity) and say we want to compute $u(\cdot, \Delta t)$, i.e. the solution to (6.2) at time Δt. Then we would first solve

$$\frac{\partial u_1}{\partial t}(p,t) = -\boldsymbol{c}_p \, u(\cdot, t)$$

with initial condition $u(\cdot, 0) = u_0$ with a timestep of Δt, i.e. we compute $u_1(\cdot, \Delta t)$. We take this solution as the initial condition for a PDE with the next term:

$$\frac{\partial u_2}{\partial t}(p,t) = -\left(-\Delta_{\mathcal{G}_1}\right)^{\nu} u(p,t)$$

with $u_2(\cdot, 0) = u_1(\cdot, \Delta t)$ and then compute $u_2(\cdot, \Delta t)$. Moving on we apply dilation per

$$\frac{\partial u_3}{\partial t}(p,t) = +\frac{1}{\alpha} \left\| \nabla_{\mathcal{G}_2^+} u_3(p,t) \right\|_{\mathcal{G}_2^+}^{\alpha}$$

with initial condition $u_3(\cdot, 0) = u_2(\cdot, \Delta t)$ and compute $u_3(\cdot, \Delta t)$. Finally we apply erosion per

$$\frac{\partial u_4}{\partial t}(p,t) = -\frac{1}{\alpha} \left\| \nabla_{\mathcal{G}_2^-} u_4(p,t) \right\|_{\mathcal{G}_2^-}^{\alpha}$$

with initial condition $u_4(\cdot, 0) = u_3(\cdot, \Delta t)$ and compute $u_4(\cdot, \Delta t)$. We then take $u_4(\cdot, \Delta t)$ as an approximation to the true solution $u(\cdot, \Delta t)$.

The order of of the substeps used is not essential, the order presented above matches the order we will cover each term in this chapter.

6.4 INTERPRETATION

Before going into the technical details of solving (6.2) in general, we first want to get a feeling for what it does in a familiar setting so that we can better interpret it in a more general setting later. In this section we will therefor have a look at the effects of the four terms of the PDE in the familiar $\mathbb{R}^2$ setting and how they are solved there, see Figure 6.1 for an illustration.

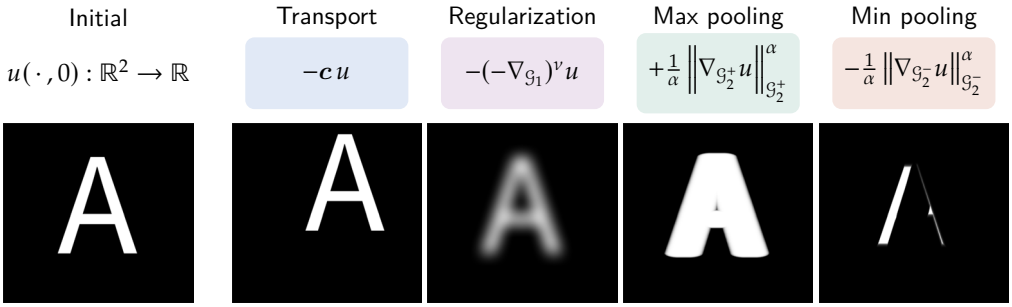


FIGURE 6.1: Geometric interpretation of the terms of (6.2) illustrated for the translation equivariant case on $\mathbb{R}^2$. The effect of each of the four terms of (6.2) separately on the initial value on the left is demonstrated.

TRANSPORT. Consider the translation equivariant setting $G = M = \mathbb{R}^2$. The left-invariant (i.e. translation invariant) vector fields on $\mathbb{R}^2$ are of the form

$$c = c^1 \partial_{x^1} + c^2 \partial_{x^2}$$

for any $c^1, c^2 \in \mathbb{R}$ where x^1, x^2 are the usual $\mathbb{R}^2$ coordinates. In this case the solution to (6.2) with just the first term $\partial_t u = -cu$ is accomplished by

$$u(p, t) = u_0\left(p - t\,(c^1, c^2)^T\right),$$

which has the effect of *transporting* the input along the vector field c, this is illustrated in the second image of Figure 6.1.

As this transport along c just amounts to a time-dependent translation of the initial value it can be readily implemented for discrete feature maps by resampling the grid with appropriate interpolation.

INVARIANT METRICS. For the other three terms we require a translation invariant metric. On $\mathbb{R}^2$ these are of the form

$$w_{11}\,\mathrm{d}x^1 \otimes \mathrm{d}x^1 + w_{12}\,\mathrm{d}x^1 \otimes \mathrm{d}x^2 + w_{21}\,\mathrm{d}x^2 \otimes \mathrm{d}x^1 + w_{22}\,\mathrm{d}x^2 \otimes \mathrm{d}x^2$$

so that the Gram matrix

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \in \mathbb{R}^{2\times 2}$$

is symmetric positive definite.

REGULARIZATION. Given a translation-invariant metric $\mathcal{G}_1$ the (fractional) diffusion PDE $\partial_t u = -\left(-\nabla_{\mathcal{G}_1}\right)^\nu u$ is solved (PODLUBNY, 1998) by convolution with a kernel $K_t^\nu \in C^\infty((0,\infty)\times\mathbb{R}^2)$ as per

$$u(p,t) = \left(K_t^\nu * u_0\right)(p),$$

for any $p \in M$, $t > 0$. When $\nu = 1$, we revert to the normal diffusion equation in which case the kernel is simply an appropriately shaped Gaussian. The effect produced is one of *regularization* of the initial feature map as can be seen in the third image of Figure 6.1.

Regardless of the choice of ν in this setting, the kernel K_t^ν will be a function of just t and the Riemannian distance $d_{\mathcal{G}_1}$ induced by the metric $\mathcal{G}_1$, i.e. just the skewed Euclidean norm. Specifically, there exists a $\tilde{K}_t^\nu : (0,\infty)\times[0,\infty) \to \mathbb{R}$ so that

$$K_t^\nu(p) = \tilde{K}_t^\nu\left(\mathrm{d}_{\mathcal{G}_1}(p,0)\right) = \tilde{K}_t^\nu\left(\|Ap\|\right),$$

for some invertible $A \in \mathbb{R}^2$ where $\|\cdot\|$ is the Euclidean norm. We will get back to fractional diffusion on homogeneous spaces in §6.7.

POOLING. The remaining two morphological terms are solved in a manner similar to the diffusion term but using morphological convolution instead of linear convolution. Let the metrics $\mathcal{G}_2^+$ and $\mathcal{G}_2^-$ be translation invariant. Then the PDE

Recall we also discussed max pooling as a morphological convolution in (5.16).

$$\partial_t u = \frac{1}{\alpha}\left\|\nabla_{\mathcal{G}_2^+} u\right\|_{\mathcal{G}_2^+}^\alpha,$$

with initial condition $u(\cdot\,,0) = u_0$ can be solved by the *Hopf-Lax* formula (EVANS, 2022, §10.3) as

$$u(p,t) = \sup_{q\in\mathbb{R}^2}\left\{u_0(q) - k_t^\alpha(p-q)\right\},$$

where the kernel $k^\alpha : \mathbb{R}_{>0}\times\mathbb{R}^2 \to \mathbb{R}\cup\{\infty\}$ depends on the choice of α. We will detail the kernel used later in §6.8. The point we would like to make now is that the way the solution is obtained is very similar to our previous formulation of continuous max pooling in (5.16). The effect obtained is indeed very reminiscent to max pooling as can be seen in the 4th panel of Figure 6.1. Additionally, for a given α and time t the kernel only depends on the

The solution concept used here is that of a viscosity solution as per EVANS (2022, §10.1).

distance between p and q, just like in the case of diffusion on $\mathbb{R}^2$ above, and so is essentially a scalar function of the distance.

The erosion PDE

$$\partial_t u = -\frac{1}{\alpha} \left\| \nabla_{\mathcal{G}_2^-} \right\|^\alpha$$

with initial condition $u(\cdot, 0) = u_0$ is solved similarly by

$$u(p,t) = \inf_{q \in \mathbb{R}^2} \left\{ u_0(q) + k_t^\alpha(p-q) \right\},$$

with the same kernel k^α as before. The effect produced is similar to that of min pooling as can be seen in the last panel of Figure 6.1.

GENERALIZATION POTENTIAL. The advantage of generating these effects through (6.2) is that we can now generalize them to other homogeneous spaces of our choice. Indeed, we can take as Lie group $G = \mathrm{SE}(2)$ and homogeneous space $M = \mathbb{M}_2$ and walk through this section again and get roto-translation equivariant transport, regularization and pooling from the same PDE. The result is illustrated in Figure 6.2, where we now obtain SE(2)-equivariant processing of feature maps that live on $\mathbb{M}_2$.

Compare and contrast Figure 6.2 with the previous Figure 6.1; the effects obtained are similar but distinct. In Figure 6.1 one can imagine that the effects applied to the 'A' would indeed commute with translations but not with rotations. In Figure 6.2 one can imagine that the effects on the 'A' commute with both translations and rotations.

Now that we have a high-level understanding of what the PDE (6.2) does we can start looking at the technical details.

6.5 CONVECTION

As we will be calculating approximate solutions to (6.2) using operator splitting we tackle each term separately, starting with the convection term in the general case of the homogeneous space $M = G/H$. With only the convection term we are left with the following PDE:

$$\begin{cases} \frac{\partial u_1}{\partial t}(p,t) = -\boldsymbol{c}_p \, u_1(\cdot, t) & \text{for all } p \in M, t \geq 0, \\ u_1(p,0) = u_0(p) & \text{for all } p \in M, \end{cases} \tag{6.3}$$

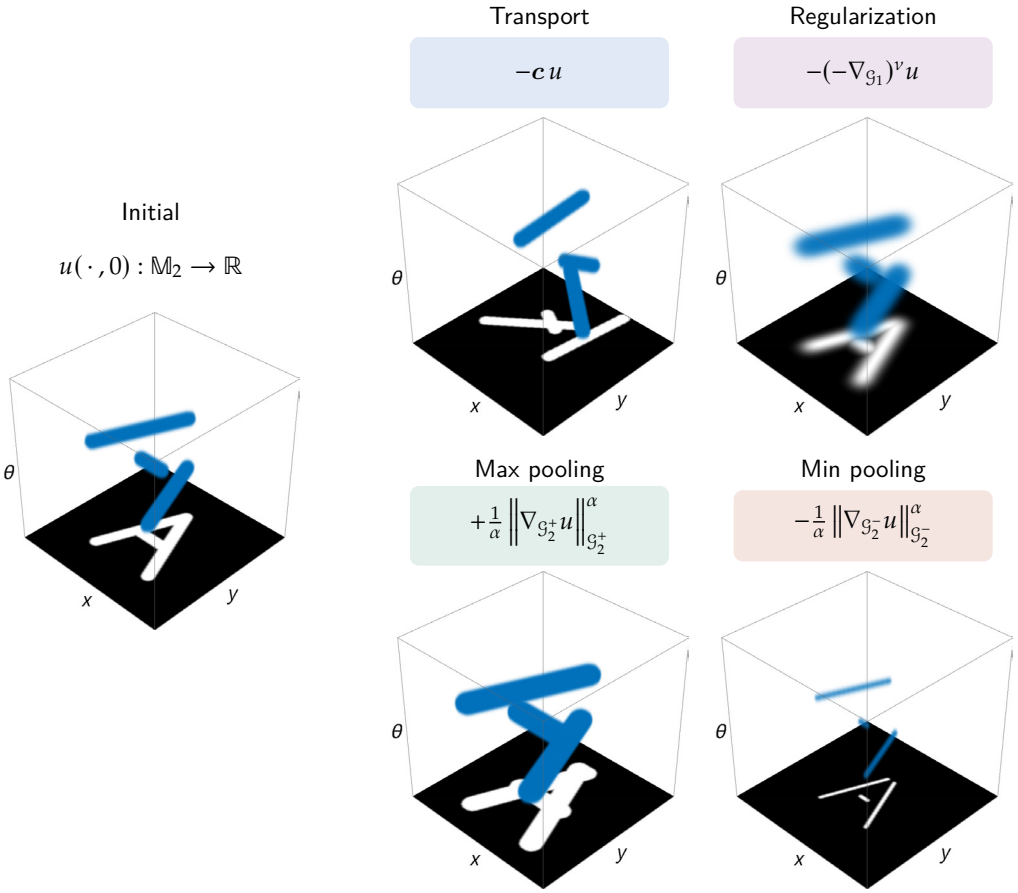


FIGURE 6.2: Geometric interpretation of the terms of (6.2) illustrated for the SE(2)-equivariant case on $\mathbb{M}_2$. We made an idealized lift of the letter 'A' to $\mathbb{M}_2$ (shown in blue), each of the three strokes of the letter is placed at its corresponding orientation. For clarity we also added a maximum projection to the base plane (shown in black and white). The effect of each of the four terms of (6.2) separately on the initial value on the left is demonstrated.

where $\boldsymbol{c} \in \Gamma(TM)$ is G-invariant.

This PDE can be solved using the *method of characteristics* as per the following result.

THEOREM 6.1 Let $u_0 \in C^\infty(G/H)$, then the convection PDE (6.3) is solved by

$$u_1(p,t) = u_0(g_p \gamma_{\boldsymbol{c}}(-t) p_0)$$

where $g_p \in p$ (i.e. $g_p p_0 = g_p H = p$) and $\gamma_{\boldsymbol{c}} : \mathbb{R} \to G$ is an exponential curve that satisfies

$$\pi_* \dot{\gamma}_{\boldsymbol{c}}(0) = \boldsymbol{c}_{p_0},$$

where $\pi : G \to G/H$ is the quotient map (2.7). The solution is again smooth for all $t > 0$.

Proof. Since π is a smooth submersion it follows that there always exists a $\dot{\gamma}_{\boldsymbol{c}}(0) \in T_e G$ so that $\pi_* \dot{\gamma}_{\boldsymbol{c}}(0) = \boldsymbol{c}_{p_0} \in T_{p_0}(G/H)$. Let $\gamma_{\boldsymbol{c}}(t) = \exp_G(t\, \dot{\gamma}_{\boldsymbol{c}}(0))$ be the corresponding exponential curve.

Denote the action of $g \in G$ on G/H by ρ_g and left multiplication in the group as L_g. Since $\boldsymbol{c}$ is assumed to be G-invariant we have

$$\boldsymbol{c}_p = (\rho_{g_p})_* \boldsymbol{c}_{p_0}$$

for all $p \in G/H$ and any $g_p \in p$. This implies

$$\boldsymbol{c}_p = (\rho_{g_p})_* \boldsymbol{c}_{p_0} = (\rho_{g_p})_* \pi_* \dot{\gamma}_{\boldsymbol{c}}(0) = \pi_* (L_{g_p})_* \dot{\gamma}_{\boldsymbol{c}}(0)$$

since $\rho_{g_p} \circ \pi = \pi \circ L_{g_p}$ and consequently $t \mapsto g_p \gamma_{\boldsymbol{c}}(t) p_0$ are the integral curves of $\boldsymbol{c}$ centered at the point p for any choice of $g_p \in p$.

Finally, let $p \in G/H$ and $t \geq 0$ then

$$\begin{aligned}
-\boldsymbol{c}_p u_1(\,\cdot\,, t) &= -\lim_{h \to 0} \frac{u_1(p,t) - u_1\left(g_p \gamma_{\boldsymbol{c}}(-h) p_0, t\right)}{h} \\
&= \lim_{h \to 0} \frac{u_0\left(g_p \gamma_{\boldsymbol{c}}(-h) \gamma_{\boldsymbol{c}}(-t) p_0\right) - u_0\left(g_p \gamma_{\boldsymbol{c}}(-t) p_0\right)}{h} \\
&= \lim_{h \to 0} \frac{u_0\left(g_p \gamma_{\boldsymbol{c}}(-(t+h)) p_0\right) - u_0\left(g_p \gamma_{\boldsymbol{c}}(-t) p_0\right)}{h} \\
&= \lim_{h \to 0} \frac{u_1(p, t+h) - u_1(p,t)}{h} \\
&= \frac{\partial u_1}{\partial t}(p,t).
\end{aligned}$$

□

TEMPORAL RESCALING. Note that $u_0(g_p\gamma_{\boldsymbol{c}}(-t)p_0) = u_0(g_p\gamma_{t\boldsymbol{c}}(-1)p_0)$. This means that there is a redundancy between scaling the convection vector-field and how long we let time run. In the context of training a neural network there is therefore no need to consider the end-time of the convection as trainable. In other words we can take $p \mapsto u_0(g_p\gamma_{\boldsymbol{c}}(-1)p_0)$ as our output and only train $\boldsymbol{c}$ as this will cover the full range of possible outputs.

CONVECTION ON $\mathbb{M}_2$. In the case of $\mathbb{M}_2 \equiv \mathrm{SE}(2)$, the G-invariant vector fields are exactly the left-invariant vector fields that are of the form

$$\boldsymbol{c} = c^1\mathcal{A}_1 + c^2\mathcal{A}_2 + c^3\mathcal{A}_3$$

per (4.3) with $c^1, c^2, c^3 \in \mathbb{R}$ being the free parameters. The exponential curve associated to this vector field is given in (x^1, x^2, θ) coordinates by (4.8) as:

$$\gamma_{\boldsymbol{c}}(t) = \begin{bmatrix} x^1(tc^1, tc^2, tc^3) \\ x^2(tc^1, tc^2, tc^3) \\ \theta(tc^1, tc^2, tc^3) \end{bmatrix} = \begin{bmatrix} \left(tc^1 \cos\frac{tc^3}{2} - tc^2 \sin\frac{tc^3}{2}\right) \operatorname{sinc}\frac{tc^3}{2} \\ \left(tc^1 \sin\frac{tc^3}{2} + tc^2 \cos\frac{tc^3}{2}\right) \operatorname{sinc}\frac{tc^3}{2} \\ tc^3 \bmod 2\pi \end{bmatrix}.$$

As we discussed in the previous paragraph, it is sufficient to consider $\gamma_{\boldsymbol{c}}(-1)$. Additionally, since we are in the principal homogeneous space and the exponential map on SE(2) is surjective it follows that we are training no more than a group element $h = \exp_{\mathrm{SE}(2)}(-\sum_{i=1}^3 c^i A_i) \in \mathrm{SE}(2)$ to right multiply with, i.e. $u_1(g, 1) = u_0(gh)$, since $p_0 = e$.

This covers the first term of the PDE (6.2). To solve the remaining three we first need to cover a way of efficiently estimating distances in homogeneous spaces.

6.6 LOGARITHMIC METRIC ESTIMATE

Previously we saw that in the $\mathbb{R}^2$ setting diffusion, dilation and erosion are solvable with the help of a kernel. We referenced that these kernels were actually scalar functions of the distance map from the origin, i.e. $\mathrm{d}_{\mathcal{G}}(\cdot, 0)$ if $\mathcal{G}$ is a translation invariant metric on $\mathbb{R}^2$. This holds more generally for the morphological kernels

on homogeneous spaces: they are scalar functions of the distance map to the reference element $p_0 \in M$ given by $\mathrm{d}_{\mathcal{G}}(\cdot, p_0)$, where the Riemannian metric $\mathcal{G}$ is invariant under action of the group. For the diffusion equation the kernel need not be a scalar function of the distance map, though estimates and bounds can still be constructed from the distance map and we will be making use of those.

Recall §1.7 for the relation between $\mathcal{G}$ and $d_{\mathcal{G}}$ and §2.4 for the group invariance of $\mathcal{G}$.

While calculating these distances on $\mathbb{R}^n$ is straightforward, this is not the case for distances on general homogeneous spaces. Indeed, in our main case of interest $M = \mathbb{M}_2$ there is no closed formula that we can conveniently plug in. To obtain a numeric distance map we would normally apply some iterative solver to the *eikonal equation*, but this presents an unacceptable computational overhead for our purposes. We intend to train the parameters of our Riemannian metrics, meaning the Riemannian metrics will not only vary across the neural network but also change constantly. This would force us to endlessly recompute the solution to the eikonal equation which would slow the training process down by orders of magnitude.

What we need to make our approach viable is an approximation that is easy to compute while still being accurate enough. Since the most important part of both the diffusion and morphological kernels are localized around the point of reference, our approximation only needs be locally accurate. The approximation that will use is the *logarithmic metric estimate* that we cover next.

DEFINITION 6.1 (Logarithmic metric estimate) Let $\mathcal{G}$ be a G-invariant metric tensor field on the homogeneous space G/H, let $\pi : G \to G/H$ be the quotient map (2.7). Then we define the *logarithmic metric estimate* $\rho_{\mathcal{G}} : G/H \to \mathbb{R}_{\geq 0}$ as

$$\begin{aligned} \rho_{\mathcal{G}}(p) &:= \inf_{g \in p} \left\| \log_G(g) \right\|_{\pi^* \mathcal{G}} \\ &= \inf_{g \in p} \sqrt{\mathcal{G}\left(\pi_* \log_G(g), \pi_* \log_G(g)\right)}. \end{aligned} \tag{6.4}$$

Recall §2.2 and the equivalence of every homogeneous space with a quotient with a closed subgroup H.

We can interpret this metric estimate as finding all exponential curves in G whose actions on the homogeneous space connect p_0 (at $t = 0$) to p (at $t = 1$), and then, from that set we choose the

exponential curve that has the lowest constant velocity according to the seminorm given by $\pi^*\mathcal{G}$ and use that velocity as the distance estimate.

Note that scaling the metric, scales the metric estimate just like it would scale the actual metric.

COROLLARY 6.2 (Logarithmic metric estimate scaling) In the setting of Definition 6.1 we have that $\rho_{\lambda\mathcal{G}} = \lambda\rho_{\mathcal{G}}$ for all $\lambda > 0$.

Intuitively, Definition 6.1 can be reformulated as *'instead of the length of the geodesic connecting two points of the homogeneous space, we take the length of the shortest exponential curve connecting those two points'*.

The following lemma quantifies how well our estimate $\rho_{\mathcal{G}}$ approximates the true distance map $\mathrm{d}_{\mathcal{G}}(p_0, \cdot)$.

LEMMA 6.3 (Bounds on the logarithmic metric estimate) For all $p \in G/H$ sufficiently close to $p_0 = H$, we have

$$\mathrm{d}_{\mathcal{G}}(p_0, p)^2 \leq \rho_{\mathcal{G}}(p)^2 \leq \mathrm{d}_{\mathcal{G}}(p_0, p)^2 + O\left(d_{\mathcal{G}}(p_0, p)^4\right)$$

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

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

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

Proof. Let $\tilde{\mathcal{G}} := \pi^*\mathcal{G}$. The left inequality follows directly from the observation that $\rho_{\mathcal{G}}(p)$ is exactly the Riemannian length of the curve

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

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

For the right inequality, consider the function $F : T_eG \to \mathbb{R}$ given by

$$F(v) := d_{\mathcal{G}}(p_0, \pi \circ \exp_G(v))^2,$$

where we recall that $\pi : G \to G/H$ was given by (2.7). With the goal of making a Taylor expansion for this function we note that:

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

Moreover, due to the G-invariance of $d_{\mathcal{G}}$, the function F is even since

$$\begin{aligned} F(v) &= d_{\mathcal{G}}(p_0, \pi \circ \exp_G(v))^2 \\ &= d_{\mathcal{G}}(p_0, \exp_G(v) \cdot p_0)^2 \\ &= d_{\mathcal{G}}(\exp_G(v)^{-1} \cdot p_0, p_0)^2 \\ &= d_{\mathcal{G}}(\pi \circ \exp_G(-v), p_0)^2 \\ &= F(-v) \end{aligned}$$

and consequently the 3rd order term of the Taylor expansion of F is zero.

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

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

Putting these facts together we find:

$$F(v) = \tilde{\mathcal{G}}|_e(v, v) + O(\|v\|^4), \tag{6.5}$$

where $\|\cdot\|$ denotes some arbitrary norm on T_eG.

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

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

and so by (6.5) we have:

$$F(v) = d_{\mathcal{G}}(p_0, \pi \circ \exp_G(v))^2 > c\|v\|^2 + O\left(\|v\|^4\right). \tag{6.6}$$

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

$$d_{\mathcal{G}}(p_0, \pi \circ \exp_G(v))^2 > \frac{c}{2}\|v\|^2. \tag{6.7}$$

In a neighborhood of the origin the Lie group exponential map $\exp_G : T_eG \to G$ is a diffeomorphism to a neighborhood of e, at the same time $\pi : G \to G/H$ is a smooth submersion by the *homogeneous space construction theorem* (Lee, 2012, Thm. 21.17). Consequently $(\pi \circ \exp_G)_*|_0 : V \to T_{p_0}(G/H)$ has full rank since $\pi \circ \exp_G$ is a local diffeomorphism between two spaces (V and G/H) with the same dimension, it follows by the inverse function theorem that there exists a neighborhood V_0 of 0 in V and a neighborhood P_0 of p_0 in G/H such that $\pi \circ \exp_G$ is a diffeomorphism from V_0 to P_0. By possibly choosing V_0 smaller, we may assume by (6.6) that there exists a $C' > 0$ such that for all $v \in V_0$:

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

which, by using (6.7) yields

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

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

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

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

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

$$\left\|\log_G g_p\right\|_{\tilde{\mathcal{G}}}^2 \leq d_{\mathcal{G}}(p, p_0)^2 + C d_{\mathcal{G}}(p, p_0)^4 .$$

Clearly $g_p \in p$. Since $\rho_{\mathcal{G}}(p)$ is the infinum of $\|\log_G g\|$ for all $g \in p$ it follows that $\rho_{\mathcal{G}}(p)$ must also satisfy:

$$\rho_{\mathcal{G}}(p)^2 \leq \left\|\log_G g_p\right\|_{\tilde{\mathcal{G}}}^2 \leq d_{\mathcal{G}}(p, p_0)^2 + C d_{\mathcal{G}}(p, p_0)^4,$$

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

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

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

for all $p \in K$. We can see this by choosing $C_1^2 = 1 + C \sup_{p \in P_1} d_{\mathcal{G}}(p_0, p)^2$, then for all by choosing $p \in P_0$ we have

$$\rho_{\mathcal{G}}(p) \leq C_1 d_{\mathcal{G}}(p_0, p).$$

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

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

and

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

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

$$\rho_{\mathcal{G}}(p) \leq M \leq \frac{M}{d_{\mathcal{G}}(p_0, p)} d_{\mathcal{G}}(p_0, p) \leq \frac{M}{m} d_{\mathcal{G}}(p_0, p).$$

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

When we take the principal homogeneous space, such as in the case of $\mathbb{M}_2 \equiv \mathrm{SE}(2)$, with a left-invariant Riemannian metric $\mathcal{G}$ then the metric estimate simplifies to

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

hence we see that this construction generalizes the logarithmic metric estimate, as used in Portegies, Sanguinetti, Meesters, and Duits (2015) and ter Elst and Robinson (1998), to homogeneous spaces other than the principal.

ON $\mathbb{M}_2$. Consider those Riemannian metrics on $\mathbb{M}_2$ that are diagonal with respect to the basis of G-invariant vector fields as per (4.7). Using the usual (x^1, x^2, θ) coordinates, these metrics are of the form

$$\mathcal{G} = \sum_{i=1}^{3} w_i^2 \, \omega^i \otimes \omega^i, \tag{6.8}$$

where we refer to $w_1 > 0$ as the cost in the *main* direction, $w_2 > 0$ as the cost in the *lateral* direction and $w_3 > 0$ as the *angular* cost. Then we can formulate the logarithmic metric estimate in terms of the exponential coordinates of the first kind (4.10) as

$$\rho_{\mathcal{G}}(x^1, x^2, \theta) = \sqrt{\sum_{i=1}^{3} \left(w_i \, c^i(x^1, x^2, \theta)\right)^2}.$$

This estimate is illustrated in Figure 6.3 where it is contrasted against the 'exact' distance map obtained by solving the eikonal equation.

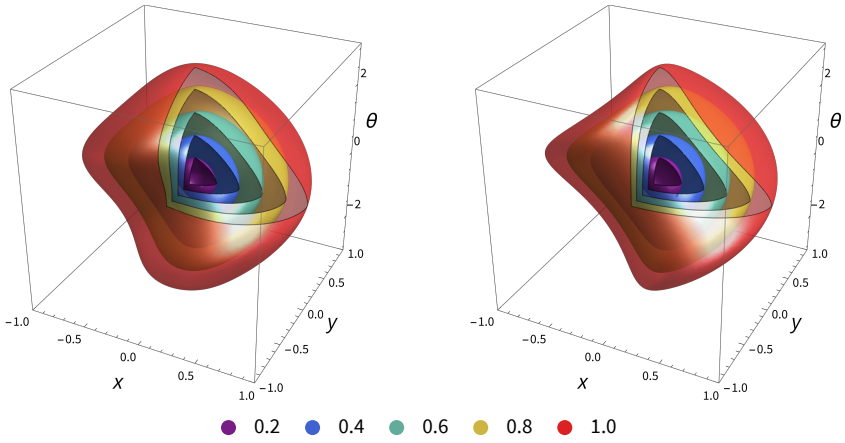


FIGURE 6.3: Comparing the 'exact' Riemannian distance (left) obtained through numerically solving the Eikonal equation (BEKKERS, DUITS, MASHTAKOV, and SANGUINETTI, 2015) versus the logarithmic metric estimate (right) on $\mathbb{M}_2$ endowed with a left-invariant Riemannian metric tensor field (6.8) with $w_1^2 = 1$, $w_2^2 = 2$, $w_3^2 = 1/\pi$. The relative L^1 error in the plotted volume is 0.20.

We can see that the metric estimate $\rho_{\mathcal{G}}$ (and consequently any function of $\rho_{\mathcal{G}}$) has the necessary symmetry to be a kernel used in

the operators of Theorem 3.3. The proof of this statement relies on the following lemma.

LEMMA 6.4 Let G be a Lie group, for all $g \in G$ let $L_g : G \to G$, $R_g : G \to G$ be the left respectively right group multiplication. Let H be a closed subgroup of G with the quotient map $\pi : G \to G/H$ given by $\pi(g) = gH$. Then for all $h \in H$ we have the following relations for the push forwards:

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

Which implies in particular that

$$\pi_*(L_h)_* = (L_h)_*\pi_* \qquad \text{and} \qquad \pi_*(R_h)_* = \pi_*$$

for all $h \in H$.

Proof.

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

□

We can now show that the metric estimate is indeed compatible in the sense of (3.12).

THEOREM 6.5 (Kernel compatibility of $\rho_{\mathcal{G}}$) Let $\mathcal{G}$ be a G-invariant metric tensor field on G/H, then we have

$$\forall p \in G/H,\ \forall h \in H : \rho_{\mathcal{G}}(hp) = \rho_{\mathcal{G}}(p). \tag{6.9}$$

Note that, since we use left cosets, $ph = p$ but $hp \neq p$ in general.

Proof. Consider the set of all exponential curves in the group whose action connects $p_0 \in G/H$ to $p \in G/H$:

$$\Gamma_{p_0,p} = \left\{ \gamma \in \mathrm{Lip}([0,1], G) \;\middle|\; \gamma(0) = e,\ \gamma(1)p_0 = p,\ \gamma(t+s) = \gamma(t)\gamma(s) \right\}.$$

We can then restate $\rho_{\mathcal{G}}$ equivalently in terms of these curves as

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

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

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

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

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

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

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

$$\begin{aligned}
\left\| \left(h\gamma(\cdot)h^{-1} \right)(0) \right\|_{\tilde{\mathcal{G}}} &= \|(L_h)_* (R_{h^{-1}})_* \dot{\gamma}(0)\|_{\tilde{\mathcal{G}}} \\
&= \|\pi_* (L_h)_* (R_{h^{-1}})_* \dot{\gamma}(0)\|_{\mathcal{G}} \\
\text{(using the previous lemma)} \qquad &= \|(L_h)_* \pi_* (R_{h^{-1}})_* \dot{\gamma}(0)\|_{\mathcal{G}} \\
&= \|(L_h)_* \pi_* \dot{\gamma}(0)\|_{\mathcal{G}} \\
\text{(using the G-invariance of } \mathcal{G}) \qquad &= \|\pi_* \dot{\gamma}(0)\|_{\mathcal{G}} \\
&= \|\dot{\gamma}(0)\|_{\tilde{\mathcal{G}}}.
\end{aligned}$$

It follows that

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

□

BETTER ESTIMATES FOR $\mathbb{M}_2$. The distance estimate from Definition 6.1 and its associated bounds from Lemma 6.7 from SMETS, PORTEGIES, BEKKERS, and DUITS (2022) are very general since they apply to all homogeneous spaces. When we restrict ourselves to

$\mathbb{M}_2$ improved distance estimates and bounds can be made as has been done by BELLAARD, BON, PAI, SMETS, and DUITS (2023). It turns out that on $\mathbb{M}_2$ with a diagonal Riemannian metric (6.8), that the quality of the approximation (6.4) depends entirely on the spatial anisotropy

$$\zeta = \max\left\{\frac{w_1}{w_2}, \frac{w_2}{w_1}\right\}, \tag{6.10}$$

see BELLAARD, BON, PAI, SMETS, and DUITS (2023, Thm. 1). As our original work (SMETS, PORTEGIES, BEKKERS, and DUITS, 2022) uses the distance estimate from Definition 6.1 we will continue with using it here.

Now that we have developed and analyzed the logarithmic metric estimate we can use it to construct an approximation of the diffusion kernel.

6.7 FRACTIONAL DIFFUSION

Diffusion — whether fractional or not — is a linear process that can be solved with the help of its Green's function, i.e. the *heat kernel*, even in the more general setting of a metric space (GRIGOR'YAN, 2009; GRIGOR'YAN, HU, and LAU, 2014). Taken separately (fractional) diffusion on some homogeneous space M solves the equation

For an overview of nonlinear and fractional diffusion in the Euclidean setting see VÁZQUEZ (2017).

$$\frac{\partial u_2}{\partial t}(p,t) = -\left(-\Delta_{\mathcal{G}}\right)_p^{\nu} u_2(\cdot,t) \tag{6.11}$$

with some initial condition $u_2(\cdot,0) = u_0$, which could be the output $u_1(\cdot,1)$ of the previous convection substep. The solution to (6.11) is obtained by

$$u_2(p,t) = \int_M \kappa_t(p,q)\, u_0(q)\, \mathrm{d}q \tag{6.12}$$

where $\kappa^{\nu} \in C^{\infty}(\mathbb{R}_{>0} \times M \times M)$ is called the *heat kernel* for a given choice of $\nu > 0$. The parameter ν can in principle be any positive value, though if $\nu < 1$ then the heat kernel is guaranteed to be positive which is not the case if $\nu > 1$. For image processing application ν is usually chosen in the range $[1/2, 1]$.

In our by now familiar setting we assume that M is a homogeneous space with respect to some Lie group G, that the integral

$\int_M \mathrm{d}q$ is at least covariant with a character $\chi_M : G \to \mathbb{R}_{>0}$ as in §3.2 and that the Riemannian metric $\mathcal{G} \in \Gamma(T^{(0,2)}M)$ is G-invariant as in §2.4. Hence, the evolution brought about by (6.11) is equivariant with respect to G. This brings us in the setting of the equivariant linear operators of Theorem 3.3, implying we can formulate a *one-argument* heat kernel $K^\nu : \mathbb{R}_{>0} \times M \to \mathbb{R}$ as

$$K_t^\nu(p) := \kappa_t^\nu(p, p_0).$$

for a choice of reference element $p_0 \in M$. The reduced kernel necessarily has the symmetry

$$h \cdot K_t^\nu = \chi_M(h)\, K_t^\nu$$

for all $h \in G_{p_0}$ and $t > 0$ as per (3.12). This gives the solution to the diffusion equation as

$$u_2(p,t) = \frac{1}{\chi_M(g_p)} \int_M (g_p \cdot K_t^\nu)(q)\, f(q)\, \mathrm{d}q \tag{6.13}$$

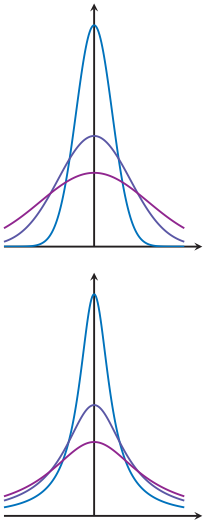
where a $g_p \in G_{p_0 \to p}$ is freely chosen for all $p \in M$, recall (2.6).

GAUSSIAN & POISSON KERNEL. The heat kernel in the $\mathbb{R}^n$ case for regular diffusion, i.e. $\nu = 1$, is given by the well known *Gaussian function*:

$$K_t^1(x) = \frac{1}{(4\pi t)^{n/2}} \exp\left(-\frac{|x|^2}{4t}\right), \tag{6.14}$$

which is also known simply as *the Gaussian*. For the fractional case $\nu = 1/2$ the heat kernel on $\mathbb{R}^n$ is called the *Poisson kernel* and is given by

$$K_t^{1/2}(x) = \frac{\Gamma(\frac{n+1}{2})}{\pi^{\frac{n+1}{2}} t^n} \left(1 + \frac{|x|^2}{t^2}\right)^{-\frac{n+1}{2}}. \tag{6.15}$$



Gaussians (6.14) (top) and Poisson kernels (6.15) (bottom) on $\mathbb{R}$ for different values of t.

Outside of $\mathbb{R}^n$ or for other values of ν the situation is less straightforward, while for some cases such as the *Heisenberg group* (GAVEAU, 1977) and $\mathbb{M}_2$ (DUITS and VAN ALMSICK, 2008) explicit formulas for their respective heat kernels exist, these are seldom convenient for numerical computation and so not suitable for repeated recomputation as is required for our application that uses spatial convolutions. In the more general setting of Riemannian manifolds, no closed formula exists and the kernel need not be a scalar function of the Riemannian distance in the same way as it is on $\mathbb{R}^n$.

TWO-SIDED BOUNDS. The best we can do in general is bound the heat kernel from above and below with Gaussian-like functions. There is much work available in this domain (GRIGOR'YAN, 2009; GRIGOR'YAN, 2006; GRIGOR'YAN, HU, and LAU, 2009; MAHEUX, 1998; MUSTAPHA, 2000; VARADHAN, 1967). Our work is based on the bounds from MAHEUX (1998) for homogeneous spaces whose Lie group is of *polynomial growth*.

DEFINITION 6.2 (Polynomial growth) A Lie group G with left-invariant Haar measure μ_G is of *polynomial growth* when the volume of a sphere of radius r around any $g \in G$:

$$B(g, r) := \{g' \in G \mid \mathrm{d}_{\mathfrak{g}}(g, g') < r\},$$

can be polynomialy bounded as follows: there exists constant $\delta > 0$ and $C_{\text{grow}} > 0$ so that

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

for all $r \geq 1$. Take note that the exponent δ is the same on both the lower and upper bound. Since μ_G is always left-invariant in our applications the choice of g is perfunctory.

On homogeneous spaces of Lie groups of polynomial growth, the following two-sided bound can be made on the two-argument heat kernel κ^1 (6.12) as per MAHEUX (1998, Thm. 2.12) for non-fractional diffusion.

LEMMA 6.6 (Maheux heat kernel bounds) Let $M = G/H$ be a homogeneous space with G being of polynomial growth as per Def. 6.2 then there exists constants $C_1, C_2 > 0$ and for every $\varepsilon > 0$ there exists a $C_\varepsilon > 0$ so that

$$C_1 \eta_{q,t} \exp\left(-\frac{\mathrm{d}_{\mathcal{G}}(p,q)^2}{4C_2 t}\right) \leq \kappa_t^1(p,q) \leq C_\varepsilon \eta_{q,t} \exp\left(-\frac{\mathrm{d}_{\mathcal{G}}(p,q)^2}{4(1+\varepsilon)t}\right) \tag{6.16}$$

for all $p, q \in M$ and all $t > 0$ where $\mathcal{G}$ is the Riemannian metric with its associated distance $d_{\mathcal{G}}$ and the normalization constant $\eta_{q,t}$ is given by

$$\eta_{q,t} := \frac{1}{\mu_{\mathcal{G}}\left(B(q, \sqrt{t})\right)}.$$

where $\mu_{\mathcal{G}}$ is the volume measure induced by $\mathcal{G}$.

DIFFERENCES WITH OUR SETTING. We adapted the formulation of Lemma 6.6 to our setting, which is distinct (but equivalent) to that of MAHEUX (1998) in two ways. Firstly, MAHEUX (1998, Thm. 2.12) proves this result for *right* cosets, while we use *left* cosets. We can translate the results by inversion in view of $(gH)^{-1} = H^{-1}g^{-1} = Hg^{-1}$. We then apply the result of Maheux to the correct G-invariant metric tensor field on G/H.

Secondly, Maheux relates distances on the group with distances on the homogeneous space in a distinct (but equivalent) way. While we use a pseudometric on G induced by a metric on G/H using the pullback of the quotient map (2.8), Maheux does it the other way around and uses a metric on G/H induced by a metric on G by:

$$\begin{aligned} d_{G/H}^{\text{maheux}}(p_1, p_2) &= \inf_{g_1 \in p_1} \inf_{g_2 \in p_2} d_G^{\text{maheux}}(g_1, g_2) \\ &= \inf_{g_2 \in p_2} d_G^{\text{maheux}}(q_1, g_2), \end{aligned} \tag{6.17}$$

for any choice of $q_1 \in p_1$ and where d_G is the initial metric on G. We avoid having to minimize inside the cosets as in (6.17) thanks to the inherent symmetries in our pseudometric.

AN APPROXIMATE KERNEL. That brings us to our proposed approximation of the K^1 heat kernel. This approximation for $\nu = 1$

will also serve as the basis for our approximation of K^ν for other values of ν.

DEFINITION 6.3 (Approximate $\nu = 1$ kernel) On a homogeneous space G/H with G-invariant Riemannian metric $\mathcal{G}$ we define

$$K_t^{1,\text{appr}}(p) := \eta_t \exp\left(-\frac{\rho_{\mathcal{G}}(p)^2}{4t}\right) \tag{6.18}$$

where η_t is a normalization constant for a given $t > 0$, this can either be the L^1 normalization constant or in the case of groups of polynomial growth we set

$$\eta_t := \frac{1}{\mu_{\mathcal{G}}\left(B(p_0, \sqrt{t})\right)}.$$

We can squeeze our approximate kernel $K^{1,\text{appr}}$ on both sides by scaled real kernels K^1 as follows.

LEMMA 6.7 Let G be of polynomial growth and let K^1 be the heat kernel for the $\nu = 1$ diffusion equation (6.11) on the homogeneous space G/H. Then there exists constants $C \geq 1$, $D_1 \in (0, 1)$ and $D_2 > D_1$ so that the following holds:

$$\frac{1}{C} K^1_{D_1 t}(p) \leq K_t^{1,\text{appr}}(p) \leq C K^1_{D_2 t}(p). \tag{6.19}$$

for all $p \in G/H$ and $t > 0$.

Proof. On a group of polynomial growth we have $\eta_t = \mu_{\mathcal{G}}\left(B(p_0, \sqrt{t})\right)^{-1}$. If G is of polynomial growth we can apply Lemma 6.6 to find that there exists constants $C_1, C_2 > 0$ and for all $\varepsilon > 0$ there exists a constant $C_\varepsilon > 0$ so that:

$$C_1 \eta_t \exp\left(-\frac{d_{\mathcal{G}}(p_0, p)^2}{4C_2 t}\right) \leq K_t^1(p) \leq C_\varepsilon \eta_t \exp\left(-\frac{d_{\mathcal{G}}(p_0, p)^2}{4(1+\varepsilon)t}\right)$$

for all $p \in G/H$ and $t > 0$.

Now using the inequalities from Lemma 6.3 we obtain:

$$C_1 \eta_t \exp\left(-\frac{\rho_{\mathcal{G}}(p)^2}{4C_2 t}\right) \leq K_t^1(p) \leq C_\varepsilon \eta_t \exp\left(-\frac{\rho_{\mathcal{G}}(p)^2}{4C_{\text{metr}}^2(1+\varepsilon)t}\right),$$

which leads to:

$$C_1 \frac{\eta_t}{\eta_{c_2 t}} K^{1,\mathrm{appr}}_{C_2 t}(p) \leq K^1_t(p) \leq C_\varepsilon \frac{\eta_t}{\eta_{C^2_{\mathrm{metr}}(1+\varepsilon)t}} K^{1,\mathrm{appr}}_{C^2_{\mathrm{metr}}(1+\varepsilon)t}(p).$$

The group G being of polynomial growth also implies G/H is a doubling space (Maheux, 1998, Thm. 2.17). Using the volume doubling and reverse volume doubling property of doubling spaces (Grigor'Yan, Hu, and Lau, 2009, Prop. 3.2 and 3.3) we find that there exist constants $C_3, C_4, \beta, \beta' > 0$ so that:

$$\frac{\eta_t}{\eta_{c_2 t}} \geq C_3 \left(\frac{\sqrt{t}}{\sqrt{C_2 t}} \right)^{\beta} = C_3 C_2^{-\beta/2},$$

$$\frac{\eta_t}{\eta_{C^2_{\mathrm{metr}}(1+\varepsilon)t}} \leq C_4 \left(\frac{\sqrt{t}}{\sqrt{C^2_{\mathrm{metr}}(1+\varepsilon)t}} \right)^{\beta'} = C_4 \left(C^2_{\mathrm{metr}}(1+\varepsilon) \right)^{-\beta'/2}.$$

Applying these inequalities we get:

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

and

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

From which we obtain:

$$C'_1 K^{1,\mathrm{appr}}_{C_2 t}(p) \leq K^1_t(p) \leq C'_\varepsilon K^{1,\mathrm{appr}}_{C^2_{\mathrm{metr}}(1+\varepsilon)t}(p).$$

Reparametrizing t in both inequalities gives:

$$\frac{1}{C'_\varepsilon} K^1_{t/(C^2_{\mathrm{metr}}(1+\varepsilon))}(p) \leq K^{1,\mathrm{appr}}_t(p) \leq \frac{1}{C'_1} K^1_{C_2^{-1} t}(p).$$

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

$$C := \max\left\{ C_1'^{-1},\ C'_\varepsilon,\ 1 \right\},$$
$$D_1 := \frac{1}{C^2_{\mathrm{metr}}(1+\varepsilon)},$$
$$D_2 := \frac{1}{C_2},$$

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

Depending on the actually achievable constants, Lemma 6.7 provides a very strong or very weak bound on how much our approximation deviates from the actual solution. Fortunately in the $\mathbb{M}_2$ case our approximation is very close to the exact kernel in the vicinity of the origin, as can be seen in Figure 6.4. In our experiments we sample the kernel on a grid around the origin, hence this approximation is good for reasonable values of the metric parameters (i.e. low spatial anisotropy, $\zeta < 4$ ideally, recall (6.10)), which we may expect from Lemma 6.3 providing a second order relative error.

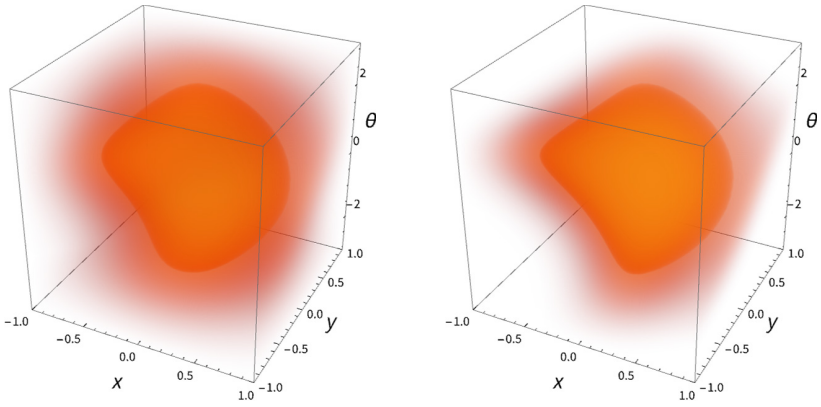


FIGURE 6.4: Comparing the numerically computed heat kernel K_t^1 (left) with our approximation $K_t^{1,\text{appr}}$ based on the logarithmic norm estimate (right) for $G/H = \mathbb{M}_2$. Shown here at $t = 1$ with the same metric as in Fig. 6.3. Especially in deep learning applications where discretization is very coarse our approximation is sufficiently accurate as long as the spatial anisotropy (6.10) does not become too high. In this case with $\zeta = w_2/w_1 = 2$ we have a relative L^2 error of 0.23 in the plotted volume.

FRACTIONAL KERNEL APPROXIMATION. Now let us develop an approximation for values of ν other than 1. From semigroup theory (YOSIDA, 1968) it follows that semigroups generated by taking fractional powers of the generator (in our case $\Delta_{\mathcal{G}} \to -(-\Delta_{\mathcal{G}})^{\nu}$)

amounts to the following key relation between the ν-kernel and the diffusion kernel:

$$K_t^\nu(p) := \int_0^\infty q_{t,\nu}(\tau)\, K_\tau^1(p)\, \mathrm{d}\tau, \tag{6.20}$$

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

DEFINITION 6.4 Let $\mathcal{L}^{-1}$ be the inverse Laplace transform then

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

For explicit formulas of this kernel see YOSIDA (1968, Ch. IX:11 eq. 17). Since e^{-tr^ν} is positive for all r it follows that $q_{t,\nu}$ is also positive everywhere.

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

DEFINITION 6.5 (Approximate $\nu \in (0,1)$ kernel) Akin to (6.20) we set $\nu \in (0,1)$, $t > 0$ and define:

$$K_t^{\nu,\mathrm{appr}}(p) := \int_0^\infty q_{t,\nu}(\tau)\, K_\tau^{1,\mathrm{appr}}(p)\ \mathrm{d}\tau \geq 0, \tag{6.21}$$

for $p \in G/H$.

The bound on K_t^1 we obtained in Lemma 6.7 transfers directly to our approximation for other ν.

THEOREM 6.8 Let G be of polynomial growth and let K_t^ν be the heat kernel of the $\nu \in (0,1]$ diffusion equation (6.11) on the homogeneous space G/H, then there exists constants $C \geq 1$, $D_1 \in (0,1)$ and $D_2 > D_1$ so that the following holds:

$$\frac{1}{C} K^\nu_{D_1^\nu t}(p) \leq K_t^{\nu,\mathrm{appr}}(p) \leq C K^\nu_{D_2^\nu t}(p) \tag{6.22}$$

for all $t > 0$ and $p \in G/H$.

Proof. The statement is an consequence of Lemma 6.7 and the fact that $q_{t,\nu}$ is positive, applying the integral from (6.21) yields:

$$\begin{aligned} K_t^{\nu,\text{appr}}(p) &= \int_0^\infty q_{t,\nu}(\tau) K_\tau^{1,\text{appr}}(p)\,\mathrm{d}\tau \\ \text{(Lem. 6.7)} \qquad &\leq C \int_0^\infty q_{t,\nu}(\tau) K_{D_2\tau}^1(p)\,\mathrm{d}\tau \\ (\tau'=D_2\tau) \qquad &= C \int_0^\infty \frac{1}{D_2} q_{t,\nu}\left(\frac{\tau'}{D_2}\right) K_{\tau'}^1(p)\,\mathrm{d}\tau' \\ \left(\begin{smallmatrix}\text{Bromwich}\\ \text{integral}\end{smallmatrix}\right) \qquad &= C \int_0^\infty q_{D_2^\nu t,\alpha}(\tau')\, K_{\tau'}^1(p)\,\mathrm{d}\tau' \\ &= C K_{D_2^\nu t}^\nu(p). \end{aligned}$$

The other inequality follows the same way. □

Although the approximation (6.21) is helpful in the proof above it contains some integration and is not an explicit expression. Our initial experiments with diffusion for $\nu = 1$ showed that (for the applications under consideration at least) adding diffusion did not improve performance. For that reason we chose not to focus further on diffusion in this work. We leave developing a more explicit and computable approximation for diffusion kernels for $0 < \nu < 1$ for future work.

6.8 DILATION & EROSION

The last part of (6.2) to discuss are the morphological terms that control dilation and erosion. In isolation the dilation or erosion processes solve the PDE

$$\begin{cases} \frac{\partial u_3}{\partial t}(p,t) = \pm\frac{1}{\alpha}\left\|(\nabla_{\mathcal{G}} u_3(\cdot,t))_p\right\|_{\mathcal{G}}^\alpha & \forall p \in M, t \geq 0, \\ u_3(\cdot,0) = u_0 \end{cases} \tag{6.23}$$

where $\alpha > 1$ and with '+' indicating dilation and '-' erosion from an initial condition u_0 which could be the output $u_2(\cdot,1)$ of the previous substep. When the smoothness parameter α is greater than 1 then we are looking at the viscosity solution of (6.23), we can also take the limit of those viscosity solutions as $\alpha \to 1$ to talk about the solution of this PDE for $\alpha = 1$ as we will see.

The dilation and erosion PDEs yield non-linear scale spaces as per VAN DEN BOOMGAARD *and* SMEULDERS *(1994) and* SCHMIDT *and* WEICKERT *(2016).*

HAMILTONIAN & LAGRANGIAN. The PDE (6.23) is a *Hamilton-Jacobi* equation

$$\begin{cases} \frac{\partial u_3}{\partial t}(p,t) = \pm \mathcal{H}_\alpha\left((\mathrm{d}u_3(\cdot,t))_p\right) & \forall p \in M, t \geq 0, \\ u_3(\cdot,0) = u_0 \end{cases} \tag{6.24}$$

where the Hamiltonian $\mathcal{H}_\alpha : T^*M \to \mathbb{R}_{\geq 0}$ is given by

$$\mathcal{H}_\alpha(\xi) := \frac{1}{\alpha}\|\xi\|_{*\mathcal{G}}^\alpha := \frac{1}{\alpha}\left\|\xi^\sharp\right\|_{\mathcal{G}}^\alpha \tag{6.25}$$

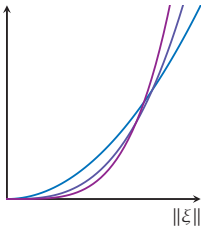
for any $\xi \in T^*M$. Going forward we will suppress the $\mathcal{G}$ subscript from the norm $\|\cdot\|$ and dual norm $\|\cdot\|_*$ as it is understood that these depend on the Riemannian metric $\mathcal{G} \in \Gamma(T^{(0,2)}M)$.

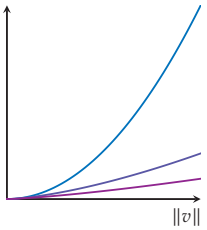
Recall that the musical isomorphism $\sharp : T^*M \to TM$ is defined so that $\mathcal{G}(\xi^\sharp, \cdot) = \xi$ for all $\xi \in T^*M$.

Clearly, the Hamiltonian is a strictly convex, $\mathcal{H}_\alpha(0) = 0$ and moreover is superlinear since

$$\frac{\mathcal{H}_\alpha(\xi)}{\|\xi\|_*} \to \infty$$

if $\|\xi\| \to \infty$. The associated Lagrangian is found by applying the Fenchel transform.





Hamiltonians (6.25) (top) and Lagrangians (6.26) (bottom) for $\alpha = 2, 3, 4$ in function of the norms.

LEMMA 6.9 (Morphological Lagrangian) The Lagrangian $\mathcal{L}_\alpha : TM \to \mathbb{R}_{\geq 0}$ belonging to the Hamiltonian (6.25) for $\alpha > 1$ is given by

$$\mathcal{L}_\alpha(v) = \frac{1}{\beta}\|v\|^\beta \tag{6.26}$$

for all $v \in TM$ and where $\beta > 1$ is the complement of α given by

$$\frac{1}{\alpha} + \frac{1}{\beta} = 1 \tag{6.27}$$

Proof. Let $v \in T_pM$ at some $p \in M$, then

$$\mathcal{L}_\alpha(v) := \sup_{\xi \in T_p^*M}\{\xi(v) - \mathcal{H}_\alpha(\xi)\} = \sup_{\xi \in T_p^*M}\left\{\xi(v) - \frac{1}{\alpha}\|\xi\|^\alpha\right\}.$$

Clearly, the optimal $\xi \in T_p^*M$ satisfies $\xi^\sharp = \lambda v$ for some $\lambda \geq 0$ so we can rewrite

$$\mathcal{L}_\alpha(v) = \sup_{\lambda \geq 0}\left\{\mathcal{G}(\lambda v, v) - \frac{1}{\alpha}\|\lambda v\|^\alpha\right\} = \sup_{\lambda \geq 0}\left\{\lambda\|v\|^2 - \frac{1}{\alpha}\lambda^\alpha\|v\|^\alpha\right\}.$$

Taking the derivative of the objective gives $\|v\|^2 - \lambda^{\alpha-1}\|v\|^\alpha$, which is zero only if $\lambda = \|v\|^{\frac{2-\alpha}{\alpha-1}}$. Filling it in into the objective function yields

$$\mathcal{L}_\alpha(v) = \|v\|^{\frac{2-\alpha}{\alpha-1}}\|v\|^2 - \frac{1}{\alpha}\left(\|v\|^{\frac{2-\alpha}{\alpha-1}}\right)^\alpha \|v\|^\alpha = \left(1 - \frac{1}{\alpha}\right)\|v\|^{\frac{\alpha}{\alpha-1}} = \frac{1}{\beta}\|v\|^\beta.$$

□

Note that both the Hamiltonian (6.25) and Lagrangian (6.26) are scalar functions of norms on their respective domains. So, we could just as easily express the Hamiltonian and Lagrangian in function of those '1D' scalar functions $\mathcal{H}_\alpha^{1D} : \mathbb{R} \to \mathbb{R}$ and $\mathcal{L}_\alpha^{1D} : \mathbb{R} \to \mathbb{R}$ as

$$\begin{aligned} \mathcal{H}_\alpha(\xi) &= \mathcal{H}_\alpha^{1D}(\|\xi\|_*) := \frac{1}{\alpha}\|\xi\|_*^\alpha, \\ \mathcal{L}_\alpha(v) &= \mathcal{L}_\alpha^{1D}(v) := \frac{1}{\beta}\|v\|^\beta, \end{aligned} \tag{6.28}$$

for all $\xi \in T^*M$, $v \in TM$ and where $\alpha, \beta > 1$ are related as per (6.27). Some works refer to the scalar functions $\mathcal{H}_\alpha^{1D}$ and $\mathcal{L}_\alpha^{1D}$ as *the* Hamiltonian respectively Lagrangian, we make the distinction here.

MORPHOLOGICAL CONVOLUTION. In $\mathbb{R}^n$, a PDE of the type (6.23) is solved with the help of a morphological convolution as in (6.1). Analogously to how we generalized linear convolutions to homogeneous spaces in Theorem 3.3 we can generalize morphological convolutions in a similar manner.

DEFINITION 6.6 (Morphological convolution) Let G/H be a homogeneous space of a Lie group G, $f \in L^\infty(G/H)$ and let $k : G/H \to \mathbb{R} \cup \{\infty\}$ be proper (i.e. not everywhere ∞) then we define the *morphological convolution* of f with kernel k as:

$$(k \square_{G/H} f)(p) := \inf_{q \in G/H} \{(g_p \cdot k)(q) + f(q)\} \tag{6.29}$$

where $g_p \in p$ is freely chosen for all $p \in G/H$. For this expression to be well-defined we require that the kernel k satisfies the compatibility condition:

$$\forall h \in H : h \cdot k = k. \tag{6.30}$$

When it is clear on what space we are applying the morphological convolution we drop the subscript from $\square_{G/H}$ and just write $\square$.

ANALOGY WITH THE LINEAR SETTING. Compare and contrast the operators from Definition 6.6 with those of Theorem 3.3. The overall construction of the two types of operators is very similar except that multiplication is changed into addition and the integral is changed into an infimum. The compatibility condition (6.30) is slightly simpler than in the linear case (3.12) since we do not have an analogue to the measure in the morphological case and taking an infimum (or supremum) is always invariant, i.e.

$$\inf_{p \in G/H} (g \cdot f)(p) = \inf_{p \in G/H} f(p)$$

for all $g \in G$.

It is also straightforward to show that the operators of Definition 6.6 are equivariant.

LEMMA 6.10 (Morphological convolution is equivariant) In the setting of Definition 6.6 it holds that

$$g \cdot (k \square_{G/H} f) = k \square_{G/H} (g \cdot f)$$

for all $g \in G$.

Proof. Take $g \in G$ and $p \in G/H$, then

$$\begin{aligned}
\left(g \cdot (k \,\square_{G/H}\, f)\right)(p) &= (k \,\square_{G/H}\, f)(g^{-1}p) \\
&= \inf_{q \in G/H} \left\{(g_{g^{-1}p} \cdot k)(q) + f(q)\right\} \\
&= \inf_{q \in G/H} \left\{(g^{-1} g_p \cdot k)(q) + f(q)\right\} \\
&= \inf_{q \in G/H} \left\{k(g_p^{-1} g q) + f(q)\right\} \\
(q'=gq \text{ and invariance of inf}) \qquad &= \inf_{q' \in G/H} \left\{k(g_p^{-1} q') + f(g^{-1}q')\right\} \\
&= \left(k \,\square_{G/H}\, (g \cdot f)\right)(p).
\end{aligned}$$

□

Naturally we also need a kernel to do the morphological convolution with. The kernel of interest is closely related to the PDE's scalar Lagrangian as in (6.28).

DEFINITION 6.7 (Morphological kernel) Let $\alpha > 1$ and let G/H be a homogeneous space of a Lie group G equipped with a G-invariant Riemannian metric $\mathcal{G}$, then we define the *morphological kernel* $k_t^\alpha : M \to \mathbb{R}$ as

$$k_t^\alpha(p) := \frac{t}{\beta}\left(\frac{d_{\mathcal{G}}(p_0, p)}{t}\right)^\beta \tag{6.31}$$

for all $t > 0$ and where $\beta > 1$ relates to α as per (6.27).

Note that we do not include the dependence of k_t^α on the Riemannian metric $\mathcal{G}$ in the notation since it is normally clear which metric we are using but this dependence should still be kept in mind.

Since we required the Riemannian metric $\mathcal{G}$ to be G-invariant it follows that $p \mapsto \mathrm{d}_{\mathcal{G}}(p_0, p)$ is G-invariant, which in particular implies that for all $h \in G_{p_0}$ the following holds

$$\mathrm{d}_{\mathcal{G}}(p_0, p) = \mathrm{d}_{\mathcal{G}}(h \cdot p_0, h \cdot p) = \mathrm{d}_{\mathcal{G}}(p_0, h \cdot p).$$

From which it follows that the morphological kernel itself satisfies the compatibility condition (6.30).

COROLLARY 6.11 (Morphological kernel compatibility) The morphological kernel from Definition 6.7 satisfies (6.30), i.e. $h \cdot k_t^\alpha = k_t^\alpha$ for all $h \in H$, $t > 0$ and $\alpha > 1$.

This gives us all the elements to state our claim that using this kernel with morphological convolution solves the dilation/erosion PDE (6.23).

THEOREM 6.12 (Morphological solution) Let G/H be a homogeneous space of a Lie group G of polynomial growth, let $\alpha > 1$ and let $u_0 : G/H \to \mathbb{R}$ be Lipschitz. Let $\mathcal{G}_2$ denote either $\mathcal{G}_2^+$ or $\mathcal{G}_2^-$ depending on whether we are solving the dilation or erosion variant of (6.23). Then $u_3 : G/H \times (0, \infty) \to \mathbb{R}$ given by

$$u_3(\cdot, t) = k_t^\alpha \square u_0 \tag{6.32}$$

solves the erosion variant of (6.23) (i.e. the '-' variant) while

$$u_3(\cdot, t) = -\left(k_t^\alpha \square (-u_0)\right) \tag{6.33}$$

solves the dilation variant of (6.23) (i.e. the '+' variant) where the kernel k_t^α is based on the appropriate metric $d_{\mathcal{G}_2}$. The solutions are in the sense of BALOGH, ENGULATOV, HUNZIKER, and MAASALO (2012, Thm. 2.1) and are both Lipschitz themselves.

Proof. The Riemannian manifold $(G/H, \mathcal{G}_2)$ is a proper length space and therefore the theory of BALOGH, ENGULATOV, HUNZIKER, and MAASALO (2012) applies. Moreover, since G is of polynomial growth, we have that G/H is a doubling space as per MAHEUX (1998, Thm. 2.17) and also admits a Poincaré constant MAHEUX (1998, Thm. 2.18). Hence we satisfy the additional requirements of BALOGH, ENGULATOV, HUNZIKER, and MAASALO (2012, Thm. 2.3 (vii) and (viii)).

The (scalar) Hamiltonian, as used in BALOGH, ENGULATOV, HUNZIKER, and MAASALO (2012), is given by (6.28) and is indeed superlinear, convex and satisfies $\mathcal{H}_\alpha^{1\mathrm{D}}(0) = 0$. The corresponding (scalar) Lagrangian is also given by (6.28), according to BALOGH, ENGULATOV, HUNZIKER, and MAASALO (2012), the solution — in

the sense of their Theorem 2.1 — to the erosion variant of (6.23) is given by

$$\begin{aligned} u_3(p,t) &= \inf_{q\in G/H}\left\{t\mathcal{L}^{1\mathrm{D}}_{\alpha}\left(\frac{d_{\mathcal{G}_2}(p,q)}{t}\right)+u_0(q)\right\}\\ &= \inf_{q\in G/H}\left\{\frac{t}{\beta}\left(\frac{d_{\mathcal{G}_2}(g_p p_0,q)}{t}\right)^{\beta}+u_0(q)\right\}\\ \text{(G–invariance of $\mathcal{G}_2$)}\quad &= \inf_{q\in G/H}\left\{\frac{t}{\beta}\left(\frac{d_{\mathcal{G}_2}(p_0,g_p^{-1}q)}{t}\right)^{\beta}+u_0(q)\right\}\\ &= \inf_{q\in G/H}\left\{(g_p\cdot k^{\alpha}_t)(q)+u_0(q)\right\}\\ &= \left(k^{\alpha}_t \square u_0\right)(p). \end{aligned}$$

The dilation variant of (6.23) is proven analogously. □

Moreover, as a consequence of Theorem 2.1(ii) in Balogh, Engulatov, Hunziker, and Maasalo (2012) the kernels k^{α}_t satisfy the *semigroup property* as one would expect since they solve an evolution equation.

Corollary 6.13 For all $\alpha > 1$ the morphological kernels from Definition 6.7 satisfy

$$k^{\alpha}_t \square k^{\alpha}_s = k^{\alpha}_{t+s},$$

for all $t, s > 0$. That is to say: they forms a semigroup together with the morphological convolution.

solution concept. The solution concept used in Theorem 6.12 is that proposed by Balogh, Engulatov, Hunziker, and Maasalo (2012). This solution concept is (potentially) different from that of the strong, weak or viscosity solution. Their point of departure is to replace the norm of the gradient with a *metric subgradient*, i.e. we replace $\|\nabla_{\mathcal{G}_2} u_3\|_{\mathcal{G}_2}$ by

$$|\nabla u_3|(p,t) := \limsup_{q\to p}\frac{\max\{u_3(p,t)-u_3(q,t),0\}}{d_{\mathcal{G}_2}(p,q)},$$

and we obtain a solution in terms of this slightly different notion of a gradient. See Balogh, Engulatov, Hunziker, and Maasalo (2012, §2.1) for details.

THE CASE $\alpha = 1$. When we set $\alpha = 1$ we lose the superlinearity of the Hamiltonian $\mathcal{H}_\alpha$ and can no longer apply the approach from Balogh, Engulatov, Hunziker, and Maasalo (2012).

The solutions for $\alpha > 1$ do converge pointwise to the solution for $\alpha = 1$ as $\alpha \downarrow 1$. However the solution concept changes from that of one in terms of a *metric subgradient* to that of a *viscosity* solution (Azagra, Ferrera, and López-Mesas, 2005; Dragoni, 2007). In the general Riemannian homogeneous space setting the result by Azagra, Ferrera, and López-Mesas (2005, Thm. 6.24) applies. It states that viscosity solutions of Eikonal PDEs on complete Riemannian manifolds are given by the distance map departing from the boundary of a given open and bounded set. As Eikonal equations directly relate to geodesically equidistant wavefront propagation on manifolds (Rund, 1966, §3; Bekkers, Duits, Mashtakov, and Sanguinetti, 2015, §4, app. E; Evans, 2022) one expects that we can solve (6.23) for $\alpha = 1$ using the kernel that results from taking the limit $\alpha \downarrow 1$ as follows.

Definition 6.8 (Morphological kernel for $\alpha = 1$) In the same setting as Definition 6.7 we let

$$k_t^1(p) := \lim_{\alpha \downarrow 1} k_t^\alpha(p) = \begin{cases} 0 & \text{if } d_{\mathcal{G}}(p_0, p) \leq t \\ \infty & \text{if } d_{\mathcal{G}}(p_0, p) > t. \end{cases} \tag{6.34}$$

Compare this kernel against the tropical indicator function (5.17), applying morphological convolution with (6.34) as in Theorem 6.12 comes down to doing max or min pooling over a Riemannian ball of a certain radius that is proportional to the time t in the fashion we discussed in §5.3. Since morphological convolution with k_t^1 constitutes conventional *hard* max/min pooling, we can think of dilation and erosion with $\alpha > 1$ as *soft* max respectively min pooling.

In many *matrix* Lie group quotients, such as the Heisenberg group (Manfredi and Stroffolini, 2002) or in $\mathbb{M}_2$, morphological

convolution with the above kernel would indeed provide a viscosity solution. One can describe G-invariant vector fields via explicit coordinates and transfer *Hamilton-Jacobi-Bellman* (HJB) systems on G/H directly towards HJB-systems on $\mathbb{R}^n$ or $\mathbb{R}^{n_1} \times S^{n_2}$, with $n = n_1 + n_2 = \dim(G/H)$. Then one can directly apply results by DRAGONI (2007, Thm. 4) and deduce that applying morphological convolution with the kernel (6.34) solves the PDE (6.23) for $\alpha = 1$ in the same manner as Theorem 6.12 does for $\alpha > 1$. Details are left for future work and we will contend ourselves with the results we obtained for $\alpha > 1$ for now.

$\mathbb{R}$ AS A REPRESENTATIVE EXAMPLE. To get an idea of how the kernel in (6.31) operates in conjunction with morphological convolution we take $G = G/H = \mathbb{R}$ and see how the operation evolves simple data, the kernels and results at $t = 1$ are shown in Figure 6.5. Observe that with α close to 1 (kernel and result in red) we obtain what amounts to an equivariant version of max/min pooling in the fashion of §5.3.

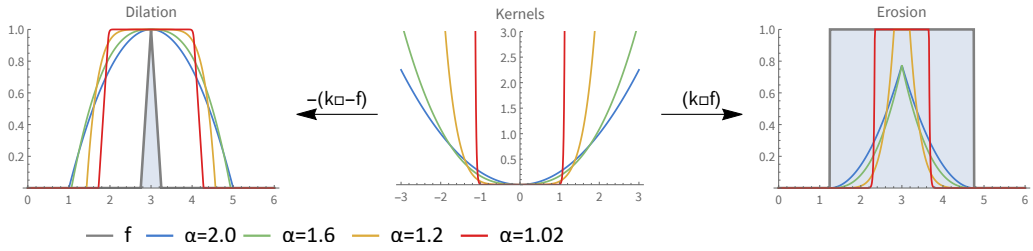


FIGURE 6.5: In the center we have kernels of the type (6.31) on $\mathbb{R}$ (or the signed distance on a manifold of choice) for some $\alpha \geq 1$ and $t = 1$, which solves dilation/erosion. For $\alpha \to 1$ this kernel converges to the type in (6.34), i.e. the solution is obtained by max/min pooling. On the left we morphologically convolve a spike (in gray) with a few of these kernels, we see that if $\alpha \to 1$ we get max pooling, conversely we can call the case $\alpha > 1$ *soft* max pooling. On the right we similarly erode a plateau (in gray), which for $\alpha \to 1$ yields min pooling. The effects of these operations in the image processing context can also be seen in the last two columns of Figure 6.1.

The level sets of the kernels k_t^α for $\alpha > 1$ are of the same shape as for the distance map, see Figure 6.3 (left). For $\alpha = 1$ these are the windows or the stencils over which we would perform pooling as in (5.11). The level sets in Figure 6.3 are balls in $G/H = \mathbb{M}_2$ that do not depend on α. It is only the growth of the kernel values when passing through these level sets that depends on α. As such the example $G/H = \mathbb{R}$ and Figure 6.5 is very representative to the general G/H case. For general homogeneous spaces Figure 6.5 still applies when one replaces the horizontal $\mathbb{R}$-axis with a signed distance along a minimizing geodesic passing through the origin or point of reference. The parameter α can be understood as regulating the *softness* of the max/min pooling.

APPROXIMATE KERNEL. Of course we still cannot conveniently calculate the distance map $d_{\mathcal{G}_2}(p_0, \cdot)$ on $\mathbb{M}_2$ or homogeneous spaces in general and instead substitute our logarithmic metric estimate (6.4) as we did for the heat kernel.

DEFINITION 6.9 (Approximate morphological kernel) Let $\mathcal{G}$ be a G-invariant Riemannian metric on homogeneous space G/H and $\rho_{\mathcal{G}}$ its associated logarithmic metric estimate as per Definition 6.1. Then we define the *approximate morphological kernel* for $\alpha > 1$ as

$$k_t^{\alpha,\mathrm{appr}}(p) := \frac{t}{\beta}\left(\frac{\rho_{\mathcal{G}}(p)}{t}\right)^\beta \tag{6.35}$$

and for $\alpha = 1$ as

$$k_t^{1,\mathrm{appr}}(p) := \begin{cases} 0 & \text{if } \rho_{\mathcal{G}}(p) \leq t, \\ \infty & \text{if } \rho_{\mathcal{G}}(p) > t, \end{cases} \tag{6.36}$$

for all $p \in G/H$. Recall that we relate $\alpha, \beta > 1$ as per (6.27).

Since we have already established in Theorem 6.5 that the logarithmic metric estimate satisfies the kernel compatibility condition (6.30) it follows that the approximate morphological kernel also satisfies this condition.

COROLLARY 6.14 (Approximate morphological kernel compatibility) The approximate morphological kernel from Definition 6.9 satisfies (6.30), i.e. $h \cdot k_t^{\alpha,\text{appr}} = k_t^{\alpha,\text{appr}}$ for all $h \in H$, $t > 0$ and $\alpha \geq 1$.

Unlike the approximate heat kernel from Definition 6.5 the approximate morphological kernel only has a single level of approximation: replacing the true distance map with the logarithmic estimate. This makes bounding the approximate morphological kernel with the exact ones straightforward by substituting Lemma 6.3 into the definition since

$$x \mapsto \frac{t}{\beta}\left(\frac{x}{t}\right)^{\beta}$$

is monotonic for $x, t > 0$.

COROLLARY 6.15 Let G/H be a homogeneous space of a Lie group G of polynomial growth. Let $\alpha > 1$, then for all compact neighborhoods of $p_0 \equiv H$ there exists a $C > 1$ so that for all $t > 0$ we have

$$k_t^{\alpha}(p) \leq k_t^{\alpha,\text{appr}}(p) \leq C k_t^{\alpha}(p) \tag{6.37}$$

for all $p \in G/H$. The constant C depends on α, the choice of compact neighborhood and the Riemannian metric as in Theorem 6.3. For the case $\alpha = 1$, the approximation is exact in the inner and outer regions:

$$k_t^{1,\text{appr}}(p) = k_t^{1}(p) = \begin{cases} 0 & \text{if } \rho_{\mathcal{G}}(p) \leq t \\ \infty & \text{if } d_{\mathcal{G}}(p_0, p) > t, \end{cases} \tag{6.38}$$

but in the intermediate region where $\rho_{\mathcal{G}}^{\beta} > t$ and $d_{\mathcal{G}}(p_0, \cdot) \leq t$ we have that $k_t^{\alpha,\text{appr}} = \infty$ while $k_t^{\alpha} = 0$.

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

COROLLARY 6.16 Let G/H be a homogeneous space of a Lie group G of polynomial growth. Let $\alpha \geq 1$, then for all compact neighborhoods of $p_0 \equiv H$ there exists a $C > 1$ so that for all $t > 0$ we have

$$k_t^{\alpha}(p) \leq k_t^{\alpha,\text{appr}}(p) \leq k_{Ct}^{\alpha}(p) \tag{6.39}$$

for all $p \in G/H$. The constant C depends on α, the choice of compact neighborhood and the Riemannian metric as in Theorem 6.3.

TEMPORAL RESCALING. With convection and diffusion we could trade scaling of the convection vector field respectively Riemannian metric against scaling time, the same holds for the morphological terms. For just this section we reintroduce the dependence of the (approximate) morphological kernel on the metric $\mathcal{G}$ into the notation: let $k_{t,\mathcal{G}}^{\alpha,\text{appr}}$ be the approximate morphological kernel induced by the Riemannian metric $\mathcal{G}$ for some $\alpha \geq 0$. Then we have for all $t \geq 0$ and $\lambda > 0$ that

$$k_{t,\mathcal{G}}^{\alpha,\text{appr}} = \frac{t}{\beta}\left(\frac{\rho_{\mathcal{G}}}{t}\right)^{\beta} = \frac{1}{\beta}\left(\frac{t^{-1/\alpha}\rho_{\mathcal{G}}}{1}\right)^{\beta} = k_{1,t^{-1/\alpha}\mathcal{G}}^{\alpha,\text{appr}},$$

since Corollary 6.2 holds. From which we conclude — as in the convection and diffusion case — that choosing an evolution end-time is perfunctory and we may as well fix it at time $t = 1$ and use only the Riemannian metric $\mathcal{G}$ to control how much dilation and/or erosion we wish to perform.

With this in mind we end our theoretical results and move on to introducing a neural network architecture that incorporates the operations we have discussed so far.

6.9 THE GENERAL PDE LAYER

Now we finally get back to neural networks. We propose to take a group convolution layer as in Figure 5.9 but do away with the convolutions and activation function and instead evolve the incoming feature maps as per the PDE (6.2). We then still make point-wise linear combinations of feature maps as before. Since

we constructed (6.2) to be equivariant with respect to the relevant Lie group it follows that such a layer is still equivariant, just like a group convolution layer (5.8) is. The architecture of such a *PDE layer* is illustrated in Figure 6.6. Next we will detail the elements depicted in that figure.

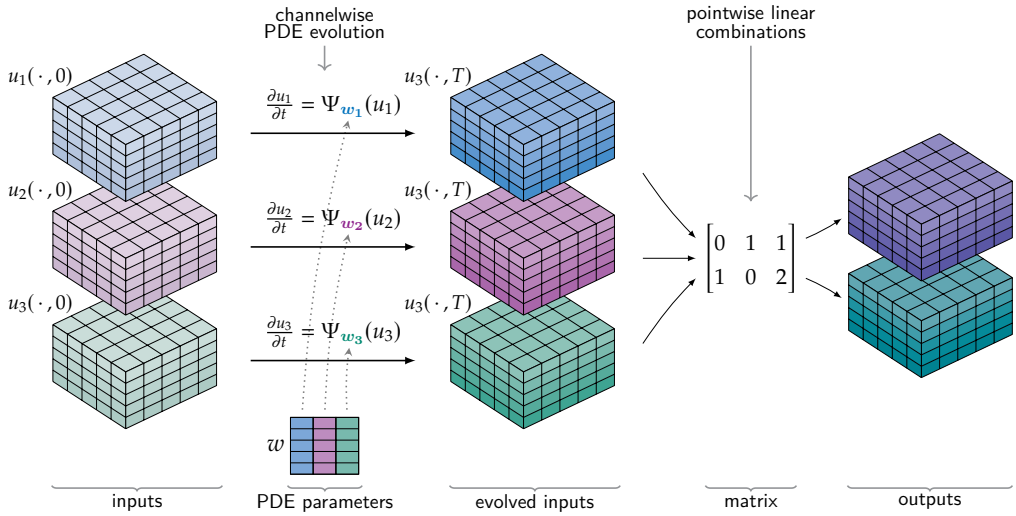


FIGURE 6.6: An example of a *PDE layer*. The convolutions, pooling operations and activation functions are removed from a single channel convolution layer (as in Figure 5.1 or 5.9) and replaced with an evolution PDE (6.2). The PDE has a certain amount of parameters as part of its generator, labeled Ψ here, each channel has its own independent parameters. Together with the coefficients of the pointwise linear combinations the PDE parameters form the trainable parameters of the layer.

FOR HOMOGENEOUS SPACES IN GENERAL. Say that we are working on a homogeneous space G/H. Let $C_{\text{in}}, C_{\text{out}} \in \mathbb{N}$ be the number of input respectively output channels of the layer and let $u_i^{\text{in}} : G/H \to \mathbb{R}$ for $i = 1 \dots C_{\text{in}}$ be the input feature maps. We use the input feature maps as the initial conditions to the PDE (6.2) where each channel has its own independent set of parameters. Since we have established previously that all terms of (6.2) have a temporal rescaling property we can fix the evolution time of

the PDE at time $T = 1$ without losing any potential solutions. We then process each input feature map in substeps according to the operator splitting scheme from §6.3.

TIMESTEPS. In a scientific computing setting we would naturally want to take multiple timesteps, iterating over each of the substeps of the operator splitting approach until we reach our target stopping time. Indeed we would expect the accuracy of the numerical result to improve as we choose smaller timesteps and we would normally choose a timestep as small as our computational budget allows. In a deep learning setting we are however so severely computationally constrained — due to having to repeatedly recompute solutions during the training process — that we want to take *as few timesteps as possible*.

Hence, in experiments we only ever do a *single* timestep, applying each substep only a single time. While this may make for a less than accurate numerical approximation it does not degrade the performance of a PDE-based neural network in our experiments. Indeed, increasing the number of timesteps had a barely measurable effect on the performance of the networks while drastically increasing their computational requirements. We will therefor proceed under the assumption that we only wish to take a single timestep.

CONVECTION SUBSTEP. We have established that equivariant convection can be solved as per Theorem 6.1. This means that we are looking to train a $g \in G$, i.e. the $\gamma_{\boldsymbol{c}}(-t)$ for $t = 1$ in Theorem 6.1. The parametrization of g can be freely chosen, but if the exponential map is surjective then it is convenient to use it. Say $X_1, \dots, X_n \in T_eG$ is a basis for T_eG with $n_1 = \dim(G)$ and let $\boldsymbol{w}_{\text{conv},i} = (w^1_{\text{conv},i}, \dots, w^{n_1}_{\text{conv},i}) \in \mathbb{R}^{n_1}$ be the trainable convection parameters for each $i = 1, \dots, C_{\text{in}}$. Then we take the first substep as follows:

$$u_i^{(1)}(p) := u_i^{\text{in}}\left(g_p \exp_G\left(\sum_{j=1}^{n_1} w^j_{\text{conv},i} X_i\right) \cdot p_0\right) \tag{6.40}$$

per Theorem 6.1 for each input channel $i = 1, \dots, C_{\text{in}}$.

DIFFUSION SUBSTEP. The results $u_i^{(1)} : G/H \to \mathbb{R}$ of the first substep are the starting points for the next substep, being diffusion. We restrict ourselves to non-fractional diffusion $\nu = 1$ here since that is the case for which we have an easy-to-compute approximation. We perform (approximate) diffusion to the nominal time $t = 1$ by convolution (6.13) with the approximate heat kernel $K_1^{\nu,\text{appr}}$ from Definition 6.3 to generate the next intermediate results.

The kernel depends on the G-invariant Riemannian metric $\mathcal{G}_1$ in (6.2). G-invariant Riemannian metrics have finite degrees of freedom and so we can parametrize them with a finite number of parameters. We may opt to further restrict the Riemannian metric beyond being G-invariant, for example by only allowing Riemannian metrics that are diagonal with respect to certain preferred vector fields. Say we have chosen a family of G-invariant Riemannian metrics parametrized by $n_2 \in \mathbb{N}$ parameters which we label as $\boldsymbol{w}_{\text{diff},i} = (w^1_{\text{diff},i}, \dots, w^{n_2}_{\text{diff},i}) \in \mathbb{R}^{n_2}$ for each channel $i = 1, \dots, C_{\text{in}}$.

These parameters then form the trainable parameters of the diffusion substep. If we include the dependency of the heat kernels on the metric parameters explicitly in the notation by writing the kernel as $K_1^{1,\text{appr}}(\boldsymbol{w}_{\text{diff},i})$ then the diffusion substep is performed as

$$u_i^{(2)} := K_1^{1,\text{appr}}(\boldsymbol{w}_{\text{diff},i}) *_{G/H} u_i^{(1)} \tag{6.41}$$

for each $i = 1, \dots, C_{\text{in}}$.

DILATION & EROSION SUBSTEPS. The dilation and erosion substeps work essentially the same as the diffusion substep in that they depend on G-invariant metrics $\mathcal{G}_2^+$ respectively $\mathcal{G}_2^-$. Using the same parametrization of the Riemannian metrics as before we have a set of parameters $\boldsymbol{w}_{\text{dil},i} = (w^1_{\text{dil},i}, \dots, w^{n_2}_{\text{dil},i}) \in \mathbb{R}^{n_2}$ to control the dilations and a set $\boldsymbol{w}_{\text{ero},i} = (w^1_{\text{ero},i}, \dots, w^{n_2}_{\text{ero},i}) \in \mathbb{R}^{n_2}$ to control the erosions for each channel.

Solving the dilation involves the approximate morphological kernel from Definition 6.9 using Theorem 6.12. The morphological kernel once again depends on the Riemannian metric and so on the parameters $\boldsymbol{w}_{\text{dil},i}$, which we make explicit again by writing

the (approximate) kernel as $k_t^{\alpha,\text{appr}}(\boldsymbol{w}_{\text{dil},i})$ for each channel $i = 1, \ldots, C_{\text{in}}$. The dilation substep is then computed as

$$u_i^{(3)} := -\left(k_1^{\alpha,\text{appr}}(\boldsymbol{w}_{\text{dil},i}) \,\square_{G/H}\, \left(-u_i^{(2)}\right)\right). \tag{6.42}$$

The erosion substep is similarly computed using Theorem 6.12 as

$$u_i^{(4)} := k_1^{\alpha,\text{appr}}(\boldsymbol{w}_{\text{ero},i}) \,\square_{G/H}\, u_i^{(3)}, \tag{6.43}$$

which completes the PDE-based processing of the PDE layer.

THE α PARAMETER. The value of $\alpha \geq 1$ we consider a non-trainable hyper-parameter that is part of the design of a given network. In the experiments that we discuss in the next chapter we found empirically that values around $\alpha = 1.3$ gave the best result. This is likely the result of the morphological kernel having a good trade-off between smoothness and sharpness around that value, see Figure 6.5.

LINEAR COMBINATIONS. The final step of the PDE layer is making pointwise linear combinations of the C_{in} processed input feature maps to make the C_{out} output feature maps. The coefficients $(a_{ij}) \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$ of the linear combinations are part of the trainable parameters of the layer. Optionally, biases $(b_i) \in \mathbb{R}^{C_{\text{out}}}$ can be include as well. The final output of the PDE layer is then computed as

$$u_i^{\text{out}} := \sum_{j=1}^{C_{\text{in}}} a_{ij}\, u_j^{(4)} + b_i \tag{6.44}$$

for all $i = 1, \ldots, C_{\text{out}}$.

We can group the trainable parameters of the substeps (6.40) (6.41) (6.42) (6.43) together with the linear combination coefficients and biases as

$$\boldsymbol{w} := \left(\boldsymbol{w}_1^{\text{conv}}, \boldsymbol{w}_1^{\text{diff}}, \boldsymbol{w}_1^{\text{dil}}, \boldsymbol{w}_1^{\text{ero}}, \ldots, \boldsymbol{w}_{C_{\text{in}}}^{\text{conv}}, \boldsymbol{w}_{C_{\text{in}}}^{\text{diff}}, \boldsymbol{w}_{C_{\text{in}}}^{\text{dil}}, \boldsymbol{w}_{C_{\text{in}}}^{\text{ero}}, (a_{ij}), (b_i)\right).$$

Training of the layer then comes down to performing gradient descent on $\boldsymbol{w} \in \mathbb{R}^{C_{\text{in}}(n_1+3n_2+C_{\text{out}})+C_{\text{out}}}$.

NONLINEARITY. Not all four terms of (6.2) need to be used all the time. Which of the substeps (6.40)(6.41)(6.42)(6.43) are included is a design choice that may depend on the application under consideration. The main caveat being that at least one morphological substep needs to be present to avoid the layer as a whole being a linear operator since convection and (non-fractional) diffusion are linear.

6.10 THE $\mathbb{M}_2$ PDE LAYER

To close this chapter we detail how the PDE layer from the previous section §6.9 is constructed for our primary case of interest: the homogeneous space $\mathbb{M}_2$.

CONVECTION SUBSTEP. On $\mathbb{M}_2$ — or any principal homogeneous space — the degrees of freedom of the G-invariant (i.e. left invariant) vector fields equals the number of dimensions of the space. This means we have 3 degrees of freedom (the $n_1 \in \mathbb{N}$ from above) in the convection substep (6.40). The natural parametrization to use is the exponential map (4.8) from §4.3 and compute the convection substep as

$$u_i^{(1)}(p) := u_i^{\text{in}}\left(p \exp_{\text{SE}(2)}\left(c_i^1 A_1 + c_i^2 A_2 + c_i^3 A_3\right)\right) \tag{6.45}$$

for all $p \in \mathbb{M}_2 \equiv \text{SE}(2)$ and where $u_i^{\text{in}} : \mathbb{M}_2 \to \mathbb{R}$ are the input feature maps for all $i = 1, \ldots, C_{\text{in}}$. Recall that C_{in} is the number of input channels. Taken together, there are $3C_{\text{in}}$ trainable parameters here that we label $\boldsymbol{w}^{\text{conv}} := \left(c_i^1, c_i^2, c_i^3\right)_{i=1}^{C_{\text{in}}}$. Note that these parameters are unconstrained, i.e. any value in $\mathbb{R}$ is valid.

Unconstrained parameters are preferable when using gradient descent since no additional checks need to be made to ensure the parameters remain in the constrained range after updates.

METRIC BASED SUBSTEPS. The diffusion (6.41), dilation (6.42) and erosion (6.43) substeps are all three dependent on their underlying SE(2)-invariant Riemannian metrics. We saw in §4.2 that these metrics have 6 degrees of freedom. The general form of a SE(2)-invariant Riemannian metrics (4.6) uses a symmetric positive definite 3×3 matrix to represent it. With the matrix being symmetric we might be tempted to use the 6 upper or lower triangle coefficients as parameters, but these are constrained in a non-trivial manner if the matrix is to be positive definite. The

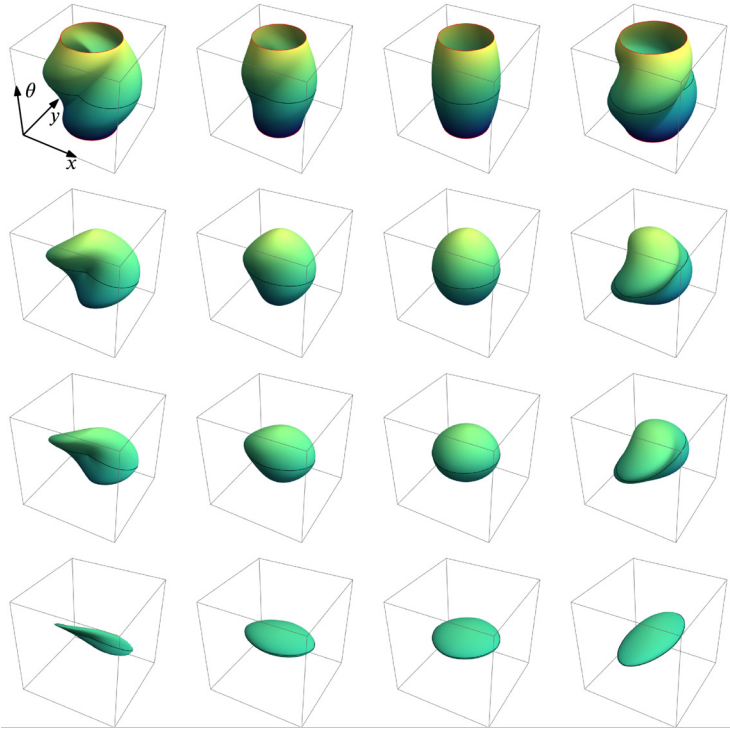


FIGURE 6.7: Unit spheres in $\mathbb{M}_2$ for different SE(2)-invariant diagonal metrics (6.8). These shapes are essentially what is being trained for the diffusion, dilation and erosion substeps and control to what degree and in what directions those substeps perform their action.

unconstrained parameterization scheme of symmetric positive definite matrices by PINHEIRO and BATES (1996) is then more convenient in a deep learning context.

For our experiments — in the next chapter — we further restricted the metrics to be diagonal with respect to our preferred frame as per (6.8) as

$$\mathcal{G} = \sum_{i=1}^{3} w_i^2 \, \omega^i \otimes \omega^i.$$

Here the actual metric coefficients w_1^2, w_2^2, w_3^2 are always non-negative courtesy of the square. The only remaining constraint is that the three parameters w_1, w_2, w_3 all have to be non-zero but this is easy to check and enforce during training by setting a small

minimum value. Figure 6.7 shows some unit balls on $\mathbb{M}_2$ for these types of diagonal metrics for different values of w_1, w_2, w_3.

Using diagonal metrics we assign a set of three parameters as per input channel for each metric based substep: $\boldsymbol{w}^{\text{diff}}, \boldsymbol{w}^{\text{dil}}, \boldsymbol{w}^{\text{ero}} \in \mathbb{R}^{C_{\text{in}} \times 3}$. When including the coefficients for the pointwise linear combinations and we end up with a total of $C_{\text{in}}(12 + C_{\text{out}})$ parameters, with an extra C_{out} parameters if we include biases.

This concludes our exposition on the design of PDE-based convolutional neural networks. In the next chapter we see how these networks perform in image processing applications.

7

APPLICATIONS OF PDE-BASED CNNS

To demonstrate the viability of PDE-based CNNs we perform two experiments where we compare the performance of PDE-G-CNNs against G-CNNs and classic CNNs. We will be doing vessel segmentation and digit classification problem: two straightforward applications of CNNs. Examples of these two applications are illustrated in Fig. 7.1.

The goal of the experiments is to compare the basic building blocks of the different types of networks in clearly defined feed-forward network architectures. So we test networks of modest size only and do not just aim for the performance that would be possible with large-scale networks.

These experiments were first presented in Smets, Portegies, Bekkers, and Duits (2022).

7.1 IMPLEMENTATION

We implemented our PDE-based operators in an extension to the PyTorch deep learning framework (Paszke et al., 2019). Our package is called *LieTorch* and is open source.

The package is available at https://gitlab.com/bsmetsjr/lietorch and through the *pip* and *conda* package managers.

The operations we have proposed in the paper have been implemented in C++ for CPUs and CUDA for Nvidia GPUs but can

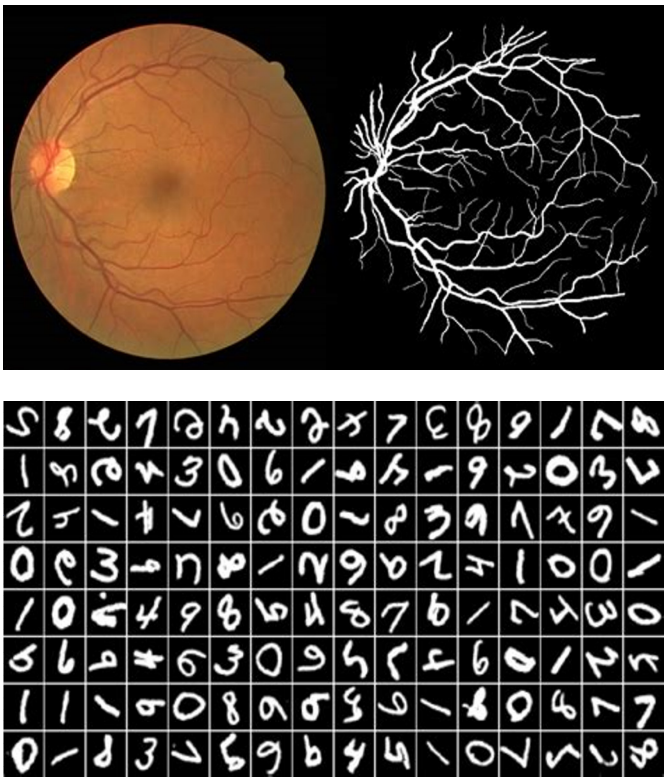


FIGURE 7.1: *Top:* example from the DRIVE dataset (STAAL, ABRÀMOFF, NIEMEIJER, VIERGEVER, and VAN GINNEKEN, 2004), showing a retinal image and its vessel segmentation. *Bottom:* examples from the RotNIST dataset (BAWEJA, 2018).

be used from Python through PyTorch. Our package was also designed with modularity in mind: we provide a collection of PyTorch modules that can be used together to implement the PDE-G-CNNs we proposed but that can also be used separately to experiment with other architectures.

All the modules we provide are differentiable and so our PDE-G-CNNs are trainable through stochastic gradient descent (or its many variants) in the usual manner. In our experiments we have had good results with using the ADAM optimizer from KINGMA and BA (2014).

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

7.2 NETWORK DESIGN CHOICES

Several design choices are common to both experiments, we will go over these in this section.

Firstly, we choose $G = \mathrm{SE}(2)$ and $G/H = \mathbb{M}_2$ for our G-CNNs and PDE-G-CNNs and so go for roto-translation equivariant networks. In all instances we lift to 8 orientations, the lifting layer being constructed as in Figure 5.8 with lifting in the manner of Figure 5.6.

Secondly, we use the convection, dilation and erosion version of the PDE (6.2), thus not incorporating the diffusion term, hence we refer to these networks as PDE-CNNs of the *CDE*-type. Each PDE-layer is implemented as in Figure 6.6 with a single timestep being performed. So no explicit diffusion is used and the layer consists of just resampling and two morphological convolutions. Since we do the resampling using trilinear interpolation this does introduce a small amount of implicit diffusion.

Thirdly, we fix $\alpha = 1.3$. We came to this value empirically; the networks performed best with α-values in the range $1.2 - 1.4$. Looking at Figure 6.5 we can conjecture that $\alpha = 1.3$ is the "sweet spot" between sharpness and smoothness. When the kernel is too sharp (α close to 1) minor perturbations in the input can have large effects on the output, when the kernel is too smooth (α close to 2 or even larger) the output will be smoothed out too much.

Fourth, all our networks are simple feed-forward networks lacking such features as residual connections.

Finally, we use the ADAM optimizer (Kingma and Ba, 2014) together with L^2 regularization uniformly over all parameters with a factor of 0.005.

ROLE OF DIFFUSION. When doing preliminary experiments we found no benefit to adding diffusion to the networks for the applications we are interested in here. Diffusion likely would be of benefit when the input data is noisy but neither datasets we used are noisy. In other applications than classification or segmentation where more regularisation is desirable (such as *inpainting* perhaps) the diffusion term is likely of more use. We leave these investigations for future work.

7.3 RETINAL VESSEL SEGMENTATION

The first experiment uses the *DRIVE* retinal vessel segmentation dataset (STAAL, ABRÀMOFF, NIEMEIJER, VIERGEVER, and VAN GINNEKEN, 2004). The object is the generate a binary mask indicating the location of blood vessels from a color image of a retina as illustrated in the top image of Figure 7.1.

We test 6- and 12-layer variants of a CNN, a G-CNN and a CDE-PDE-CNN. The layout of the 6-layer networks is shown in Figure 7.2, the 12-layer networks simply add more convolution, group convolution or CDE layers. All the networks were trained on the same training data and tested on the same testing data.

The output of the network is passed through a sigmoid function to produce a 2D map a of values in the range $[0, 1]$ which we compare against the known segmentation map b with values in $\{0, 1\}$. We use the continuous DICE coefficient as the loss function:

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

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

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

We measure the performance of the network by the DICE coefficient obtained on the 20 images of the testing dataset. We trained each model 10 times, the results of which are summarized in Table 7.1 and Figure 7.3.

We achieve similar or better performance than CNNs or G-CNNs but with a vast reduction in parameters. Scaling from 6 to 12 layers even allows us to reduce the total number of parameters of the PDE-G-CNN while still increasing performance, this is achieved by reducing the number of channels (i.e. the width) of the network, see also Table 7.2.

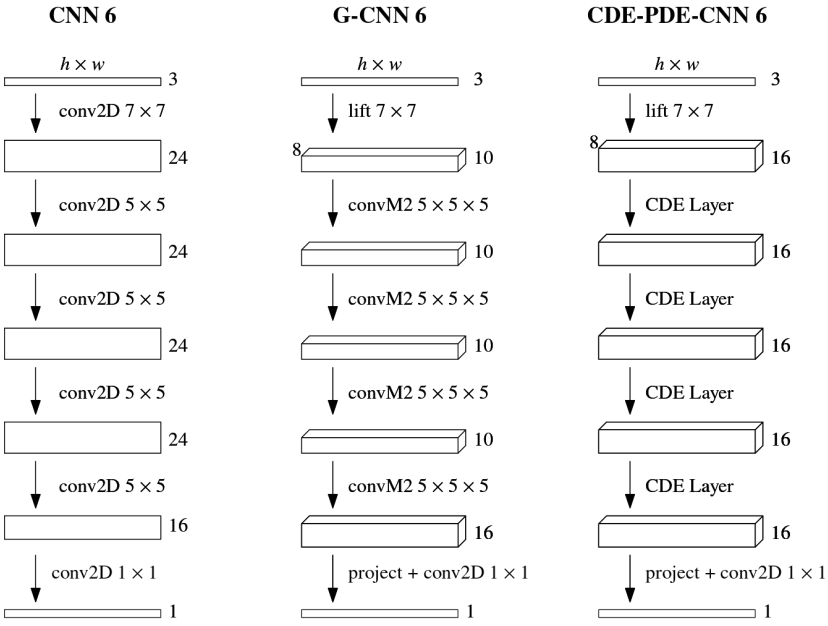


FIGURE 7.2: Schematic of the 6-layer models used on our segmentation experiments. Kernel sizes and number of feature channels in each layer are indicated, depth indicates that the data lives on $\mathbb{M}_2$. Omitted are activation functions, batch normalization, padding and dropout modules. The 12-layer models are essentially the same but with double the number of layers and with reduced number of channels per layer (i.e. reduced width) for the CDE-PDE-CNN (hence the reduction in parameters going from 6 to 12 layers).

7.4 DIGIT CLASSIFICATION

The second experiment we performed is the classic digit classification experiment. Instead of using the plain MNIST dataset we did the experiment on the *RotNIST* dataset (BAWEJA, 2018). RotNIST contains the same images as MNIST but rotated to various degrees. Even though classifying rotated digits is a fairly artificial problem we include this experiment to show that PDE-based CNNs also work in a context very different from the first segmentation experiment. While our choice of PDEs derives from more traditional image processing methods, this experiment shows their utility in a basic image classification context.

Model	Parameters	DICE score ± std.dev.
CNN 6	47,352	0.8058 ±0.0017
G-CNN 6	39,258	0.8085 ±0.0022
CDE-PDE-CNN 6	4,128	0.8115 ±0.0018
CNN 12	129,432	0.8189 ±0.0005
G-CNN 12	114,378	0.8192 ±0.0012
CDE-PDE-CNN 12	3,678	0.8220 ±0.0007

TABLE 7.1: Average DICE coefficient achieved on the 20 images of the DRIVE testing dataset and the number of trainable parameters of each model. The G-CNNs and CDE-PDE-CNNs are roto-translation equivariant by construction. Note the vast reduction in parameters allowed by using PDE-based networks.

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

TABLE 7.2: Allocation of parameters for the 6- and 12-layer CDE-PDE-CNNs used in the vessel segmentation experiment. The added depth of the networks allows us to shrink the overall width. With the network having less channels over all we can also shrink the number of channels in the lifting layer, which drastically reduces the total number of parameters.

We tested three networks: the classic LeNet5 CNN (LECUN et al., 1989) as a baseline, a 4-layer G-CNN and a 4-layer CDE-PDE-CNN. The architectures of these three networks are illustrated in Figure 7.4.

All three networks were trained on the same training data and tested on the same testing data. We train with a learning rate of 0.05 and a learning rate gamma of 0.96. We trained the LeNet5

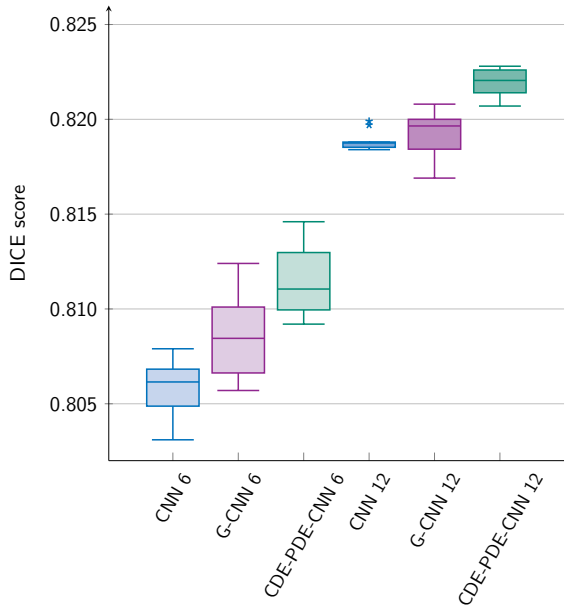


FIGURE 7.3: Comparison of PDE-based CNNs vs. G-CNNs vs. CNNs on the retinal vessel segmentation task. We test 6- and 12-layer variants of conventional CNNs, G-CNNs and our CDE-PDE-CNNs, each network is trained 10 times, the chart shows the distribution of DICE performances on the test dataset, a higher score is better.

model for 120 epochs and the G-CNN and CDE-PDE-CNN models for 60 epochs.

We measure the performance of the network by its accuracy on the testing dataset. We trained each model 10 times, the results of which are summarized in Table 7.3 and Figure 7.5.

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

7.5 COMPUTATIONAL PERFORMANCE

Care was taken in optimizing the LieTorch implementation to show that PDE-based networks can still achieve decent running times despite their higher computational complexity. In Table 7.4

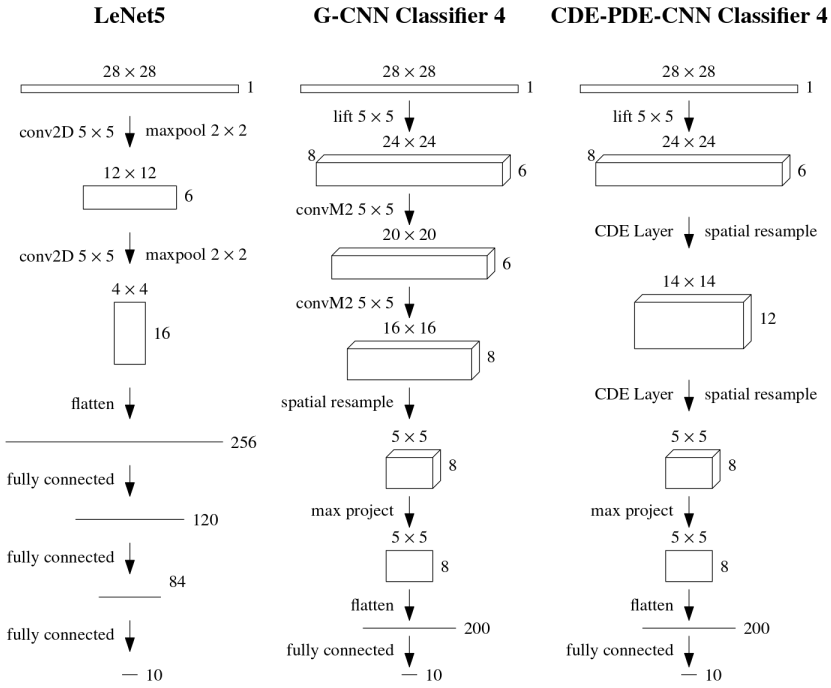


FIGURE 7.4: Schematic of the three models tested with the Rot-NIST data. Kernel sizes and number of feature channels in each layer are indicated. Omitted are activation functions, batch normalization and dropout modules.

Model	Parameters	Error rate ± std.dev.
CNN (LeNet5)	44,426	2.59% ±0.66%
G-CNN Classifier 4	12,700	1.14% ±0.21%
CDE-PDE-CNN Classifier 4	2,542	1.10% ±0.10%

TABLE 7.3: Accuracy of the digit classification models on the Rot-NIST testing dataset and number of parameters for each model.

we summarized the inferencing performance of each model we experimented with.

Our approach simultaneously gives us equivariance, a decrease in parameters and higher performance but at the cost of an increase in flops and memory footprint. While our implementation is reasonably optimized it has had far less development time ded-

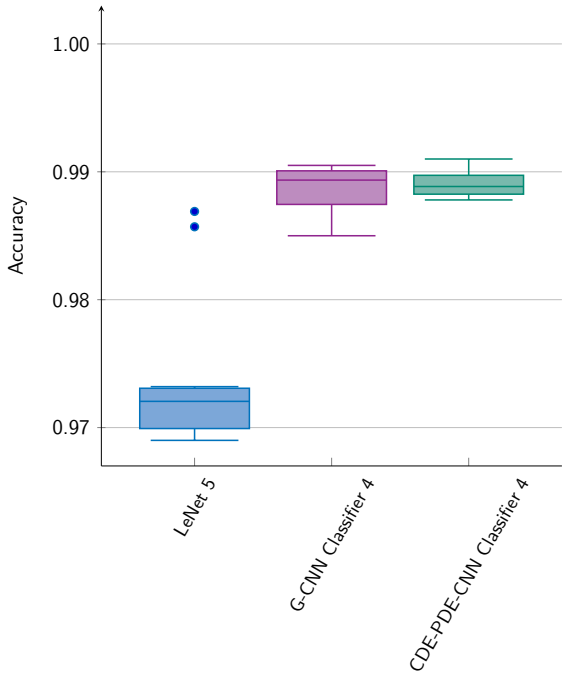


FIGURE 7.5: Performance comparison of digit classification on the RotNIST dataset. We compare the classic 5-layer LeNet against a 4-layer G-CNN and PDE-CNN. LeNet was trained for 120 epochs, the other two for 60 epochs.

icated to it than the traditional CNN implementation provided by PyTorch/cuDNN, so we are confident more performance gains can be found.

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

7.6 CONCLUDING REMARKS

In the previous Chapter 6 we presented the general mathematical framework of geometric PDEs on homogeneous spaces that under-

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

TABLE 7.4: Time in seconds it took to run each model on the testing dataset of its respective experiment. The DRIVE testing dataset contains 20 images while the RotNIST testing dataset contains 10,000 digits.

lies our PDE-G-CNNs. PDE-G-CNNs allow for a geometric and probabilistic interpretation of CNNs opening up new avenues for the study and development of these types of networks. We showed in this chapter that, additionally, PDE-G-CNNs have increased performance with a reduction of parameters.

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

PDE-G-CNNs have three types of layers: convection, diffusion and morphological — dilation & erosion — layers. We have shown that these layers generalize the standard nonlinear operations in CNNs such as max pooling and ReLU activation.

To efficiently evaluate PDE evolution in the layers, we provided tangible analytical approximations to the relevant kernel operators on homogeneous spaces. We have underpinned the quality of the approximations in Theorem 6.8 and Theorem 6.12 of Chapter 6.

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

8

SEMIRING ACTIVATION

In the previous chapters, we built neural networks that lacked the traditional activation function. Instead, we used a class of trainable nonlinear operators based on dilation/erosion PDEs. Specifically, we performed morphological convolutions as per Definition 6.6. While morphological convolution is not linear, that does not mean it is completely lacking in structure. Indeed, let k, f_1, f_2 be some appropriate functions—the domain is not important right now—and let $a_1, a_2 \in \mathbb{R} \cup \{\infty\}$ be some scalars. Then morphological convolution has the property that

$$k \square (\min\{a_1 + f_1,\ a_2 + f_2\}) = \min\{a_1 + (k \square f_1),\ a_2 + (k \square f_2)\}, \quad (8.1)$$

which is structurally similar to normal convolution being linear:

$$k * (a_1 f_1 + a_2 f_2) = a_1 (k * f_1) + a_2 (k * f_2),$$

only with multiplication changed to addition and addition changed to taking the minimum.

Looking at (8.1) as a separate property, we can say that PDE-G-CNNs add 3 ingredients to neural networks:

(i) equivariance,
(ii) trainable operators with the *semiring-linear* property (8.1),
(iii) kernels derived from PDEs to be used in (ii).

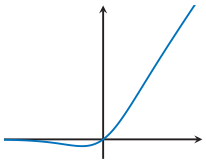
Taking just ingredient (i) gives you (G-)CNNs, taking (i) & (ii) gives you morphological convolutions and finally adding (iii) gives PDE-G-CNNs. In this chapter we wonder about the impact of ingredient (ii) on the final result and how viable it is to replace activation functions with just trainable operators that have a similar *semiring-linear* property as in (8.1). We will make precise what we mean with *semiring-linear* in general later in the chapter.

This chapter covers the material of Smets, Donker, Portegies, and Duits (2024). Before we present *semiring activation* as a framework for building trainable nonlinear operators we first cover related works that introduce similar but distinct ideas.

8.1 RELATED WORK

While neural networks come in a large variety of types and architectures, one common characteristic the majority of them share is the alternation between trainable linear operations (or affine operations if one includes bias) on the one hand and non-trainable nonlinear operators in the form of a scalar activation function on the other hand. Sometimes a multi-variate nonlinear function like max/min-pooling is also used, but this is again a fixed, non-trainable function similar in nature to a scalar activation function. Even in the case of transformers (Vaswani et al., 2017) one trains linear maps that are then composed in a fixed manner through the inner product and a soft-max function. While this binary setup is the de facto standard in machine learning, exceptions exist. These exceptions can be roughly classified into three classes: trainable activation functions, non-standard neurons and morphological operations. We will briefly discuss these three approaches. For a comprehensive overview of trainable activation functions and non-standard neurons see Apicella, Donnarumma, Isgrò, and Prevete (2021).

TRAINABLE ACTIVATION FUNCTIONS. The most straightforward way of obtaining a trainable nonlinearity is using an activation function that has one or more parameters affecting it and train these parameters as part of the neural network. This is typically done either by adding some shape parameter to an existing fixed activation function or building the activation function as an ensemble of some fixed basis functions and have the ensemble coefficients be trainable parameters. An example of the first kind is the *swish* function (Ramachandran, Zoph, and Le, 2017) defined as



The 'swish' function for $\alpha = 1$.

$$\text{swish}_\alpha(x) := x \cdot \text{sigmoid}(\alpha x) = \frac{x}{1 + e^{-\alpha x}},$$

where $\alpha \in \mathbb{R}$ is a parameter that can either be trained or is a

chosen constant. Note that for $\alpha = 1$ the swish function reduces to the SiLU (ELFWING, UCHIBE, and DOYA, 2018) and for $\alpha \to \infty$ reduces to the familiar ReLU.

An example of the second kind is the *adaptive blending unit* (SÜTFELD, BRIEGER, FINGER, FÜLLHASE, and PIPA, 2018) given by

$$\mathrm{ABU}(x) := \sum_{i=1}^{k} \alpha_i \, f_i(x),$$

where the α_i's are the trainable parameters and the functions f_i are a selection of fixed activation functions such as tanh, ReLU, id, swish, etc. In the original study the α_i parameters were initialized with $\frac{1}{k}$ and constrained using a normalization scheme.

As APICELLA, DONNARUMMA, ISGRÒ, and PREVETE (2021) notes, in most cases these trainable activation functions can also be expressed (or approximated) by a small feed-forward neural network with fixed activation functions. This is not surprising since the fixed activation functions remain the core building blocks. As a consequence, the same benefit of using these trainable activation functions can be achieved by simply making the neural network deeper.

NON-STANDARD NEURONS. The alternative to making the activation function trainable is replacing the standard neuronal model (linear map followed by an activation function) altogether. A basic example is a *maxout network* (GOODFELLOW, WARDE-FARLEY, MIRZA, COURVILLE, and BENGIO, 2013) where each neuron has a number of linear maps with scalar codomain and the final output is the maximum of the group.

A more elaborate non-standard neuron is used in an *morphological neural network* as introduced by RITTER and SUSSNER (1996). Here the idea is to replace the inner linear combination in the classic neuron given by

$$y_j = \sigma\left(\textstyle\sum_{i=1}^{n} w_{ij} \, x_i + b_j\right),$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function , with a *tropical* combination given by:

$$y_j = \sigma\left(\max\left\{w_{ij} + x_i \,\middle|\, i = 1 \ldots n\right\} + b_j\right).$$

and where σ is again a choice of scalar activation function. More examples of non-standard neurons can be found in APICELLA, DONNARUMMA, ISGRÒ, and PREVETE (2021).

MORPHOLOGICAL OPERATIONS. Mathematical morphology is a theory and set of techniques for analyzing and processing geometric features, most commonly in images (SERRA, 1982). In fact, the morphological convolution from Definition 6.6 that we have been using throughout takes its name from mathematical morphology. With the rise in popularity of neural networks, many researchers have been exploring ways of integrating ideas from mathematical morphology into them.

The earliest work we are aware of is the *PConv* operator in MASCI, ANGULO, and SCHMIDHUBER (2013), here the morphological convolution is not implemented directly but approximated through counter-harmonic means, which is also the approach taken by MELLOULI, HAMDANI, AYED, and ALIMI (2017). Later work uses other approximations like soft maximum and minimum in SHIH, SHEN, and ZHONG (2019). Both approaches have some trouble when executed in floating point arithmetic, that can however be ameliorated to some degree by a smart choice of bias as in SHEN, ZHONG, and SHIH (2019). Direct computation of morphological convolution is also possible as we did in previous chapters.

In this chapter we look at linear convolution and morphological convolution as two special cases of a more general family of operators where a particular choice of codomain algebra and domain translation equivariance naturally yields a convolution type operator.

8.2 A DIFFERENT APPROACH TO NEURAL NETWORKS

Instead of looking at deep neural networks as consisting of layers of neurons, we can also take the view that we are alternating between linear and nonlinear operators. Sure, we usually choose the nonlinear operator to be a scalar activation function and let the linear operators be the trainable part but that is but one possible design choice.

Suppose we could place the nonlinear operator on an equal footing to the linear one, what would this look like? Clearly, the nonlinear operator needs some sort of structure similar to the linear operators to make this work. If $A : V \to W$ is some map between vector spaces V and W then we say it is linear if

$$A(av_1 + bv_2) = aA(v_1) + bA(v_2)$$

for all $v_1, v_2 \in V$ and scalars a and b. We could require a similar structure of a nonlinear operator $B : X \to Y$, so that for some binary operations $\oplus$ and $\odot$ we have

$$B(a \odot x \oplus b \odot y) = a \odot B(x) \oplus b \odot B(y) \tag{8.2}$$

for all $x_1, x_2 \in X$ and scalars a and b and where $\odot$ takes precedence over $\oplus$. The spaces X and Y would have to be spaces where these operations make sense of course. If this allows for the nonlinear operator B to be written as a matrix, similar to A then we could train B in the same way we train A and effectively have linear and nonlinear operators on the same footing.

This then will be our approach: look at a class of nonlinear operators that are *semiring-linear* as in (8.2) and then instead of building neural networks with trainable linear (or affine) operators A_i and activation functions σ as

$$A_L \circ \sigma \circ A_{L-1} \circ \cdots \circ A_2 \circ \sigma \circ A_1,$$

we build it with linear operators A_i and nonlinear operators B_i as

$$A_L \circ B_{L-1} \circ A_{L-1} \circ \cdots \circ A_2 \circ B_1 \circ A_1,$$

where both types are trainable. We will refer to this idea as *semiring activation*.

COMPATIBILITY WITH EQUIVARIANCE. An advantageous property of this approach is that it is entirely compatible with the notion of *geometric equivariance* insofar we restrict ourselves to scalar codomains. This makes it possible to combine the techniques from this chapter with the equivariance theory from Chapter 3.

8.3 SEMIRINGS & SEMIMODULES

To construct quasilinear operators as in (8.2) we need to have operations $\oplus$ and $\odot$ that have some structure to them. In the linear case we use a *field* (usually $\mathbb{R}$ or $\mathbb{C}$) and a *vector space* (such as $\mathbb{R}^n$ or $C^\infty(M)$ for some manifold M), we will now generalize this by using *semirings* and *semimodules* instead.

SEMIRING. A semiring is an algebraic structure in which we can add and multiply elements, but in which neither subtraction nor division are necessarily possible.

DEFINITION 8.1 (Semiring) A *semiring* is a set R equipped with two binary operations $\oplus$ and $\odot$, called addition and multiplication, such that

(i) addition and multiplication are associative,
(ii) addition is commutative,
(iii) addition has an identity element $\mathbb{0}$,
(iv) multiplication has an identity element $\mathbb{1}$,
(v) multiplication distributes over addition:

$$a \odot (b \oplus c) = a \odot b \oplus a \odot c,$$
$$(a \oplus b) \odot c = a \odot c \oplus b \odot c,$$

(vi) multiplication by $\mathbb{0}$ annihilates:

$$\mathbb{0} \odot a = a \odot \mathbb{0} = \mathbb{0}.$$

Additionally if $a \oplus a = a$ we say the semiring is *idempotent* and if $a \odot b = b \odot a$ we say the semiring is *commutative* or *abelian*.

Some works refer to semirings as rigs, *from rings 'without negatives', hence the missing 'n'.*

Just like with standard multiplication it is conventional to abbreviate $ab \equiv a \odot b$ if there can be no confusion. We also let $\odot$ take precedence over $\oplus$, i.e. $a \odot b \oplus c = (a \odot b) \oplus c$.

All fields are semirings—though not the other way around—and so we can talk about the *linear semiring* as just being $(R, \oplus, \odot, \mathbb{0}, \mathbb{1}) = (\mathbb{R}, +, \cdot, 0, 1)$, which we denote as R_{lin}.

SEMIRINGS OF INTEREST. Semirings, being fairly general, come in a large variety. For our current purpose we are only interested in

semirings that have the real numbers, possibly extended with $\pm\infty$, as their underlying set. This way the new quasilinear operators can coexist with the linear operators since each can deal with the output of the other and we stay in the realm of floating point arithmetic. The new addition and multiplication operations should also be easy enough to compute in practice so as to not incur an unreasonable performance penalty. Therefor we will focus on two families of semirings: the *tropical* and the *logarithmic*.

TROPICAL SEMIRINGS. The first two closely related semiring that we will consider—and already have used implicitly—are the *tropical semirings*. The name 'tropical' refers to *Brazil* where the Hungarian-born mathematician *Imre Simon* lived and worked while studying these semirings.

DEFINITION 8.2 (Tropical semirings) The *min-plus* semiring consists of the set $\mathbb{R} \cup \{\infty\}$ and the operations

$$a \oplus b = \min\{a, b\} \quad \text{and} \quad a \odot b = a + b.$$

The identity element for 'addition' is $\mathbb{0} = \infty$ and for 'multiplication' is $\mathbb{1} = 0$. We write

$$R_{\min} := (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)\,.$$

The *max-plus* semiring consists of the set $\mathbb{R} \cup \{-\infty\}$ and the operations

$$a \oplus b = \max\{a, b\} \quad \text{and} \quad a \odot b = a + b.$$

The identity element for 'addition' is $\mathbb{0} = -\infty$ and for 'multiplication' is again $\mathbb{1} = 0$. We write

$$R_{\max} := (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)\,.$$

Clearly, the min-plus and max-plus semirings are isomorphic through negation $x \mapsto -x$, therefore it is common to refer to either of them as *the* tropical semiring.

Since $\min\{a, b\} = \min\{b, a\}$ and $\min\{a, a\} = a$ the tropical semirings are commutative and idempotent.

LOGARITHMIC SEMIRINGS. The second family of semirings we consider is that of the *logarithmic semirings*.

DEFINITION 8.3 (Logarithmic semirings) Let $\mu \neq 0$ and define

$$a \oplus_\mu b := \frac{1}{\mu} \log\left(e^{\mu a} + e^{\mu b}\right) \quad \text{and} \quad a \odot b := a + b,$$

for all $a, b \in \mathbb{R} \cup \{-\operatorname{sgn}(\mu) \cdot \infty\}$. The identity element for $\oplus_\mu$ is $-\operatorname{sgn}(\mu) \cdot \infty$ and $\odot$ is 0. We write

$$R^\mu_{\log} := \left(\mathbb{R} \cup \{-\operatorname{sgn}(\mu) \cdot \infty\}, \oplus_\mu, +, -\operatorname{sgn}(\mu) \cdot \infty, 0\right)$$

and call this the *logarithmic semiring*.

Here we adopted the convention that $e^{-\infty} = 0$ and correspondingly $\log(0) = -\infty$. Associativity, commutativity and distributivity are easy to check. The annihilation axiom is also satisfied since $\mathbb{0} \odot a = -\operatorname{sgn}(\mu) \cdot \infty + a = -\operatorname{sgn}(\mu) \cdot \infty = \mathbb{0}$.

The logarithmic semiring is closely related to both the linear and tropical semiring. Indeed, the linear semiring R_{lin} is isomorphic via the mapping $x \mapsto e^{\mu x}$ and its inverse $x \mapsto \frac{1}{\mu}\log(x)$. For the addition $\oplus_\mu$ this was already evident from its definition, for multiplication the same is true since

$$a \odot b = \frac{1}{\mu} \log\left(e^{\mu a} e^{\mu b}\right) = \frac{1}{\mu} \log\left(e^{\mu(a+b)}\right) = a + b.$$

The tropical semirings are not isomorphic to the logarithmic semirings but they can be seen as the limits of the logarithmic semirings in the sense of the following lemma.

LEMMA 8.1 (Limits of logarithmic addition) Let $\mu \neq 0$ then

$$\lim_{\mu \to -\infty} a \oplus_\mu b = \min\{a, b\}$$

for all $a, b \in \mathbb{R} \cup \{\infty\}$ and

$$\lim_{\mu \to \infty} a \oplus_\mu b = \max\{a, b\}$$

for all $a, b \in \mathbb{R} \cup \{-\infty\}$.

We will prove the first statement, the proof of the second one goes analogously.

Proof. Let $a, b \in \mathbb{R} \cup \{\infty\}$, and assume $a < b$ then

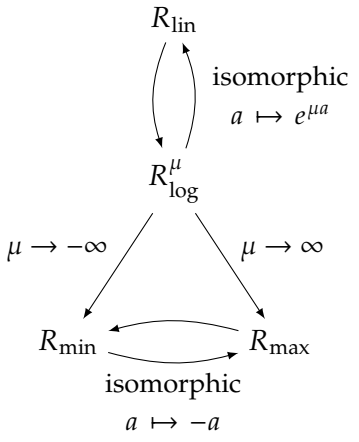
$$\begin{aligned}
\lim_{\mu\to-\infty} a \oplus_\mu b &= \lim_{\mu\to-\infty} \frac{1}{\mu} \log\left(e^{\mu a} + e^{\mu b}\right) \\
&= \lim_{\mu\to-\infty} \frac{1}{\mu} \log\left(e^{\mu a}\left(1 + e^{\mu(b-a)}\right)\right) \\
&= \lim_{\mu\to-\infty} \frac{1}{\mu} \left(\mu a + \log\left(1 + e^{\mu(b-a)}\right)\right) \\
&= a + \lim_{\mu\to-\infty} \frac{1}{\mu} \log\left(1 + e^{\mu(b-a)}\right) \\
&= a.
\end{aligned}$$

Similarly, when we assume $b < a$ we have $\lim_{\mu\to-\infty} a \oplus_\mu b = b$. It remains to check the case $a = b$:

$$\begin{aligned}
\lim_{\mu\to-\infty} a \oplus_\mu a &= \lim_{\mu\to-\infty} \frac{1}{\mu} \log\left(2e^{\mu a}\right) \\
&= \lim_{\mu\to-\infty} \frac{1}{\mu} \left(\log(2) + \mu a\right) \\
&= a.
\end{aligned}$$

We conclude that indeed $\lim_{\mu\to-\infty} a \oplus_\mu b = \min\{a, b\}$. □

We can summarize the relationship between the linear, logarithmic and tropical semirings in the following diagram.



SEMIMODULE. The counterpart to the vector space in the semiring setting is the *semimodule*.

DEFINITION 8.4 (Semimodule) Let $(R, \oplus, \odot, \mathbb{0}, \mathbb{1})$ be a semiring. A left R-semimodule or *semimodule* over R is a commutative monoid $(M, +_M)$ with additive identity 0_M and a map $R \times M \to M$ denoted by $(a, m) \mapsto am$, called *scalar multiplication*, so that for all $a, b \in R$ and $m, m' \in M$ the following hold:

(i) $(a \odot b)m = a(bm)$,

(ii) $a(m +_M m') = am +_M am'$,

(iii) $(a \oplus b)m = am +_M bm$,

(iv) $\mathbb{1}m = m$,

(v) $a0_M = 0_M = \mathbb{0}m$.

Naturally every semiring is a semimodule over itself just like every field is a vector space over itself.

Let R be a semiring and S a non-empty set, then R^S is a semimodule with addition and scalar multiplication defined element-wise: let $f, f' \in R^S$ and $r \in R$ then

$$(f \oplus f')(s) := f(s) \oplus f'(s) \quad \text{and} \quad (r \odot f)(s) := r \odot f(s)$$

for all $s \in S$. Here we used $\oplus$ and $\odot$ for the operations in the semimodule as well since they correspond with the $\oplus$ and $\odot$ operations in the semiring. The additive identity of the semimodule is the constant function $s \mapsto \mathbb{0}$. This generalizes the prototypical vector space $\mathbb{R}^n$ to the semimodule R^n using the notational convention $n \equiv \{1, \ldots, n\}$. For non-finite S we get the generalization of function vector spaces to function semimodules.

SEMIMODULE HOMOMORPHISMS. Now we can formulate a generalization of linear maps in the form of *semimodule homomorphisms*.

DEFINITION 8.5 (Semimodule homomorphism) Let R be a semiring and let X and Y be semimodules over R. Then a map $A : X \to Y$ is an *R-homomorphic map* or an *R-homomorphism* if for all $a, b \in R$ and $f, f' \in X$:

$$A(af + bf') = a(Af) + b(Af'),$$

where on the left the addition and scalar multiplication happens in X and on the right the addition and scalar multiplication happens in Y.

Just like with the definition of linearity the single condition above is equivalent to the following two conditions:

$$A(af) = a(Af') \quad \text{and} \quad A(f + f') = Af + Af'$$

for all $a \in R$ and $f, f' \in X$.

With this definition we can understand 'linear' as homomorphic with respect to the real linear semigroup $(\mathbb{R}, +, \cdot, 0, 1)$. But now, instead of just considering only linear maps we can pick another semiring and consider homomorphisms with respect to this semiring. This allows us to construct equivariant semimodule homomorphic operators in the same fashion as we did with equivariant linear operators in § 3.3.

In fact, the morphological convolutions from Definition 6.6 are exactly those operators that are i) equivariant and ii) homomorphisms with respect to the tropical semiring $R_{\min}$ since they satisfy (8.1).

8.4 TRAINABLE SEMIRING OPERATORS

In the same way we construct linear maps $\mathbb{R}^m \to \mathbb{R}^n$ through matrix multiplication we can now construct a quasilinear map. Let $R \equiv (R, \oplus, \odot, \mathbb{0}, \mathbb{1})$ be some commutative semiring, $x \in R^m$ and $B \in R^{n \times m}$ and define

$$B \odot x = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix} \odot \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} := \begin{bmatrix} b_{11} \odot x_1 \oplus \cdots \oplus b_{1m} \odot x_m \\ \vdots \\ b_{n1} \odot x_1 \oplus \cdots \oplus b_{nm} \odot x_m \end{bmatrix}. \tag{8.3}$$

This is nothing more than the usual matrix-vector multiplication operation when we take the linear semiring $R = (\mathbb{R}, +, \cdot, 0, 1)$. When we further overload the symbols $\oplus$ and $\odot$ to componentwise addition resp. scalar multiplication in R^m, i.e.:

$$a \odot \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} := \begin{bmatrix} a \odot x_1 \\ \vdots \\ a \odot x_m \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \oplus \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} := \begin{bmatrix} x_1 \oplus y_1 \\ \vdots \\ x_m \oplus y_m \end{bmatrix},$$

then it can be verified that B satisfies

$$B \odot (a \odot x \oplus b \odot y) = a \odot (B \odot x) \oplus b \odot (B \odot y). \tag{8.4}$$

This makes $x \mapsto B \odot x$ satisfy (8.2), we say it is a *semiring-linear operator with respect to* R. We can also add a bias in this setting: $x \mapsto B \odot x \oplus c$ with $c \in R^n$.

We can now just treat the coefficients of B as the trainable weights of this operator and treat them like any other weights in the network. This manner of constructing nonlinear operators works for any commutative semiring. We will be applying it for the tropical and logarithmic semirings from Definition 8.2 and Definition 8.3.

8.5 PARAMETER INITIALIZATION

An important aspect of training neural networks is proper initialization of its parameters. The standard parameter initialization schemes like *Xavier* initialization (GLOROT and BENGIO, 2010) and *Kaiming* initialization (HE, ZHANG, REN, and SUN, 2015) are derived specifically around the forward and backward stability of linear maps and so do not apply to semiring-based maps. Consequently, we need to come up with new initialization schemes for the semiring weights, specifically for tropical and logarithmic cases.

TROPICAL. For tropical operators (i.e. max-plus and min-plus) we propose an initialization scheme based on 'fair' backpropagation of gradients. Consider the max-plus operator $R_{\max}^m \to R_{\max}^n$ given by

$$y_i = \max_{j=1\ldots m} w_{ij} + x_j \tag{8.5}$$

where $x_1, \ldots, x_m \in R_{\max}$ are the inputs, $y_1, \ldots, y_n \in R_{\max}$ are the outputs and $[w_{ij}]_{ij} \in R_{\max}^{n \times m}$ are the trainable parameters. Then the partial derivatives are given by

$$\frac{\partial y_i}{\partial x_j} = \begin{cases} 1 & \text{if } j = \arg\max_{k=1\ldots m} w_{ik} + x_k, \\ 0 & \text{else,} \end{cases}$$

for $i = 1 \ldots n$ and $j = 1 \ldots m$. This could be problematic for the backward pass since if an input x_j never 'wins' one of the maxima, i.e. $\frac{\partial y_i}{\partial x_j} = 0$ for all $i = 1, \ldots, m$, then its gradient will always be zero. Conversely, if an input x_j happens to be very large on a consistent basis then it will accumulate all the gradients of all the outputs $y_1, \ldots, y_n$ to itself. The result would be a very unbalanced gradient distribution during the backward pass, something we know from previous research on parameter initialization (Glorot and Bengio, 2010; He, Zhang, Ren, and Sun, 2015) is undesirable.

The extreme case for the operator (8.5) would be having an x_1 value (for example) that is consistently much larger than any other $x_j + w_{ij}$ so that for all $i = 1, \ldots, n$ we always have $y_i = w_{i1} + x_1$.

Of course, we have limited control over the degree that this effect will manifest during training, but we can at least avoid it at the start by choosing an appropriate initialization scheme. The idea is to make sure that at initialization there is a high probability that there is at least one $i = 1, \ldots, n$ so that $\frac{\partial y_i}{\partial x_j} = 1$ for each $j = 1, \ldots, m$. This is of course only possible when $n \geq m$, but in all our experiments we have used models for which this is the case (see Figure 8.1 and 8.2). Our aim will be to initialize the weights w_{ij} in (8.5) so that there is a high probability of each of the n inputs 'winning' roughly $\frac{n}{m}$ of the m outputs.

Assuming the inputs x_j generally stay in a range $\left[-\frac{K}{2}, \frac{K}{2}\right]$ for some $K > 0$. Then in the max-plus case we can initialize the weight matrix $W = [w_{ij}]_{ij} \in \mathbb{R}^{m \times n}$ as

$$w_{ij} = \text{Unif}[-\varepsilon, \varepsilon] + \begin{cases} 0 & \text{if } i = j \bmod m \\ -K & \text{else.} \end{cases} \tag{8.6}$$

The second term applies a penalty of $-K$ to each input unless $i = j \mod m$ ensuring that the m available 'wins' are fairly distributed among the n inputs (at least with high probability based

on our assumptions on the inputs). We additionally add a modest uniform distribution Unif$[-\varepsilon, \varepsilon]$ to keep the initialization scheme from being deterministic from run to run. The variable ε is chosen on the order of $\frac{K}{2}$.

A matrix initialized like this, but without the stochastic term, looks like

$$\begin{bmatrix} 0 & -K & \cdots & -K \\ -K & 0 & & -K \\ \vdots & & \ddots & \\ -K & & & 0 \\ 0 & -K & \cdots & -K \\ -K & 0 & & -K \\ \vdots & & \ddots & \end{bmatrix}. \tag{8.7}$$

We see that in each row all coefficients are equal to $-K$ except for on that is equal to 0. Assuming that the inputs are in the range $\left[-\frac{K}{2}, \frac{K}{2}\right]$, and forgetting the stochastic term for the moment, then the input corresponding to the column with the zero value will achieve the maximum for the output corresponding with that row. This has the effect of $(m \bmod n)$ number of inputs contributing to $\lfloor m/n \rfloor$ outputs and the remaining inputs contributing to $\lceil m/n \rceil$ outputs. This avoids the vanishing and exploding gradient problem that would be caused by a single input dominating.

For the min-plus case we similarly set

$$w_{ij} = \text{Unif}[-\varepsilon, \varepsilon] + \begin{cases} 0 & \text{if } i = j \bmod m \\ K & \text{else.} \end{cases} \tag{8.8}$$

In our experiments, normalization layers and weight decay keep input values fairly small, so our starting assumption of inputs being in a range $\left[-\frac{K}{2}, \frac{K}{2}\right]$ generally holds. We found $K = 1$ gave the intended effect of an initial fair distribution during the backward pass without overly biasing the initialization. We call this scheme *fair tropical initialization*.

LOGARITHMIC. For the logarithmic semiring (for some choice of $\mu \in \mathbb{R} \setminus \{0\}$) consider the operator $(R_{\log}^{\mu})^m \to (R_{\log}^{\mu})^n$ given by

$$y_i = \frac{1}{\mu} \log\left(\sum_{j=1}^{m} e^{\mu(w_{ij}+x_j)}\right) \tag{8.9}$$

for all $i = 1 \dots n$, where $x_1, \dots, x_m \in R_{\log}^{\mu}$ are the inputs, $y_1, \dots, y_n \in R_{\log}^{\mu}$ are the outputs and $[w_{ij}]_{ij} \in (R_{\log}^{\mu})^{n \times m}$ are the trainable parameters. Then the partial derivatives are given by

$$\frac{\partial y_i}{\partial x_j} = \frac{e^{\mu(x_j+w_{ij})}}{\sum_{k=1}^{m} e^{\mu(x_k+w_{ik})}},$$

which is nothing but the *softmax* function with temperature $\frac{1}{\mu}$ over the values $\{x_k + w_{ik}\}_{k=1}^{m}$.

Let us now take the same approach as GLOROT and BENGIO (2010) and look at the inputs x_j and outputs y_i as random variables and control forward variance. We assume all the inputs x_j are i.i.d. with expected value $\mathbb{E}[x_j] = 0$ and some finite variance $\mathrm{Var}(x) < \infty$. Then we can estimate the variance of the outputs using the *delta method* as

$$\mathrm{Var}(y_i) \approx \sum_{j=1}^{m} \frac{\partial y_i}{\partial x_j} (\mathbb{E}[x_1], \dots, \mathbb{E}[x_m])^2 \,\mathrm{Var}(x_j),$$

and since we assumed the input distributions to be i.i.d. and centered we have

$$\mathrm{Var}(y_i) \approx \mathrm{Var}(x) \sum_{j=1}^{m} \frac{\partial y_i}{\partial x_j}(0, \dots, 0)^2 = \mathrm{Var}(x) \sum_{j=1}^{m} \left(\frac{e^{\mu w_{ij}}}{\sum_{k=1}^{m} e^{\mu w_{ik}}}\right)^2.$$

Ideally we want $\mathrm{Var}(y_i) \approx \mathrm{Var}(x)$, which is the case if

$$\sum_{j=1}^{m} \left(\frac{e^{\mu w_{ij}}}{\sum_{k=1}^{m} e^{\mu w_{ik}}}\right)^2 \approx 1$$

or

$$\sum_{j=1}^{m} (e^{\mu w_{ij}})^2 \approx \left(\sum_{j=1}^{m} e^{\mu w_{ij}}\right)^2 \tag{8.10}$$

for all $i = 1, \dots, n$. We can satisfy this condition exactly by choosing a single $J_i = 1, \dots, m$ for each $i = 1, \dots, n$ and set that weight to $w_{i,J_i} = 0$ and set the other weights to $w_{ij} = -\operatorname{sgn}(\mu)\infty$. In that case $e^{\mu w_{ij}} = 1$ if $j = J_i$ and $e^{\mu w_{ij}} = 0$ if not and consequently both sides of (8.10) are equal to 1.

We can also make a similar analysis for the backward pass. Let $(\overline{y}_i)_{i=1}^n$ be the loss gradients of the outputs and $(\overline{x}_j)_{j=1}^m$ be the loss gradients of the inputs. We interpret these as random variables, where we assume the output gradients $\overline{y}_i$ are i.i.d. with $\mathbb{E}[\overline{y}_i] = 0$ and finite variance $\operatorname{Var}(\overline{y}_i) = \operatorname{Var}(\overline{y}) < \infty$ for all $i = 1, \dots, n$. Then the loss gradients of the inputs are computed as

$$\overline{x}_j = \sum_{i=1}^n \frac{\partial y_i}{\partial x_j}(x_1, \dots, x_m)\overline{y}_i,$$

or if we are talking about the expected backward pass over the possible inputs we can say

$$\overline{x}_j \approx \sum_{i=1}^n \frac{\partial y_i}{\partial x_j}(0, \dots, 0)\overline{y}_i.$$

The variance of the input gradients can then be approximated as

$$\operatorname{Var}(\overline{x}_j) \approx \operatorname{Var}\left(\sum_{i=1}^n \frac{\partial y_i}{\partial x_j}(0, \dots, 0)\overline{y}_i\right) = \operatorname{Var}(\overline{y}) \sum_{i=1}^n \left(\frac{e^{\mu w_{ij}}}{\sum_{k=1}^n e^{\mu w_{ik}}}\right)^2,$$

which implies that to get $\operatorname{Var}(\overline{x}_j) \approx \operatorname{Var}(\overline{y})$ we need

$$\sum_{i=1}^n \left(\frac{e^{\mu w_{ij}}}{\sum_{k=1}^n e^{\mu w_{ik}}}\right)^2 \approx 1.$$

or

$$\sum_{i=1}^n (e^{\mu w_{ij}})^2 \approx \left(\sum_{i=1}^n e^{\mu w_{ij}}\right)^2 \tag{8.11}$$

for all $j = 1, \dots, m$. We can satisfy this condition exactly by choosing a single $I_j = 1, \dots, n$ for each $j = 1, \dots, m$ and set that weight to $w_{I_j,j} = 0$ and set the other weights to $w_{ij} = -\operatorname{sgn}(\mu)\infty$. In that case $e^{\mu w_{ij}} = 1$ if $i = I_j$ and $e^{\mu w_{ij}} = 0$ if not and consequently both sides of (8.11) are equal to 1.

Satisfying both forward (8.10) and backward condition (8.11) is only possible in the case that $m = n$ where we can set $I_j = j$ and $J_i = i$. But even then, this scheme is not satisfactory. If we initialize parameters to $\pm\infty$ (depending on the sign of μ) they can not be changed by the addition of finite numbers thus making training impossible. We might then be tempted to substitute $\pm\infty$ by a very large negative of positive value to initialize with. But this still causes a problem for training.

Consider the loss gradient for an individual parameter:

$$\overline{w}_{ij} = \frac{e^{\mu(x_j+w_{ij})}}{\sum_{k=1}^{m} e^{\mu(x_k+w_{ik})}}\overline{y}_i. \tag{8.12}$$

If we try to satisfy (8.10) and (8.11) then we would set $w_{ik} = 0$ for some $k = 1, ..., n$ for every $i = 1, \dots, n$. This implies that the denominator in (8.12) is greater than or equal to 1. If we subsequently set $w_{ij} = -\operatorname{sgn}(\mu)K$ for $j \neq k$ and K a very large number then the numerator of (8.12) becomes vanishingly small, possible zero in floating point format. This essentially freezes the parameter's value since any updates applied to it would become practically zero.

Assuming the inputs $(x_j)_{j=1}^m$ are centered, the expected value of the fraction in (8.12) is given by

$$\frac{e^{\mu w_{ij}}}{\sum_{k=1}^{m} e^{\mu w_{ik}}}, \tag{8.13}$$

which gives values in the range $[0, 1]$ for any $i = 1, \dots, n$ and $j = 1, \dots, m$. We can avoid this fraction becoming (effectively) zero from the start by initializing all the parameters to roughly the same value, i.e. for all $i = 1, \dots, n$ set

$$w_{i1} \approx w_{i2} \approx \ldots \approx w_{im}. \tag{8.14}$$

In this case we achieve the *maximum of the minimum* where (8.13) gives the value $1/m$ for all $i = 1, \dots, n$ and $j = 1, \dots, m$.

Satisfying the forward condition (8.10), backward condition (8.11) and parameter trainability condition (8.14) all at the same time is not possible. However, the fair tropical initialization scheme does present a reasonable trade-off between these three clashing requirements. Indeed if we initialize using this scheme (assuming

$n > m$ for the moment) we get a matrix like (8.7), where we let $K > 0$ if $\mu > 0$ and $K < 0$ if $\mu < 0$. First, most entries have the same $-K$ value and so satisfy (8.14) to some degree. Second, in every row we have a single element (the zero values) that are relatively dominant with respect to the other elements in the row, thus catering to (8.10). Third, in every column we have at most $\lceil n/m \rceil$ zero values that are relatively dominant with respect to the other elements in the column, thus catering to (8.11).

8.6 FULLY CONNECTED EXPERIMENTS

To check the viability of our proposed scheme we first run a set of experiments with small fully connected networks on a variety of datasets. The datasets for this series of experiments are:

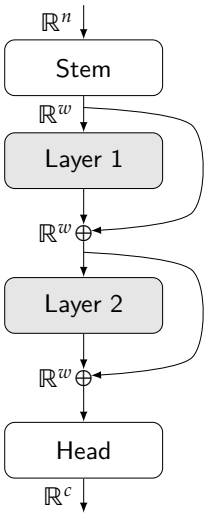
- the classic iris dataset from ANDERSON (1936), available at www.kaggle.com/datasets/uciml/iris,
- the heart disease dataset available at www.kaggle.com/datasets/johnsmith88/heart-disease-dataset,[1]
- the circles and spheres dataset from NAITZAT, ZHITNIKOV, and LIM (2020), available at www.github.com/topnn/topnn_framework,
- the FashionMNIST dataset from XIAO, RASUL, and VOLLGRAF (2017), available at www.github.com/zalandoresearch/fashion-mnist.

The code of the experiments is available at www.github.com/bmnsmets/semitorch.

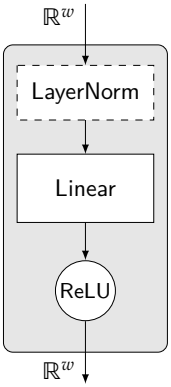
ARCHITECTURES. We will train a set of network architectures that only vary slightly based on the dataset. Every model will be based on a common architecture with a linear stem and head with two layers with residual connections in between, see Figure 8.1a. The stem and head are there to convert the features in the dataset

1 The provenance of this dataset is complicated and its contents are of questionable value for predicting heart disease, see SIMMONS II (2021) for an investigation into this dataset. For our purpose of comparing network architectures this dataset is still perfectly serviceable to see how well models deal with poor data.

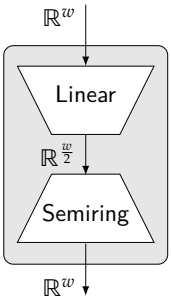
($n \in \mathbb{N}$ features) to internal network features ($w \in \mathbb{N}$ features) and back to the output classes ($c \in \mathbb{N}$ classes) from the dataset. The internal layers we will vary between a traditional ReLU based layer (Figure 8.1b) and semiring based layers (Figure 8.1c and 8.1d). For the iris and heart disease datasets we use the ReLU layer without layer normalization and semiring layer from Figure 8.1c. For the circles, spheres and FashionMNIST dataset we use the ReLU layer with layer normalization and the semiring layer from Figure 8.1d.



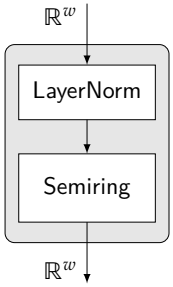
(A) Common model layout shared by all our fully connected networks.



(B) Baseline Linear-ReLU layer, normalization is only added for the Circles, Spheres and FashionMNIST experiments.



(C) Alternating between linear and second semiring operators, as used in the Iris and Heartdisease experiments.



(D) Combining layer normalization with a quasilinear operator as used in the Circles, Spheres and FashionMNIST experiments.

FIGURE 8.1: Network architecture for our fully connected experiments. The Head and Stem modules are linear modules. None of the modules include biases. The layer normalization modules include affine transforms. The number of input features n and number of output classes c are dataset dependent, the internal width parameter w is chosen per experiment. Each network under consideration has the exact same number of parameters per experiment.

TRAINING. Training the networks with semiring based activation proved challenging. The usual setups that are known to work well for conventional neural network do not necessarily carry over. Indeed, parameter initialization, normalization layers, optimizers, and schedulers have all seen much research to obtain the best possible results with conventional neural networks. We spent a lot of time finding a training setup that worked for our modified networks but we expect that this is an area where more gains could be found. We document the training setup that we used for our fully-connected experiments next.

OPTIMIZER & LEARNING RATE SCHEDULER. We ended up settling on a combination of the *AdamW* optimizer (LOSHCHILOV and HUTTER, 2019) and the *1-cycle* learning rate scheduler (SMITH and TOPIN, 2019). Key in getting comparable performance out of the semiring based networks as compared to the baseline network was assigning a separate optimizer to the linear parameters and the parameters of the semiring module. In general we needed to assign smaller learning rates to the semiring parameters to have stable training. The FashionMNIST experiment is an exception to this, there we obtained best results by increasing the semiring learning rate and decreasing the linear learning rate. The hyperparameters for all the experiments are listed in Table 8.1.

RESULTS. We perform 10 training runs for each type of network on each dataset using the training setup we described. We measure performance by the accuracy (mean ± standard deviation over the runs) on the testing dataset that was not seen during training. We use the same training/testing split of the dataset for every type of network. The results are summarized in Table 8.2.

The semiring activation networks manage to modestly outperform the classic network in 3 of the 5 cases and are only slightly behind in the other 2 cases. A standout failure is the logarithmic semiring networks with $\mu \in \{-1, 1\}$ in the *circles* and *spheres* datasets. We can explain this based on the fact that these are low-dimensional problem (2 respectively 3 input dimensions) where there is a sharp boundary between the classes. At the same time the logarithmic maps are fairly gradual for small absolute values of μ and weight decay keeps the parameters from becoming

Hyperparameter	Iris	Heartdisease	Circles	Spheres	FashionMNIST
Epochs	40	40	100	100	40
Batchsize	8	16	32	16	512
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Scheduler	1-cycle lin	1-cycle lin	1-cycle lin	1-cycle lin	1-cycle lin
Learning rate (linear)	0.020	0.010	0.020	0.020	0.008
Learning rate (tropical)	0.004	0.008	0.010	0.010	0.040
Learning rate (logarithmic)	0.040	0.008	0.008	0.008	0.040
Weigh decay	0.01	0.05	0.01	0.01	0.01
Warmup epochs	18	18	45	45	18
Warmup factor	$1/10$	$1/10$	$1/10$	$1/10$	$1/10$
Annihilation factor	$1/1000$	$1/1000$	$1/1000$	$1/1000$	$1/1000$
Internal width w	4	48	16	32	8
RNG seed	42	42	42	42	42
Parameters	60	5328	2336	2336	2288

TABLE 8.1: Hyperparameters for the fully connected semiring experiments.

Model / dataset	Iris	Heartdisease	Circles	Spheres	FashionMNIST
ReLU	97.14 ±0.62	**83.93** ±2.16	84.50 ±0.39	80.91 ±1.62	**83.82** ±0.35
maxplus	97.52 ±1.01	83.50 ±2.27	84.84 ±0.86	**81.69** ±0.42	83.50 ±0.34
minplus	97.62 ±0.49	82.84 ±1.22	84.91 ±0.35	81.61 ±0.60	83.39 ±0.24
logplus $\mu = -10$	97.58 ±1.00	81.72 ±1.62	85.06 ±0.25	81.52 ±0.71	83.46 ±0.37
logplus $\mu = -1$	97.90 ±0.39	83.26 ±2.15	73.92 ±6.84	69.41 ±5.05	83.50 ±0.28
logplus $\mu = 1$	**97.97** ±0.49	82.38 ±1.73	75.06 ±7.89	67.28 ±5.53	83.46 ±0.16
logplus $\mu = 10$	97.46 ±0.62	81.86 ±1.15	**85.16** ±0.26	81.62 ±0.45	83.56 ±0.37
Parameters	60	5328	2336	2336	2288

TABLE 8.2: Accuracy percentage (mean ± standard deviation) of the trained fully connected networks on the testing sets of the various classification datasets. The best result for each dataset is indicated in **purple**, the second best result in green.

large enough to compensate for that. Consequently, the logarithmic networks have a hard time separating classes that are close together.

On the whole though we can conclude that the semiring approach is at least as viable as the standard approach for these smaller problems. We can now try to see if we can scale up to some larger networks.

8.7 CONVNEXT EXPERIMENTS

We perform a final experiment to see how viable replacing activation functions with semiring activation is in an existing modern and optimized network architecture. We start from the *ConvNeXt* network from LIU et al. (2022), specifically the *atto* variant available from WIGHTMAN (2019), and train it on FashionMNIST.

ARCHITECTURE MODIFICATIONS. The core *block* at the heart of any ConvNeXt consists of two phases:

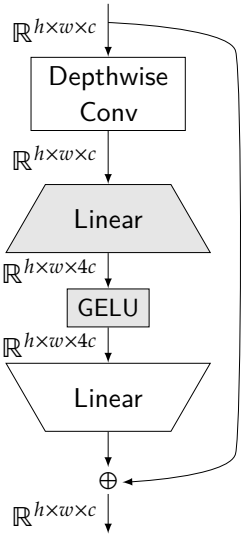
1. a *depthwise convolution* that applies a kernel per feature map and
2. a *normalization* followed by a *reverse bottleneck MLP* that is applied *pixel-wise*.

The result of these two operations is added to the input to form a residual connection, see Figure 8.2a.

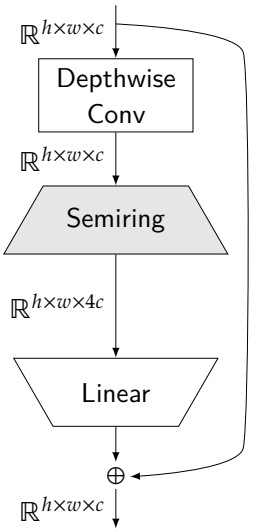
Our adaptation keeps the (linear) depthwise convolution and normalization but replaces the MLP with a concatenation of a non-linear semiring operator and a linear operator. The semiring operation performs a 4× fan-out while the linear operation reduces again to the original amount of channels, thus retaining the same reverse bottleneck of the original, see Figure 8.2b.

RESULTS. We compare the performance of the semiring based networks against the baseline network using an MLP. We perform 10 training runs per model and record accuracy on the test dataset. We train 4 semiring based models: maxplus, minplus, logplus($\mu = -1$) and logplus($\mu = 1$). We train all models for the same 50 epochs and batchsize of 512 and adapt the rest of the training setup to each model. Details for the training setup can be found in Table 8.3. The results are summarized in Table 8.4.

Training with these (relatively) large scale networks proved more challenging than with the previous toy networks. In partic-



(A) Standard ConvNeXt block with two linear maps and an activation function in between.



(B) Modified ConvNeXt block with a semiring and linear layer without activation function.

FIGURE 8.2: Standard and semiring-based ConvNeXt (LIU et al., 2022) blocks compared. Normalization and dropout modules are omitted.

ular the logarithmic networks proved challenging and required us to remove the affine transform from the normalization layer to avoid numerical stability issues. The problem is that in single precision floating point, the function $x \mapsto e^x$ already overflows at $x = 89$. This would not happen if x is normalized but is likely enough to happen if we let x be an affine transform of a normalized input. The overflow would only need to happen at one place in the network, after which 'inf' values would propagate throughout the network and ruin training. As we already experienced in the previous experiments, both the tropical and logarithmic variants proved to be more sensitive to the training hyperparameters than the baseline network. This sensitivity makes finding a good training setup harder for the semiring networks than for the baseline network.

As the second column of Table 8.4 shows, both the tropical and logarithmic network's accuracy fall significantly short of the

Hyperparameter	MLP	tropical-linear	logplus-linear
Epochs	50	50	50
Batchsize	512	512	512
Affine LayerNorm	Yes	Yes	No
Optimizer	AdamW	AdamW	AdamW
Scheduler	1-cycle cos	1-cycle cos	1-cycle cos
Learning rate (linear)	$4 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	$6 \cdot 10^{-3}$
Learning rate (semiring)	-	$1 \cdot 10^{-3}$	$5 \cdot 10^{-4}$
Weight decay (linear)	$5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
Weight decay (semiring)	-	$1 \cdot 10^{-2}$	$1 \cdot 10^{-2}$
Warmup epochs	5	5	5
Warmup factor (linear)	$1/25$	$1/50$	$1/50$
Warmup factor (semiring)	-	$1/50$	$1/50$
Annihilation factor	$1/500$	$1/250$	$1/250$
RNG seed	42	42	42
Initialization (linear)	Kaiming	Kaiming	Kaiming
Initialization (semiring)	-	Tropical fair	Tropical fair
Parameters	3,375,850	3,375,850	3,372,170

TABLE 8.3: Hyperparameters settings for the ConvNeXt experiments.

Feed-forward type	Test accuracy(%)	Train accuracy(%)	Gap(%)
MLP (linear-GELU-linear)	91.24 ±0.13	99.97 ±0.07	−8.63
maxplus-linear	89.08 ±0.29	93.26 ±2.32	−4.18
minplus-linear	89.93 ±0.24	94.74 ±2.79	−4.81
logplus-linear $\mu = -1$	88.15 ±0.15	91.72 ±2.52	−3.57
logplus-linear $\mu = 1$	88.31 ±0.23	91.60 ±2.52	−3.29

TABLE 8.4: Accuracy(%) (mean ± standard deviation) of the trained ConvNeXt models on the FashionMNIST test dataset and on the last 100 training batches. The generalization gap is the difference between the two mean accuracies.

baseline network. This shortfall is not favorable to the semiring idea but there are some nuances to be made.

Looking at the third column of Table 8.4 we see that the baseline network has saturated its performance on the training data. During the last 100 batches the baseline network has virtually 100%

training accuracy. Consequently it can not benefit much more from further training. At the same time there is a fairly large gap between the training and testing accuracy as can be seen in the last column of Table 8.4. On the side of the semiring networks we see that after 50 epochs there is still significant room for improvement on the training data and that the gap between the testing and training accuracy is much more modest.

We conjecture that this difference is partly explained by the standard training regime being very well suited for the baseline network but that the modifications we made—chiefly the parameter initialization scheme—are not sufficient to extract maximum performance from the semiring networks. Indeed, the methods for neural network training have evolved much over the last decade with much research into optimizers, schedulers, initialization, regularization, etc., all focused on the linear with activation type networks. It would not be unreasonable to assume similar efforts could yield similar progress in training semiring based networks.

8.8 DISCUSSION & CONCLUDING REMARKS

VIABILITY. In this chapter we have proposed a general framework for constructing trainable nonlinearities for neural networks. We constructed several such trainable nonlinearities based on the tropical and logarithmic semirings and introduced an associated parameter initialization scheme. We did a series of experiments to show the viability of replacing the traditional activation function in neural networks with nonlinearities based on the aforementioned semirings.

UNREALIZED POTENTIAL. Our experiments showed that the semiring approach is viable in that we can get very good results for small networks and decent but not state-of-art results for larger networks. With regard to the larger scale experiment we concluded that there is more performance on the table for semiring networks that can potentially be unlocked by designing more suitable optimizers, schedulers, initialization schemes, regularization schemes, etc. Whether such a line of research would be worthwhile is debatable for two reasons.

First, while our experiments show viability, they do not immediately show a clear advantage over traditional neural networks with activation functions.

Second, the current development of speciality deep learning hardware (Dhilleswararao, Boppu, Manikandan, and Cenkeramaddi, 2022) focuses to a large degree on optimizing linear operations. Most GPUs in general use today already contain hardware dedicated to linear matrix multiplication (Markidis, Der Chien, Laure, Peng, and Vetter, 2018), this makes linear computations much more efficient than doing semiring computations that would have to be executed by general purpose computing units at a higher cost in time and energy.

CONCLUSION ON PDE-G-CNNS. Recall that our interest in the semiring structure was instigated by the PDE-G-CNNs of previous chapters. In those networks we also had trainable nonlinear operators based on the tropical semiring, but these operators were further structured to satisfy equivariant and PDE properties. We asked whether the benefits from PDE-G-CNNs were perhaps not largely a consequence of the tropical semiring structure rather than the equivariant and PDE structures.

We can now answer this question in the negative. Without the equivariance and PDE constraints the tropical operator becomes very hard to deal with and a challenge to train. PDE-G-CNNs can be successfully trained without much adaptation of the training regime and are not unusually sensitive to hyperparameters. PDE-G-CNNs also do not require more epochs to saturate their training data than normal CNNs, which was clearly an issue for the tropical semiring networks in our ConvNeXt experiment.

Overall, we conclude that the equivariant and PDE structures play an important role in making PDE-G-CNNs work and the semiring structure is not—by itself—the cause of the benefits of PDE-G-CNNs.

Part III

IMAGE DENOISING

Any real imaging application has to deal with inputs that are noisy to some degree. Often then, denoising an image is the first stage of processing or analyzing an image. In this part we detail our research in developing new denoising methods that are good at preserving edges in an image while also being roto-translation equivariant.

9

TOTAL VARIATION & MEAN CURVATURE DENOISING

Two key ideas have greatly improved techniques for image enhancement and denoising: the lifting of image data to multi-orientation distributions, and the application of nonlinear PDEs such as *total variation flow* (TVF) and *mean curvature flow* (MCF). These two ideas were recently combined by Chambolle and Pock (2019) (for TVF) and Citti, Franceschiello, Sanguinetti, and Sarti (2016) (for MCF) for two-dimensional images.

In this chapter, we extend their approach to enhance and denoise images of arbitrary dimension, creating a unified geometric and algorithmic PDE framework, relying on (sub-)Riemannian geometry. In particular, we follow a different numerical approach, for which we prove convergence in the case of TVF by an application of *Brezis-Komura gradient flow* theory. Our framework also allows for additional data-adaptation through the use of locally adaptive frames and coherence enhancement techniques.

We apply TVF and MCF to the enhancement and denoising of elongated structures in 2D images via orientation scores, and compare the results to *Perona-Malik diffusion* (Perona and Malik, 1990) and BM3D (Dabov, Foi, Katkovnik, and Egiazarian, 2007). We also demonstrate our techniques in 3D in the denoising and enhancement of crossing fiber bundles in DW-MRI. In comparison to data-driven diffusions, we see a better preservation of bundle boundaries and angular sharpness in fiber orientation densities at crossings.

outline. This chapter is based on the contents of Duits, St-Onge, Portegies, and Smets (2019) and Smets, Portegies, St-Onge, and

Duits (2021). We will first give a broad introduction to related work. Much of the theoretical preliminaries we have already covered in Part I, so we can limit our theoretical coverage to *orientation scores* and *total variation* and *mean curvature flows*.

9.1 INTRODUCTION

LIFTING. In the last decade, many PDE-based image-analysis techniques for tracking and enhancement of curvilinear structures in images took advantage of lifting image data, typically defined on $\mathbb{R}^d$, to a multi-orientation distribution (e.g. an orientation score) defined on the homogeneous space $\mathbb{M}_d$ of d-dimensional positions and orientations. See Figure 9.1 and Bekkers (2017), Boscain, Chertovskih, Gauthier, Prandi, and Remizov (2018), Chambolle and Pock (2019), Citti and Sarti (2006), Duits (2005), and Vogt and Lellmann (2018). After lifting the image to a multi-orientation distribution, the distribution is taken as an initial condition of a PDE flow. After solving a limited number of iterations of the PDE model, one obtains a regularized version of the original distribution, and by integration over all orientations, one obtains a regularized version of the original image.

The key advantage of lifting the images from $\mathbb{R}^d$ to the homogeneous space $\mathbb{M}_d$ is that the PDE flow can act differently on substructures with different orientations (Bekkers, 2017; Chambolle and Pock, 2019; Duits, Meesters, Mirebeau, and Portegies, 2018). For instance, if the image contains two crossing lines, the PDE can regularize the two lines independently, rather than regularizing the whole crossing. Similarly, if the image contains a corner, the corner is preserved in the regularized image.

APPLICATIONS USING LIFTING. This idea of lifting images has been successfully implemented for image enhancement (Citti, Franceschiello, Sanguinetti, and Sarti, 2016; Franken and Duits, 2009), geodesic tracking (Bekkers, Duits, Mashtakov, and Sanguinetti, 2015; Chambolle and Pock, 2019; Schmidt and Weickert, 2016), image denoising (Ghimpeţeanu, Batard, Bertalmío, and Levine, 2015), contrast perception and optical illusions (Bertalmío, Calatroni, Franceschi, Franceschiello, and Prandi, 2019). For

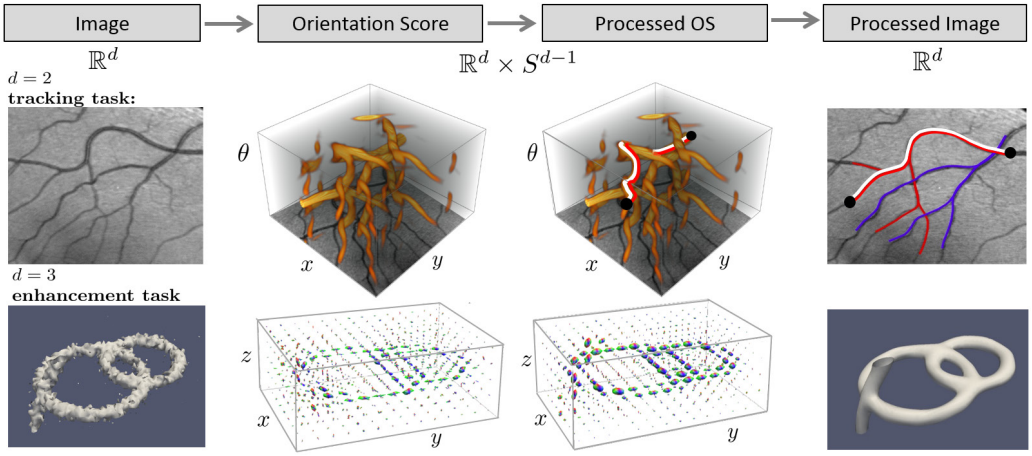


FIGURE 9.1: An illustration of the lifted image workflow. Instead of direct PDE-based processing of an image, we apply PDE-based processing on a lifted image on $\mathbb{R}^d \rtimes S^{d-1}$, which is obtained by convolving the image with a set of rotated wavelets allowing for stable reconstruction (BEKKERS, 2017; DUITS, 2005; JANSSEN, JANSSEN, BEKKERS, BESCÓS, and DUITS, 2018). *2nd row:* Vessel-tracking in a 2D image via geodesic PDE-flows on the lifted image (BEKKERS, 2017; CHAMBOLLE and POCK, 2019; DUITS, MEESTERS, MIREBEAU, and PORTEGIES, 2018). *3rd row:* CED-OS diffusion of a 3D image DUITS, JANSSEN, HANNINK, and SANGUINETTI, 2016; JANSSEN, JANSSEN, BEKKERS, BESCÓS, and DUITS, 2018 visualized as a field of angular profiles.

instance, Perona-Malik diffusion has been lifted to the homogeneous space $\mathbb{M}_d$ (CREUSEN, DUITS, VILANOVA, and FLORACK, 2013) and *coherence enhancing diffusion* (CED) (WEICKERT, 1999) has been lifted to the method of *coherence enhancing diffusion on invertible orientation scores* (CED-OS) (FRANKEN and DUITS, 2009) and to its 3D generalization (JANSSEN, JANSSEN, BEKKERS, BESCÓS, and DUITS, 2018).

PDE-flows on orientation lifts of *3D images* are relevant for applications such as fiber enhancement (CREUSEN, DUITS, VILANOVA, and FLORACK, 2013; DUITS, DELA HAIJE, CREUSEN, and GHOSH, 2013; REISERT and BURKHARDT, 2008; VOGT and LELLMANN, 2018) and fiber tracking (PORTEGIES, 2018) in *diffusion-weighted magnetic reso-*

nance imaging (DW-MRI) , and in enhancement (JANSSEN, JANSSEN, BEKKERS, BESCÓS, and DUITS, 2018) and tracking (COHEN, DEFFIEUX, DEMENÉ, COHEN, and TANTER, 2018) of blood vessels in 3D images.

WORKFLOW. The general workflow is illustrated in Figure 9.1. The original image is described by a function $f : \Omega_f \to \mathbb{R}^+$, where $\Omega_f \subset \mathbb{R}^d$ is its support. From $f \in L^2(\Omega_f)$, one computes an orientation lift $U : \mathbb{M}_d \to \mathbb{C}$, compactly supported within

$$\Omega = \Omega_f \times S^{d-1} \subset \mathbb{M}_d. \tag{9.1}$$

There are various ways to construct such a lift: it can be (the real part of) an invertible orientation score (DUITS and FRANKEN, 2010) (cf. Figure 9.1), a channel-representation (FELSBERG, FORSSÉN, and SCHARR, 2005), a lift by Gabor wavelets (BASPINAR, CITTI, and SARTI, 2018), or a fiber orientation density (PORTEGIES, FICK, et al., 2015). In all of these approaches the absolute value $|U(\boldsymbol{x}, \boldsymbol{n})|$ can be regarded (after normalization) as a probability density of finding a fiber structure at position $\boldsymbol{x} \in \mathbb{R}^d$ with local orientation $\mathbf{n} \in S^{d-1}$. We set the orientation lift U as an initial condition of a PDE flow $U \mapsto \Phi_t(U)$ with evolution time $t > 0$. Finally, the processed multi-orientation representation $\Phi_t(U)$ is integrated over all orientations to obtain the enhanced image f_t. In this chapter, we will work with the orientation score, with the main motivation being that this operation is invertible (DUITS, 2005), so that when taking $t \downarrow 0$, the output equals the input, i.e. $\lim_{t \downarrow 0} f_t = f$ in the L^2-sense.

TYPES OF FLOWS. The enhanced image that one obtains after running a PDE flow, (the bottom-right picture in Figure 9.1), naturally depends on the type of flow used. One flow may be more suitable than another, depending on the requirements imposed on the resulting image. In case it is important to preserve sharp transitions in the image, while maintaining plateaus, nonlinear flows such as total variation flows (TVF) and mean curvature flow (MCF) (SAPIRO, 2006) are typically more suited than nonlinear diffusion flows such as Perona & Malik diffusion (PERONA and MALIK, 1990) and coherence enhancing diffusion (WEICKERT, 1999).

For $d = 2$, TVF and MCF were recently generalized to lifted images by CHAMBOLLE and POCK (2019) and CITTI, FRANCESCHIELLO, SANGUINETTI, and SARTI (2016) respectively.

Their promising results have motivated us to generalize TVF and MCF to lifted images for general dimension d and provide a general geometric and algorithmic framework that can accommodate features such as locally adaptive frames and coherence enhancement.

OUR CONTRIBUTION. The benefits of our approach are that we obtain a single unifying geometric and algorithmic framework for arbitrary d, with efficient algorithms (for $d \in \{2, 3\}$) that preserve crossing lines, corners, plateaus, edges and bundle boundaries and can improve curvature adaptation via the optional inclusion of locally adaptive frames. Such frames account for curvature of lines and allow us to remove bias towards sampled orientations in orientation scores.

Our PDE-methods on $\mathbb{M}_d$ are computationally more expensive than their counterparts acting only on $\mathbb{R}^d$, but they are still practical. Similar to crossing-preserving nonlinear diffusion on $SE(2) \equiv \mathbb{M}_2$, locally adaptive frames allow us to remove orientation sampling bias in orientation scores (FRANKEN, 2008, Fig. 6.11), and to use only 4 orientation samples (FRANKEN and DUITS, 2009). For our crossing-preserving MCF and TVF on $\mathbb{M}_2$ we sample our (processed) orientation scores only on 8 orientations. On $\mathbb{M}_3$ we compute regularized orientation lifts on a grid with 16 orientations, where we rely on efficient numerical schemes for PDEs on $\mathbb{M}_3$ relying on the low-order PDE discretization schemes explained in CREUSEN, DUITS, VILANOVA, and FLORACK (2013) and MARTIN, BEKKERS, and DUITS (2017), instead of higher order schemes via spherical harmonics (JANSSEN, DELA HAIJE, MARTIN, BEKKERS, and DUITS, 2017, Ch. 3.4), in order to reduce computation time.

9.2 ORIENTATION SCORES

In order to disentangle all local orientations in an image we lift the data from position space $\mathbb{R}^d$ to the homogeneous space $\mathbb{M}_d$ of positions and orientations. This means that we extend the domain of an image. See Fig. 9.1, where we lift the data from $\mathbb{R}^d$ towards $\mathbb{M}_d$ via invertible orientation scores.

Refer to §2 for homogeneous spaces in general and §4.4 for an in-depth treatment of $\mathbb{M}_2$.

Building an orientation score starts with selecting an orientation-sensitive filter (or wavelet) $\psi \in L^1 \cap L^2\left(\mathbb{R}^d\right)$. We can then—under appropriate conditions (Duits, 2005; Janssen, Janssen, Bekkers, Bescós, and Duits, 2018)—filter out a particular direction from an image $f \in L^2\left(\mathbb{R}^d\right)$ by convolving the image with this filter aligned to that direction. An orientation score $\mathcal{W}_\psi f$ can then be constructed by applying this filtering for all directions $n \in S^{d-1}$:

$$\mathcal{W}_\psi f(x, n) = \int_{\mathbb{R}^d} \overline{\psi(R_n^{-1}(y - x))}\, f(y)\, \mathrm{d}y, \tag{9.2}$$

for all $x \in \mathbb{R}^d$ and rotations $R_n \in \mathrm{SO}(d)$ that map a reference axis $a \in S^{d-1}$ to n, i.e. $R_n a = n$.

In this work we will be using *cake wavelets* (Duits, 2005; Janssen, Janssen, Bekkers, Bescós, and Duits, 2018) for our filter ψ, illustrated in Figure 9.2 for $d = 2$. These wavelets are directional filters that have the property that we can accurately reconstruct the original image from the orientation score—again under appropriate conditions—by integration over S^{d-1}, i.e.

$$f(x) \approx \int_{S^{d-1}} \mathcal{W}_\psi f\,(x, n)\, \mathrm{d}\sigma(n), \tag{9.3}$$

where σ denotes the usual surface measure over S^{d-1}. We always use standard cake wavelet parameter settings from Martin, Bekkers, and Duits (2017) in our experiments.

The explicit formulas for these cake wavelets that allow invertible orientation scores are available in Bekkers (2017) and Duits (2005) and specifically for $d = 3$ in Janssen, Janssen, Bekkers, Bescós, and Duits (2018). An intuitive illustration of an orientation score is given in Figure 9.2.

SE(d) and the homogeneous space $\mathbb{M}_d$. We talked extensively about SE(2) and $\mathbb{M}_2$ in §4. As we will now also be considering the 3D case using SE(3) and $\mathbb{M}_3$ we will briefly introduce the general case of SE(d) and $\mathbb{M}_d$.

Consider the *special Euclidean group* in d dimensions $\mathrm{SE}(d) := \mathbb{R}^d \rtimes SO(d)$, the semi-direct product of the translation group $\mathbb{R}^d$ and the rotation group $SO(d)$ of orthogonal $d \times d$ matrices. We

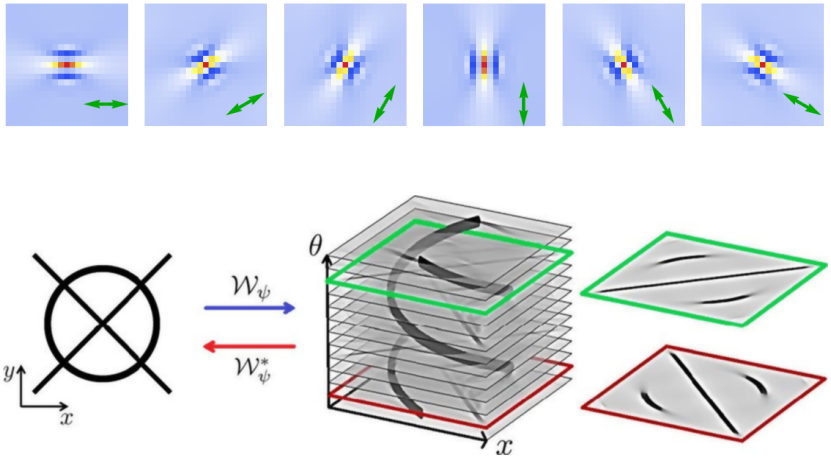


FIGURE 9.2: *Top:* cake wavelets (DUITS, 2005) for $d = 2$ used to directionally filter an image and construct an orientation score. *Bottom:* how orientation scores disentangle orientations

call elements of $\mathrm{SE}(d)$ *roto-translations*. The product of two roto-translations $g_i = (x_i, R_i)$ in $\mathrm{SE}(d)$ is given by

$$g_1 g_2 = (x_1, R_1)\,(x_2, R_2) := (x_1 + R_1 x_2,\ R_1 R_2)\,, \tag{9.4}$$

just like it is in 2 dimensions in (4.1) These roto-translations act transitively on the space $\mathbb{M}_d := \mathbb{R}^d \times S^{d-1}$ by

$$(x, R) \cdot (y, n) := (x + Ry, Rn), \tag{9.5}$$

for all $(y, n) \in \mathbb{M}_d := \mathbb{R}^d \times S^{d-1}$ and all roto-translations $(x, R) \in \mathrm{SE}(d)$.

We choose an *a priori reference vector* $a \in S^{d-1}$, say $a = (1, 0)^T$ if $d = 2$ or $a = (0, 0, 1)^T$ if $d = 3$. Then the stabilizer of the element $(0, a)$ is given by

$$\mathrm{SE}(d)_{(0,a)} = \left\{ g \in \mathrm{SE}(d) \;\middle|\; g \cdot (0, a) = (0, a) \right\}, \tag{9.6}$$

which is isomorphic to $\mathrm{SO}(d-1)$.

As we saw in §2.2, the homogeneous space is equivalent to the partition of left cosets of a stabilizer of one if its elements. Consequently, we have

$$\mathbb{M}_d := \mathbb{R}^d \times S^{d-1} \equiv \mathrm{SE}(d) / \big(\mathrm{SE}(d)_{(0,a)}\big)\,.$$

For $d = 2$, the subgroup $\mathrm{SE}(2)_{(0,a)} = \{(0, I)\}$ consists only of the unit element, and thereby the manifold $\mathbb{M}_2$ is diffeomorphic to $SE(2)$. However, for $d > 2$ the manifolds $\mathbb{M}_d$ and $\mathrm{SE}(d)$ are not diffeomorphic since $\mathrm{SE}(d)_{(0,a)}$ is not trivial.

For $d = 3$, the stabilizer can be described by

$$\mathrm{SE}(3)_{(0,a)} = \left\{ (0, R_{a,\alpha}) \;\middle|\; \alpha \in [0, 2\pi) \right\},$$

where $R_{a,\alpha} \in \mathrm{SO}(3)$ denotes a (counter-clockwise) rotation over angle α around the reference axis a. This means that two roto-translations $g_1 = (x_1, R_1)$ and $g_2 = (x_2, R_2)$ are equivalent if and only if

$$x_1 = x_2 \;\text{ and }\; \exists \alpha \in [0, 2\pi) : R_1 = R_2 R_{a,\alpha}.$$

The equivalence classes $[g] = \left\{ g' \in \mathrm{SE}(3) \;\middle|\; g' \sim g \right\}$ are usually just denoted by $p = (x, n)$ as they consist of all rigid body motions $g = (x, R_n)$ that map the reference point $(0, a)$ onto $(x, n) \in \mathbb{R}^3 \times S^2$:

$$g \cdot (0, a) = (x, n).$$

DOMAIN OF AN ORIENTATION SCORE. The orientation score is well-defined on the domain $\mathbb{M}_d$ if we assume ψ is not affected under the action of subgroup $\mathrm{SE}(d)_{(0,a)}$. For $d = 3$ this means we must impose axial symmetry on the wavelets, for details see JANSSEN, JANSSEN, BEKKERS, BESCÓS, and DUITS (2018).

LIE ALGEBRA. As a manifold, we can view the group $\mathrm{SE}(d)$ in a standard way as a submanifold of $\mathbb{R}^d \times \mathbb{R}^{d \times d}$. The Lie algebra $\mathrm{se}(d)$ of $\mathrm{SE}(d)$ has dimension $D = \frac{1}{2} d(d+1)$. We choose a basis $(A_i)_{i=1}^D$ for $\mathrm{se}(d)$ with the following properties. The basis is orthonormal with respect to the inner product belonging to the standard Euclidean metric on $\mathbb{R}^d \times \mathbb{R}^{d \times d}$, the vectors $\{A_1, \ldots, A_d\}$ span the spatial part of the Lie algebra, which is isomorphic to $\mathbb{R}^d$ with the vector A_d corresponding to the derivative in the direction of $\boldsymbol{a}$. Recall that for $d = 2$ the subgroup $\mathrm{SE}(d)_{(0,a)}$ is trivial. For $d \geq 3$ one has that the set $\{A_{2d}, \ldots, A_D\}$ forms a basis for the stabilizer subgroup $\mathrm{SE}(d)_{(0,a)}$. We take the convention that the Lie algebra vector A_{i+d} generates the in-plane rotation in the plane spanned by A_i and a for $i = 1, \ldots, d-1$.

For the case $d = 2$ this gives use two spatial generators, A_1 and A_2, and one rotation generator A_3. Moving to $d = 3$ we have 3

spatial degrees of freedom and 3 rotational degrees of freedom, but only 2 of those rotational degrees of freedom will change the orientation, we denote the generator corresponding to the rotation that preserves the reference axis by A_6. This gives us the following basis:

$$\Big\{ \underbrace{A_1, A_2, A_3,}_{\text{spatial generators}} \quad \underbrace{A_4, A_5,}_{\text{generators of rotations that move } a} \quad \underbrace{A_6}_{\text{generator of rotation that fixes } a} \Big\}.$$

As is illustrated in Fig. 9.3 for $d = 3$ we have a rotational degree of freedom that does not change the orientation of the reference axis. As a result $\mathbb{M}_3$ is not isomorphic to SE(3). It is rather a 5-dimensional quotient of the 6-dimensional Lie group SE(3).

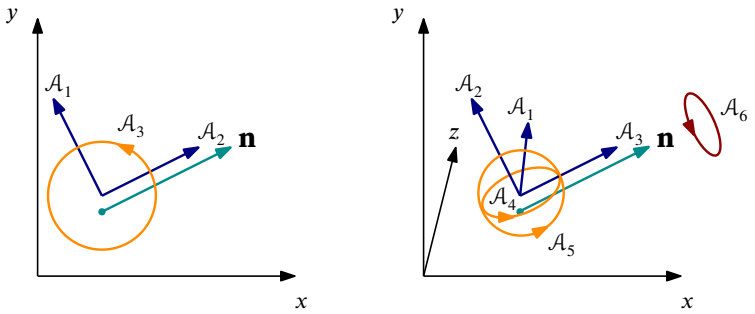


FIGURE 9.3: Illustration of the standard basis of left invariant vector fields $(\mathcal{A}_i)_{i=1}^{D}$ of SE(d) for $d = 2$ on the left and $d = 3$ on the right, note the orientation-preserving rotation generator $\mathcal{A}_6$.

Generalizing this scheme for $d > 3$ we would have the following basis for the Lie algebra:

$$\Big\{ \underbrace{A_1, \ldots, A_d,}_{\text{spatial generators}} \underbrace{A_{d+1}, \ldots, A_{2d-1},}_{\text{generators of rotations that move } a} \underbrace{A_{2d}, \ldots, A_{\frac{1}{2}d(d+1)}}_{\text{generators of rotations that fix } a} \Big\}.$$

(CO)VECTOR FIELDS. We extend the Lie algebra basis vectors to a left-invariant vector field basis through the pushforward of the left multiplication as per

$$(\mathcal{A}_i)_g = (L_g)_* A_i,$$

recall §2.1.

We denote the corresponding covector fields by $\omega^i \in T^*(\mathrm{SE}(d))$. The covector fields are characterized in the usual manner by

$$\omega^i\left(\mathcal{A}_j\right) = \delta^i_j, \tag{9.7}$$

where δ^i_j denotes the Kronecker delta.

METRICS. We introduce metric tensor fields on $\mathrm{SE}(d)$ we will use in terms of just the left-invariant covector fields $\left(\omega^i\right)_{i=1}^{2d-1}$, since the covector fields $\left(\omega^i\right)_{i=2d}^{D}$ correspond to directions that do not change orientation.

DEFINITION 9.1 (Metric tensor fields on $\mathrm{SE}(d)$) Given positive constants $D_S > 0$ and $D_A > 0$, and a nonnegative real number $\mathfrak{e} \geq 0$, we define the left-invariant metric tensor field $\mathcal{G} \in \Gamma(T^{(0,2)}\,\mathrm{SE}(d))$ by

$$\mathcal{G} = \begin{cases} \frac{\omega^d \otimes \omega^d}{D_S} + \frac{\sum_{i=1}^{d-1} \omega^i \otimes \omega^i}{\mathfrak{e}^2 D_S} + \frac{\sum_{i=d+1}^{2d-1} \omega^i \otimes \omega^i}{D_A} & \text{if } \mathfrak{e} > 0, \\ \frac{\omega^d \otimes \omega^d}{D_S} + \frac{\sum_{i=d+1}^{2d-1} \omega^i \otimes \omega^i}{D_A} & \text{if } \mathfrak{e} = 0. \end{cases} \tag{9.8}$$

We think of D_S as the speed of moving in space, where we penalize lateral movement via the additional $\mathfrak{e}$ factor. Similarly, D_A controls how fast we can change direction.

Observe that this sub-Riemannian metric tensor is defined (and invertible) on a sub-bundle of the tangent bundle on the group as it does not contain any of the covectors dual to the subbundle induced by subgroup $\mathrm{SE}(d)_{(0,a)}$. Furthermore it is spatially isotropic orthogonal to the primary spatial direction. Also spherically we impose isotropy in the metric as can be seen from the last term in the above definition.

In light of our proposed metric being degenerate in the direction of $\langle \mathcal{A}_d, \ldots, \mathcal{A}_D \rangle$ we define a subbundle for the nondegenerate directions.

DEFINITION 9.2 (Nondegenerate subbundle)

$$T_{\mathrm{ND}}(g) := \left\{ \dot{g} \in T_g(\mathrm{SE}(d)) \;\middle|\; \omega^i(\dot{g}) = 0 \text{ for all } i = 2d, \ldots, D \right\},$$

for all $g \in \mathrm{SE}(d)$.

SUB-RIEMANNIAN. Henceforth we refer to the $\mathfrak{e} = 0$ case as the sub-Riemannian case where tangent vectors are constrained to the span of $\mathcal{A}_d, \ldots, \mathcal{A}_{2d-1}$. Intuitively when $\mathfrak{e} \downarrow 0$ the other tangent directions get infinite cost and become prohibited. This means that we restrict ourselves to so-called *horizontal* tangent vectors.

DEFINITION 9.3 (Horizontal subbundle)

$$T_{\mathrm{Hor}}(g) := \left\{ \dot{g} \in T_{\mathrm{ND}}(g) \;\middle|\; \omega^i(\dot{g}) = 0 \text{ for all } i = 1, \ldots, d-1 \right\},$$

for all $g \in \mathrm{SE}(d)$.

The metric Definition 9.1 induces an associated norm in the normal manner: if $\dot{g} \in T_{\mathrm{ND}}(g)$, then

$$\|\dot{g}\|_{\mathcal{G}} := \sqrt{\mathcal{G}_g(\dot{g}, \dot{g})}, \tag{9.9}$$

where again in the sub-Riemannian case we only allow $\dot{g} \in T_{\mathrm{Hor}}(g)$.

GRADIENT & DIVERGENCE. Now that we have $\mathrm{SE}(d)$ equipped with a (sub)-Riemannian metric tensor, we can derive the basic tools that are required to formulate our geometric PDEs. These basic tools include the gradient, its norm, and the divergence of a vector field. Let us relabel our parameters as

$$D_i = \begin{cases} D_S & \text{if } i = d, \\ \mathfrak{e}^2 D_S & \text{if } 1 \leq i \leq d-1, \\ D_A & \text{if } d+1 \leq i \leq 2d-1. \end{cases} \tag{9.10}$$

Let $\tilde{U} : \mathrm{SE}(d) \to \mathbb{R}$ carry the axial symmetry:

$$\tilde{U}(x, R) = U(x, Ra), \tag{9.11}$$

for some $U : \mathbb{M}_d \equiv \mathbb{R}^d \times S^{d-1} \to \mathbb{R}$.

Then in the Riemannian setting the gradient of a differentiable function $\tilde{U} : \mathrm{SE}(d) \to \mathbb{R}$ on the group induced by this metric tensor becomes

$$\nabla_{\mathcal{G}} \tilde{U} = \sum_{i=1}^{2d-1} D_i \left(\mathcal{A}_i \tilde{U} \right) \mathcal{A}_i,$$

where the sum only runs to $2d-1$ and not to $\dim(\mathrm{SE}(d)) = D = \frac{1}{2}d(d+1)$ since (9.11) implies that $\mathcal{A}_i \tilde{U} = 0$ for $2d-1 < i \leq D$.

The gradient then has the following norm

$$\left\| \nabla_{\mathcal{G}} \tilde{U} \right\|_{\mathcal{G}}^2 = \sum_{i=1}^{2d-1} D_i \left(\mathcal{A}_i \tilde{U} \right)^2. \tag{9.12}$$

The divergence of a vector field is given by

$$\mathrm{div}_{\mathcal{G}} X = \sum_{i=1}^{2d-1} \mathcal{A}_i \omega^i(X)$$

for all $X \in \Gamma(T_{\mathrm{ND}})$

In the sub-Riemannian setting, where we restrict ourselves to vector fields spanned by $(\mathcal{A}_i)_{i=d}^{2d-1}$, we have

$$\nabla_{\mathcal{G}} \tilde{U} = \sum_{i=d}^{2d-1} D_i \left(\mathcal{A}_i \tilde{U} \right) \mathcal{A}_i$$

and

$$\mathrm{div}_{\mathcal{G}} X = \sum_{i=d}^{2d-1} \mathcal{A}_i \omega^i(X)$$

for all $X \in \Gamma(T_{\mathrm{Hor}})$.

9.3 LOCALLY ADAPTIVE FRAMES ON $\mathrm{SE}(d)$

As an alternative to the left-invariant frame we can choose a frame (and subsequently a metric tensor field) that is adapted to the data, which we also refer to as *gauge frames* in analogy with Duits, Janssen, Hannink, and Sanguinetti (2016). Specifically, instead of having the vector field $\mathcal{A}_d$ as a static forward direction we want to choose a vector field $\mathcal{B}_d$ that locally aligns with the data. In

particular, $\mathcal{B}_d$ can take on a angular component, meaning the local 'straight forward' will follow the curve of the data, consequently flows can better follow curved structures, see Figure 9.4 for an illustration.

We can induce an entire frame in SE(d) *from a choice of main vector, see* DUITS, JANSSEN, HANNINK, *and* SANGUINETTI (2016, *App. A) for details. In this chapter we will focus on the method by which the main gauge vector is obtained.*

Next we will present a singular value decomposition of the Hessian. We will choose the eigenvector associated with the smallest eigenvalue as $\mathcal{B}_d$. Geometrically, this can be seen as the direction in which the gradient changes the least. This procedure is formulated in terms of the *exponential curves* from §2.1.

DEFINITION 9.4 (Main gauge vector.) We define $(\mathcal{B}_d)_g \in T_{\mathrm{ND}}(g)$ for all $g \in \mathrm{SE}(d)$ by

$$
\begin{aligned}
(\mathcal{B}_d)_g &:= \underset{\substack{\dot{g}\in T_{\mathrm{ND}}(g),\\ \|\dot{g}\|_{\mathcal{G}}=1}}{\arg\min} \left\|\mathrm{Hess}\,\tilde{U}(\dot{g},\cdot)\right\|^2_{\mathcal{G}^*} \\
&= \underset{\substack{\dot{g}\in T_{\mathrm{ND}}(g)\\ \|\dot{g}\|_{\mathcal{G}}=1}}{\arg\min} \left\| \left(\frac{d}{dt}\right)_0 \left[\left(L_{g\exp(-t\dot{g})g^{-1}}\right)_* \nabla_{\mathcal{G}}\tilde{U}\left(g\exp(t\dot{g})\right)\right] \right\|^2_{\mathcal{G}},
\end{aligned}
$$

where $\exp \equiv \exp_{\mathrm{SE}(d)}$ is the Lie group exponential map of SE(d) and where we assume that $\tilde{U}$ is such that we have a unique minimizer.

Recall the definition of the *Hessian* from §1.9.

MATRIX FORM. Writing the tangent vector in terms of the local left invariant frame as $\dot{g} = \sum_{i=1}^{2d-1} \dot{g}^i (\mathcal{A}_i)_g \in T(g)$ lets us write out the Hessian as follows:

$$
\begin{aligned}
(\mathcal{B}_d)_g &= \underset{\substack{\dot{g}\in T_{\mathrm{ND}}(g),\\ \|\dot{g}\|_{\mathcal{G}}=1}}{\arg\min} \sum_{i=1}^{2d-1} D_i\left(\dot{g}\left(\mathcal{A}_i\tilde{U}\right)\right)^2 \\
&= \underset{\substack{\dot{g}\in T_{\mathrm{ND}}(g)\\ \|\dot{g}\|_{\mathcal{G}}=1}}{\arg\min} \sum_{i=1}^{2d-1} D_i\left(\sum_{j=1}^{2d-1} \dot{g}^j \left(\mathcal{A}_j\right)_g \left(\mathcal{A}_i\tilde{U}\right)\right)^2. \qquad (9.13)
\end{aligned}
$$

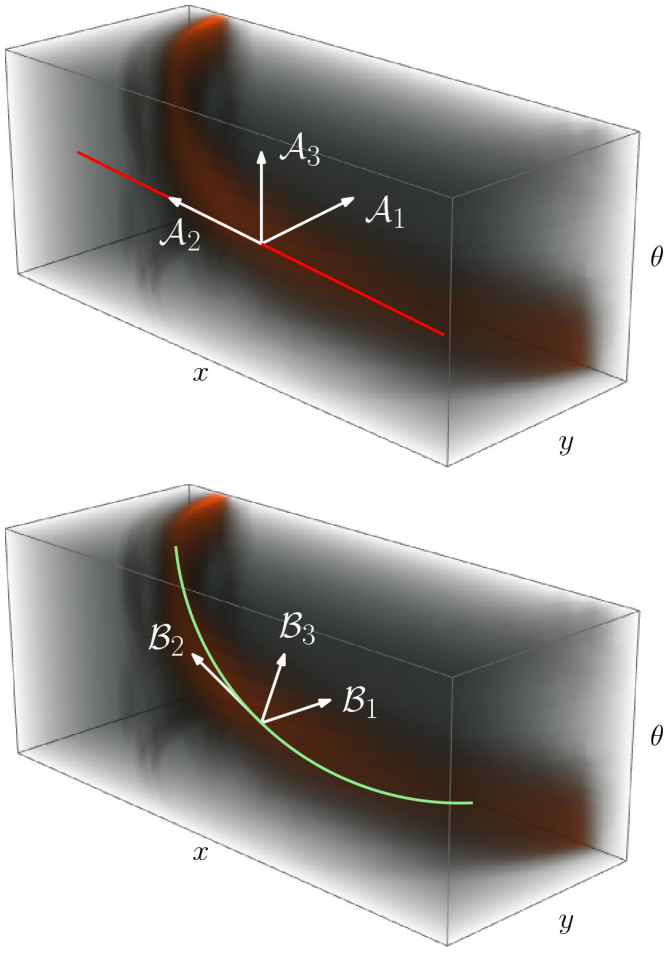


FIGURE 9.4: Illustrating gauge frame fitting in SE(2). *Top*: left invariant frame, the red line indicates the exponential curve associated with $\mathcal{A}_2$. *Bottom*: we choose a frame that takes into account the local curvature, here the green line indicates the exponential curve associated with $\mathcal{B}_2$.

We can write this problem in terms of matrices by defining the following:

$$\dot{\boldsymbol{g}} = \left(\dot{g}^1, \dot{g}^2, \ldots, \dot{g}^{2d-1}\right)^T,$$
$$M = \operatorname{diag}\left(\sqrt{D_1}, \sqrt{D_2}, \ldots, \sqrt{D_{2d-1}}\right),$$
$$K = K\left(\tilde{U}\right)\big|_g := \left[\left(\mathcal{A}_j\right)_g \left(\left(\mathcal{A}_i\right)_g \tilde{U}\right)\right]_{ij}.$$

with i as row index and j as column index.

Using these the objective function in (9.13) becomes

$$(MK\dot{\boldsymbol{g}})^T (MK\dot{\boldsymbol{g}}) = \dot{\boldsymbol{g}}^T K^T M^2 K \dot{\boldsymbol{g}}, \tag{9.14}$$

which we want to minimize under the constraint

$$\sum_{i=1}^{2d-1} \frac{\left(\dot{g}^i\right)^2}{D_i} = \left\| M^{-2}\dot{\boldsymbol{g}} \right\|^2 = 1. \tag{9.15}$$

Taking the derivative of the Lagrangian of this convex optimization problem gives us optimality under the following condition ($\lambda \in \mathbb{R}$):

$$K^T M^2 K \dot{\boldsymbol{g}} = \lambda M^{-2} \dot{\boldsymbol{g}} \iff M^2 K^T M^2 K \dot{\boldsymbol{g}} = \lambda \dot{\boldsymbol{g}}, \tag{9.16}$$

i.e. $\dot{\boldsymbol{g}}$ needs to be an eigenvector of the matrix $M^2 K^T M^2 K$ with eigenvalue λ (serving as the Lagrangian multiplier). If for a moment we rewrite (9.16) as

$$(MKM)^T (MKM) \left(M^{-1}\dot{\boldsymbol{g}} \right) = \lambda \left(M^{-1}\dot{\boldsymbol{g}} \right), \tag{9.17}$$

we see that λ is indeed real since $(MKM)^T (MKM)$ is symmetric. With this eigenvalue and vector the objective function (9.14) evaluates to

$$\begin{aligned} \dot{\boldsymbol{g}}^T K^T M^2 K \dot{\boldsymbol{g}} &= \dot{\boldsymbol{g}}^T M^{-2} M^2 K^T M^2 K \dot{\boldsymbol{g}} \\ &= \dot{\boldsymbol{g}}^T M^{-2} \lambda \dot{\boldsymbol{g}} \\ &= \lambda \dot{\boldsymbol{g}}^T M^{-2} \dot{\boldsymbol{g}} \\ &= \lambda \left\| \dot{g} \right\|_{\mathcal{G}}^2 = \lambda. \end{aligned}$$

This last equation incidentally proves that $M^2 K^T M^2 K$ is positive semi-definite and, more importantly, that to minimize the change in gradient we need to look at the eigenvector belonging to the smallest eigenvalue.

REGULARIZATION. In practice, we do not immediately calculate the eigenvectors and eigenvalues from the scheme we have just

proposed, but for the purpose of stability we first apply a componentwise Gaussian smoothing on the matrix K as follows:

$$\begin{aligned}\tilde{K}\big|_g &:= (G * K)(x, R_n) \\ &= \int_{\mathbb{R}^d} \int_{S^{d-1}} G\left(x - y,\ R_m^T n\right)\ K\big|_{(y,\, R_m)}\ \mathrm{d}\sigma(m)\,\mathrm{d}y,\end{aligned}$$

with the usual surface measure σ on S^{d-1} and with the smoothing kernel

$$G(y, m) := G^{\mathbb{R}^d}_{\rho_s}(0, y) \cdot G^{S^{d-1}}_{\rho_a}(a, m),$$

where G^M_ρ is the heat kernel on the Riemannian manifold M with time/scale $\rho > 0$, the spatial kernel is centered on $0 \in \mathbb{R}^d$ and the orientation kernel is centered on the reference direction $a \in S^{d-1}$.

ISOTROPIC SMOOTHING. It is important in the context of $\mathbb{M}_d$ to choose diffusion that is isotropic spatially (with time/scale ρ_s) and spherically (with time/scale ρ_a) since this is the only diffusion that commutes with the left-invariant vector fields. Note that $G^{\mathbb{R}^d}_{\rho_s}(0, y)$ depends only on $\|y\|$ and $G^{S^{d-1}}_{\rho_a}(a, m)$ depends only on $\arccos(a \cdot m)$ making $G(y, m)$ the heat kernel on the product manifold $\mathbb{R}^d \times S^{d-1}$. This smoothing method is a variant on the one used in DUITS, JANSSEN, HANNINK, and SANGUINETTI (2016).

COMPLETING THE FRAME. The remaining basis vectors are determined by considering a rotation that maps $(\mathcal{A}_d)_g$ to $(\mathcal{B}_d)_g$ and then applying a specific rotation to the remaining $(\mathcal{A}_i)_g$ that keeps the remaining spatial generators spatial. For an illustration see Figure 9.4. How this rotation is chosen and applied is detailed in DUITS, JANSSEN, HANNINK, and SANGUINETTI (2016, App. B).

GAUGE METRIC. Having determined a data adaptive frame $(\mathcal{B}_i)_{i=1}^{2d-1}$ (induced by $\mathcal{B}_d$, recall Figure 9.4), we equip it with the following straightforward metric, where again we rely on the corresponding dual frame $\left(\beta^i\right)_{i=1}^{2d-1} \subset \Gamma(T^*(\mathrm{SE}(d)))$ given by

$$\beta^i(\mathcal{B}_j) = \delta^i_j.$$

DEFINITION 9.5 (Gauge metric) We define the *gauge metric tensor field* $\mathcal{J} \in \Gamma(T^{(0,2)}(\mathrm{SE}(d)))$ as

$$\mathcal{J} := \sum_{i=1}^{2d-1} \beta^i \otimes \beta^i.$$

Which induces a norm on $\dot{g} \in T_g(\mathrm{SE}(d))$:

$$\|\dot{g}\|_{\mathcal{J}} := \sqrt{\mathcal{J}_g(\dot{g}, \dot{g})},$$

a gradient on $\tilde{U} \in C^1(\mathrm{SE}(d))$:

$$\nabla_{\mathcal{J}} \tilde{U} := \sum_{i=1}^{2d-1} \left(\mathcal{B}_i \tilde{U}\right) \mathcal{B}_i,$$

with norm

$$\left\|\nabla_{\mathcal{J}} \tilde{U}\right\|_{\mathcal{J}}^2 = \sum_{i=1}^{2d-1} \left(\mathcal{B}_i \tilde{U}\right)^2,$$

and finally gives the divergence of a vector field $X \in \Gamma(T(\mathrm{SE}(d)))$ as:

$$\mathrm{div}_{\mathcal{J}} X = \sum_{i=1}^{2d-1} \mathcal{B}_i \left(\beta^i(X)\right),$$

which means that if we apply it to a vector field expressed in the gauge frame as $X = \sum_{i=1}^{2d-1} X^i \mathcal{B}_i$ we have:

$$\mathrm{div}_{\mathcal{J}} X = \sum_{i=1}^{2d-1} \mathcal{B}_i X^i.$$

9.4 COHERENCE ENHANCEMENT

Coherence enhancing diffusion is a well-known technique for image enhancement (WEICKERT, 1999). It is intended for line amplification rather than strictly denoising. Crossing-preserving versions on $\mathbb{M}_d$ have been developed (DUITS, JANSSEN, HANNINK, and SANGUINETTI, 2016) and evaluated for denoising. Here, crossing lines are well-enhanced but plateaus and boundaries of line structures are damaged. Therefore we propose to include the coherence enhancement technique into TV and MC flows.

Next we explain how this coherence enhancement operator is constructed from an orientation confidence map.

In $\mathbb{R}^3$, orientation confidence is calculated by the Laplacian in the subspace orthogonal to the line structure. We can take a similar approach in $\mathbb{M}_d$ by taking the Laplacian in the space spanned by $(\mathcal{A}_i)_{i=1,i\neq d}^{2d-1}$. Recall that $\mathcal{A}_d$ is implicitly aligned with the local line structure along $n \in S^{d-1}$. In the gauge-frame setting $\mathcal{B}_d$ is explicitly aligned with the line structure (see Figure 9.4) and therefore we take the Laplacian in the span of $(\mathcal{B}_i)_{i=1,i\neq d}^{2d-1}$.

In the sub-Riemannian case $\mathfrak{e} = 0$ this just reduces to the second derivatives in the $(d-1)$-dimensional spaces spanned by $(\mathcal{A}_i)_{i=d+1}^{2d-1}$ and $(\mathcal{B}_i)_{i=d+1}^{2d-1}$ respectively. With that in mind we define orientation confidence in $\mathrm{SE}(d)$ as follows.

ORIENTATION CONFIDENCE. Let $\tilde{U} : \mathrm{SE}(d) \to \mathbb{R}$, then in the left-invariant case we define

$$C_{\tilde{U},\mathrm{LI}}(g) := \left| \sum_{i=1,i\neq d}^{2d-1} D_i \ (\mathcal{A}_i)_g \left(\mathcal{A}_i \tilde{U}\right) \right|, \tag{9.18}$$

or

$$C_{\tilde{U},\mathrm{GF}}(g) := \left| \sum_{i=1,i\neq d}^{2d-1} (\mathcal{B}_i)_g \ \left(\mathcal{B}_i \tilde{U}\right) \right|. \tag{9.19}$$

in the gauge-frame case.

Note that the $\mathcal{B}_i$'s are normalized with respect to the old metric from Definition 9.1 and as such the parameters D_i are still included in (9.19). In the case that $\mathcal{B}_d$ is aligned with $\mathcal{A}_d$ we have $\mathcal{B}_i = \sqrt{D_i}\mathcal{A}_i$ and (9.18) and (9.19) coincide.

DEFINITION 9.6 (Isotropy factor) Let $c > 0$ be a chosen scaling constant, then the *isotropy factor* is defined as:

$$\alpha_{\tilde{U}}(g) := e^{-\frac{1}{c} C_{\tilde{U}}(g)},$$

with $C_{\tilde{U}}$ defined by (9.18) respectively (9.19).

What is convenient about this quantity is that it gives a number in the range $(0, 1]$ with a number close to zero indicating a high

degree of anisotropy and a 1 indicating perfect isotropy. This is the quantity that we can use to steer flow.

The choice of c controls how steep the decline of the isotropy factor is. Its appropriate value depends on the application and on exactly how the data is represented numerically (normalized to $[0,1]$ in our case) and is best determined heuristically or by histogram. For our experiments we have used $c = 0.2$.

COHERENCE ENHANCEMENT. Using this scalar function $\alpha_{\tilde{U}}$ on the group $\mathrm{SE}(d)$, we can locally modify vectors based on how certain we are the data is locally aligned. We refer to this modification of vector fields as *coherence enhancement*, as in coherence enhancing diffusion (FRANKEN and DUITS, 2009). Tangent vectors (such as the gradient as we will see) are modified as follows. Let X be a vector field on $\mathrm{SE}(d)$, then the coherence enhanced vector field is given as

$$E_{\mathcal{G}}(X) := \omega^d(X)\,\mathcal{A}_d + \alpha_{\tilde{U}} \sum_{\substack{i=1 \\ i\neq d}}^{2d-1} \omega^d(X)\,\mathcal{A}_i, \tag{9.20}$$

for the left-invariant geometry and as

$$E_{\mathcal{J}}(X) := \beta^d(X)\,\mathcal{B}_d + \alpha_{\tilde{U}} \sum_{\substack{i=1 \\ i\neq d}}^{2d-1} \beta^d(X)\,\mathcal{B}_i. \tag{9.21}$$

and for the gauge geometry.

Intuitively, these linear operators $E_{\mathcal{G}}, E_{\mathcal{J}} : T(\mathrm{SE}(d)) \to T(\mathrm{SE}(d))$ preserve the magnitude of the vector in the main direction and weaken it orthogonal to the main direction if we are certain the data is locally aligned to the main direction.

9.5 DESCENDING TO THE HOMOGENEOUS SPACE

So far we have developed two distinct geometries on the group $\mathrm{SE}(d)$ that are summarized in Table 9.1. We can bring these geometries down to the homogeneous space $\mathbb{M}_d$ by considering the natural extension of functions and vector fields on $\mathbb{M}_d$ to $\mathrm{SE}(d)$. Consider a function $U \in C^\infty(\mathbb{M}_d)$, then the function $\tilde{U}$, given by

$$\tilde{U}(g) = \tilde{U}(x,R) := U(x, Ra) \tag{9.22}$$

for all $g = (x, R) \in \mathrm{SE}(d)$, is its natural extension to $\mathrm{SE}(d)$ and is clearly also smooth. Similarly, a tangent vector field $X \in \Gamma(T(\mathrm{SE}(d)))$ can be extended to a tangent vector field $\tilde{X} \in \Gamma(T\mathbb{M}_d)$ as follows:

$$\left(\tilde{X}\tilde{U}\right)(g) = \left(\tilde{X}\tilde{U}\right)(x, R) := (XU)(x, Ra)\,, \tag{9.23}$$

under the additional constraint that $\tilde{X}$ vanishes in the direction induced by the subgroup $\mathrm{SE}(d)_{(0,a)}$, i.e. for all $i \geq 2d$ we have $\omega^i(X) = 0$, this extension is unique.

Having extended functions and vector fields upward to the group, we can apply the tools from Table 9.1 to them and subsequently project the results back to the homogeneous space by the mapping $(x, R) \mapsto (x, Ra)$. This mapping is not injective. Nevertheless, thanks to the metrics from Definition 9.1 and 9.5 being laterally and spherically isotropic and the way we extend functions to the group by (9.22) and (9.23), all the tools we list in Table 9.2 are well-defined on $\mathbb{M}_d$.

Choice of geometry	Left-invariant	Gauge
Frame	$\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{2d-1}$	$\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{2d-1}$
Co-frame	$\omega^1, \omega^2, \ldots, \omega^{2d-1}$	$\beta^1, \beta^2, \ldots, \beta^{2d-1}$
Metric tensor	$\mathcal{G}$	$\mathcal{J}$
Gradient ∇	$\nabla_{\mathcal{G}}$	$\nabla_{\mathcal{J}}$
Divergence div (Riemannian)	$\sum_{i=1}^{2d-1} \mathcal{A}_i \circ \omega^i$	$\sum_{i=1}^{2d-1} \mathcal{B}_i \circ \beta^i$
Divergence div (sub-Riemannian)	$\sum_{i=d}^{2d-1} \mathcal{A}_i \circ \omega^i$	$\sum_{i=d}^{2d-1} \mathcal{B}_i \circ \beta^i$
Norm $\|\cdot\|$	$\|\cdot\|_{\mathcal{G}}$	$\|\cdot\|_{\mathcal{J}}$
Volume form $\mathrm{d}\tilde{\mu}$	$\frac{\omega^1 \wedge \omega^2 \wedge \ldots \wedge \omega^{2d-1}}{\sqrt{D_S}\sqrt{D_S D_A}^{d-1}}$	$\beta^1 \wedge \beta^2 \wedge \ldots \wedge \beta^{2d-1}$

TABLE 9.1: We formulate our PDEs independently of our choice of frame and metric and then apply it to both the left-invariant and gauge geometries.

NONINJECTIVITY. While the choice of mapping $n \in S^{d-1} \mapsto R_n \in \mathrm{SO}(d)$ does not matter for the final result, a choice does have to be made for an implementation when $d \geq 3$. The most straightforward manner is selecting that R_n which is an in-plane rotation,

$\mathbb{M}_d$ tools	Left-invariant	Gauge
$\nabla U(x,n) =$	$\left(\nabla_{\mathcal{G}}\tilde{U}\right)(x,R_n)$	$\left(\nabla_{\mathcal{J}}\tilde{U}\right)(x,R_n)$
$(X,Y)_{\mathbb{M}_d}\Big\vert_{(x,n)} =$	$\mathcal{G}_{(x,R_n)}\left(\tilde{X},\tilde{Y}\right)$	$\mathcal{J}_{(x,R_n)}\left(\tilde{X},\tilde{Y}\right)$
$\|X\|_{\mathbb{M}_d} =$	$\left\|\tilde{X}\right\|_{\mathcal{G}}$	$\left\|\tilde{X}\right\|_{\mathcal{J}}$
$(\operatorname{div} X)(x,n) =$	$\left(\operatorname{div}_{\mathcal{G}}\tilde{X}\right)(x,R_n)$	$\left(\operatorname{div}_{\mathcal{J}}\tilde{X}\right)(x,R_n)$
$\mathrm{d}\mu(x,n) =$	$\mathrm{d}\tilde{\mu}_{\mathcal{G}}(x,R_n)$	$\mathrm{d}\tilde{\mu}_{\mathcal{J}}(x,R_n)$
$(EX)(x,n)$	$\left(E_{\mathcal{G}}\tilde{X}\right)(x,R_n)$	$\left(E_{\mathcal{J}}\tilde{X}\right)(x,R_n)$

TABLE 9.2: The two geometries we developed on $\mathrm{SE}(d)$ can be naturally applied to the homogeneous space $\mathbb{M}_d$ using the natural extension of scalar functions (9.22) and vector fields (9.23). These objects are well defined despite the non-injective mapping used since the choice of particular $R_n \in \mathrm{SE}(d)$ that maps $a \in S^{d-1}$ to $n \in S^{d-1}$ does not affect any of them.

meaning the plane of rotation is spanned by $a \in S^{d-1}$ and $n \in S^{d-1}$. In the two cases where this is not possible (i.e. $n = \pm a$) we pick $R_a = 0$ and $R_{-a} = R_{e_z,\pi}$, where $R_{e_y,\pi}$ denotes the rotation around the axis e_y by an angle π. Concretely the in-plane rotation in 3D is given in terms of the ZYZ-Euler angles α, β, γ by requiring that $\alpha = -\gamma$, which gives the mapping

$$n \mapsto R_{e_z,-\alpha} R_{e_y,\beta} R_{e_z,\alpha}, \tag{9.24}$$

for the unique $\alpha \in [0, 2\pi)$ and $\beta \in (0, \pi)$ so that the resulting rotation maps a to n.

9.6 TOTAL VARIATION, MEAN CURVATURE PDE FLOWS ON $\mathbb{M}_d$

On $\mathbb{R}^n$ the formulation of total variation is built on the identity $\operatorname{div}(fX) = f \operatorname{div}(X) + \nabla f \cdot X$. Similarly on $\mathbb{M}_d$ we have:

$$\begin{aligned} \operatorname{div}(UX) &= U \operatorname{div}(X) + \mathrm{d}U(X) \\ &= U \operatorname{div}(X) + (\nabla U, X)_{\mathbb{M}_d}. \end{aligned} \tag{9.25}$$

From the last equation we deduce the following integration by parts formula:

$$\int_\Omega U(p)\ \mathrm{div}\, X(p)\, \mathrm{d}\mu(p) = \int_\Omega \big(\nabla U(p),\ X(p)\big)_{\mathbb{M}_d}\, \mathrm{d}\mu(p), \qquad (9.26)$$

for all $U \in C^1(\Omega)$ and all smooth vector fields X vanishing at the boundary $\partial\Omega$. This formula allows us to build a weak formulation of TVF on $\mathbb{M}_d$ starting from functions of bounded variation (BV) (Ambrosio, Ghezzi, and Magnani, 2015).

Definition 9.7 (Weak formulation of TV on $\mathbb{M}_d$) Let $\mathfrak{e} \geq 0$. Let $U \in L^1(\Omega)$, recall (9.1). Let $\chi_0(\Omega)$ denote the vector space of smooth vector fields that vanish at the boundary $\partial\Omega$ and let $\varepsilon \geq 0$. Then we define

$$\mathrm{TV}_\varepsilon(U) := \sup_{\substack{\psi \in C_c^\infty(\Omega) \\ X \in \chi_0(\Omega) \\ \|X(p)\|_{\mathfrak{e}}^2 + |\psi(p)|^2 \leq 1}} \int_\Omega \varepsilon\, \psi(p) + U(p)\ \mathrm{div}(X)(p)\ \mathrm{d}\mu(p). \qquad (9.27)$$

If $\mathrm{TV}_0(U) < \infty$ we say that $U \in BV(\Omega)$.

For all $U \in BV(\Omega)$ we have

$$\mathrm{TV}_0(U) \leq \mathrm{TV}_\varepsilon(U) \leq \mathrm{TV}_0(U) + \varepsilon|\Omega|.$$

Recall the sub-Riemannian setting, and recall the notion of horizontal tangent vectors from Definition 9.3. So (9.27) also covers the sub-Riemannian setting, i.e. $\mathfrak{e} = 0$, when setting

$$\|X_p\|_{\mathcal{G}} = \begin{cases} \mathcal{G}_p(X_p, X_p) & \text{if } X_p \text{ is horizontal}, \\ \infty & \text{else}, \end{cases} \qquad (9.28)$$

for any tangent vector $X_p \in T_p\mathbb{M}_d$.

LEMMA 9.1 Let $\varepsilon \geq 0$. For $U \in C^1(\Omega, \mathbb{R})$ we have

$$\mathrm{TV}_\varepsilon(U) = \int_\Omega \sqrt{\|\nabla U(p)\|^2_{\mathbb{M}_d} + \varepsilon^2}\ \mathrm{d}\mu(p). \tag{9.29}$$

Furthermore for $U \in C^2(\mathbb{M}_d, \mathbb{R})$ and $\mathfrak{e}, \varepsilon > 0$ we have that

$$\partial TV_\varepsilon(U) = \mathrm{div} \circ \left(\frac{\nabla U}{\sqrt{\|\nabla U\|^2_{\mathbb{M}_d} + \varepsilon^2}} \right).$$

Proof. First we substitute (9.26) into (9.27), then we apply Gauss's theorem and use $UX|_{\partial\Omega} = 0$. Then we apply Cauchy-Schwarz on $V_p := \mathbb{R} \times T_p\mathbb{M}_d$ for each $p \in \mathbb{M}_d$, with inner product

$$(\epsilon_1, X) \cdot (\epsilon_2, Y) = \epsilon_1\epsilon_2 + (X, Y)_{\mathbb{M}_d},$$

which holds with equality iff the vectors are linearly dependent. Therefore we smoothly approximate

$$\frac{1}{\sqrt{\varepsilon^2 + \|\nabla U\|^2_{\mathbb{M}_d}}}(\varepsilon, \nabla U)$$

by (ψ, X) one obtains (9.29).

For $U \in C^2(\Omega, \mathbb{R})$, $\delta \in C^\infty_c(\Omega, \mathbb{R})$ we obtain

$$\begin{aligned}(\partial TV_\varepsilon(U),\ \delta)_{L^2(\Omega)} &= \lim_{h \downarrow 0} \frac{TV_\varepsilon(U + h\,\delta) - TV_\varepsilon(U)}{h} \\ &\overset{(9.26)}{=} \left(\mathrm{div} \circ \left(\frac{\nabla U}{\sqrt{\|\nabla U\|^2_{\mathbb{M}_d} + \varepsilon^2}} \right),\ \delta \right)_{L^2(\Omega)},\end{aligned}$$

and the result follows. □

For vector fields X on $\mathbb{M}_d$ define the regularized norm:

$$\|X\|_\varepsilon := \sqrt{\|X\|^2_{\mathbb{M}_d} + \varepsilon^2}. \tag{9.30}$$

This is a common way to regularize denominators and we will use Section 9.8 to prove that this approach converges for $\varepsilon \to 0$.

Now we propose the following roto-translation equivariant enhancement PDE on $\Omega \subset \mathbb{M}_d$, recall (9.1).

DEFINITION 9.8 (Equivariant enhancement PDE) Given $U \in BV(\Omega)$ with $\varepsilon > 0$ then we call $W^\varepsilon : \mathbb{M}_d \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ (being smooth) obeying

$$\begin{cases} \frac{\partial W^\varepsilon}{\partial t}(p,t) = \|\nabla W^\varepsilon(p,t)\|_\varepsilon^a \ \operatorname{div}\left(\frac{\nabla W^\varepsilon(\cdot,t)}{\|\nabla W^\varepsilon(\cdot,t)\|_\varepsilon^b}\right)(p), \\ \nabla_{\mathbb{R}^d} W^\varepsilon(p,t) \cdot \mathbf{N}(x) = 0 \\ \qquad\qquad\qquad\qquad \text{for all } p = (x,n) \in \partial\Omega, \ t \geq 0, \\ W^\varepsilon(p,0) = U(p) \qquad\qquad\qquad \text{for all } p \in \Omega, \end{cases} \tag{9.31}$$

the gradient flow started at U with evolution time $t \geq 0$ and parameters $a, b \in \{0,1\}$. Here we use Neumann boundary conditions with $\mathbf{N}(x)$ as the normal to the spatial boundary at $x \in \Omega_f$.
The coherence enhancement version of this PDE is given by replacing div by $\operatorname{div} \circ E$, recall (9.20) and (9.21):

$$\frac{\partial W^\varepsilon}{\partial t}(p,t) = \|\nabla W^\varepsilon(p,t)\|_\varepsilon^a \ \operatorname{div} \circ E\left(\frac{\nabla W^\varepsilon(\cdot,t)}{\|\nabla W^\varepsilon(\cdot,t)\|_\varepsilon^b}\right)(p).$$

VARIANTS. This PDE system on the homogeneous space $\mathbb{M}_d$ has two versions depending on whether one chooses the left invariant or gauge geometry as outlined in Table 9.1 and 9.2.

We then have the following cases:

- For $(a,b) = (1,1)$ we arrive at *mean curvature flow*, studied for $d = 2$ in CITTI, FRANCESCHIELLO, SANGUINETTI, and SARTI (2016).
- For $(a,b) = (0,1)$ we arrive at *total variation flow*, studied for $d = 2$ in CHAMBOLLE and POCK (2019).

- For $(a,b) = (0,0)$ we arrive at a *linear diffusion* for which exact smooth solutions exist for both $d = 2$ and $d = 3$ PORTEGIES and DUITS (2017).

IN TERMS OF CURVATURE. By the product rule (9.25) the right-hand side of (9.31) for $\varepsilon \downarrow 0$ becomes

$$\frac{\partial W^0}{\partial t} = \|\nabla W^0\|^{a-b} \Delta W^0 + 2b\, \overline{\kappa}_I \, \|\nabla W^0\|^a, \tag{9.32}$$

with the mean curvature $\overline{\kappa}_I(p,t)$ of the level set

$$\left\{ q \in \mathbb{M}_d \;\middle|\; W^0(q,t) = W^0(p,t) \right\},$$

akin to SAPIRO (2006, Ch. 3.2), and with (possibly hypo-elliptic) Laplacian $\Delta = \mathrm{div} \circ \nabla$.

LACK OF REGULARITY. For MCF and TVF smooth solutions to the PDE (9.31) exist only under special circumstances. This lack of regularity is an advantage in image processing to preserve step-edges and plateaus in images, yet it forces us to define a concept of weak solutions. Here, we distinguish between MCF and TVF:

For MCF one relies on viscosity solution theory developed by EVANS and SPRUCK (1991), see also GIGA and SATO (1991) and SATO (1992) for the case of MCF with Neumann boundary conditions. In CITTI, FRANCESCHIELLO, SANGUINETTI, and SARTI (2016, Thm 3.6) existence of C^1-viscosity solutions is shown for $d = 2$.

For TVF we will rely on gradient flow theory by AMBROSIO, GIGLI, and SAVARÉ (2005) and BRÉZIS (1973).

SUB-RIEMANNIAN CONVERGENCE. In this work we do not address convergence of our PDE-solutions in the sub-Riemannian setting $\mathfrak{e} \downarrow 0$, and we only focus on convergence results for $\varepsilon \downarrow 0$. In previous work—by others—convergence to the sub-Riemannian setting is addressed for special cases. For the special case $(a,b) = (0,0)$ convergence of the solutions with respect to $\mathfrak{e} \downarrow 0$ is clear from the exact solutions see PORTEGIES and DUITS (2017, Ch. 2.7). For such convergence in the challenging case $(a,b) = (1,1)$ (MCF), see BASPINAR (2018) and CITTI, FRANCESCHIELLO, SANGUINETTI, and SARTI (2016). For Eikonal PDEs convergence of viscosity solutions towards the sub-Riemannian setting holds as well, see DUITS,

Meesters, Mirebeau, and Portegies (2018, Thm. 2). It is therefore interesting to see if convergence results towards the sub-Riemannian setting hold for the general case including the TVF case, but this falls outside the scope of this work. In Section 9.8 we only focus on convergence results for $\varepsilon \downarrow 0$ for $\mathfrak{e} > 0$ fixed.

9.7 NUMERICS

We implemented the PDE system (9.31) by Euler forward time discretization, relying on standard B-spline or linear interpolation techniques for derivatives in the underlying tools on $\mathbb{M}_d$ given by Table 9.2. For details see Creusen, Duits, Vilanova, and Florack (2013) and Franken and Duits (2009). Also, the explicit upper bounds for stable choices of stepsizes can be derived by the Gershgorin circle theorem.

For $d = 2$ the discretization is straightforward (Franken and Duits, 2009), for $d = 3$ we discretized per Creusen, Duits, Vilanova, and Florack (2013) in the software package *Lie Analysis* for Mathematica developed by Martin, Bekkers, and Duits (2017), to our PDEs of interest (9.31) on $\mathbb{M}_3$.

The Euler-forward discretizations are not unconditionally stable. For $a = b = 0$, the *Gershgorin circle theorem* (Creusen, Duits, Vilanova, and Florack, 2013, Ch. 4.2) gives the stability bound

$$\Delta t \leq (\Delta t)_{\text{crit}} := \left(\frac{(d-1)D_A + D_S}{2h^2} + \frac{(d-1)D_A}{2h_a^2} \right)^{-1},$$

when using linear interpolation with spatial stepsize h and angular stepsize h_a. In our experiments, for $d = 2$ we set $h = 1$ and for $d = 3$ we took $h_a = \frac{\pi}{25}$ using an almost uniform spherical sampling from a tessellated isocahedron with $N_A = 162$ points. TVF required smaller times steps when ε decreases. Keeping in mind (9.32) but then applying the product rule (9.25) to the case $0 < \varepsilon \ll 1$, we concentrate on the first term as it is of order ε^{-1} when the gradient vanishes. Then we find $\Delta t \leq \varepsilon \cdot (\Delta t)_{crit}$ for TVF. For MCF we do not have this limitation.

While both the quantitative and qualitative results of our proposed methods are encouraging we have to end our numerics section mentioning the computational cost. In Table 9.3 we summarize the relative computational time of our methods versus

spatial *Perona-Malik*, this summary shows our methods being several orders of magnitude slower. In the interest of fairness we need to add that benchmarking our prototype Mathematica implementation against the built-in Perona-Malik implementation is not a fair comparison, we are confident that an optimized native implementation would fare much better in a performance comparison.

Method	Relative computational time
Perona-Malik 2D	**1.0**
Left-invariant MCF	88.8
Left-invariant TVF	60.4
Gauge MCF	149.1
Gauge TVF	123.6

TABLE 9.3: Relative computational time needed by our methods relative to spatial Perona-Malik for the denoising results in Figure 9.10. The Perona-Malik implementation is built-in while ours is a naively implemented proof-of-concept, hence the benchmark is not comparing optimized implementation against optimized implementation.

9.8 GRADIENT FLOW & CONVERGENCE

In this section we provide a gradient flow formulation that we will use to prove the convergence of our regularization scheme for TVF. The reader who is more interested in the experimental results than the technical convergence results can safely choose to skip this section and continue reading Section 9.10.

PRELIMINARIES. The total variation flow can be seen as a gradient flow of a lower-semicontinuous, convex functional in a Hilbert space, as we explain next. If $F : H \to [0, \infty]$ is a proper (i.e. not everywhere equal to infinity), lower semicontinuous, convex functional on a Hilbert space H (not to be confused with the subgroup H above, as we will not need the subgroup anymore we will stick with convention and use H for the Hilbert space

from now on), the subdifferential of F in a point u in the finiteness domain of F is defined as

$$\partial F(u) := \left\{ z \in H \;\middle|\; (z, v-u)_H \leq F(v) - F(u) \text{ for all } v \in H \right\}.$$

The subdifferential is closed and convex, and thereby it has an element of minimal norm, called "the gradient of F in u" denoted by $\operatorname{grad} F(u)$. Let $u_0 \in H$ be in the closure of the finiteness domain of F. As per Brezis-Komura theory (Ambrosio, Gigli, and Savaré, 2005, Thm. 2.4.15) there is a unique locally absolutely continuous curve $u : [0, \infty) \to H$ such that

$$u'(t) = -\operatorname{grad} F(u(t))$$

for a.e. $t > 0$ and $\lim_{t \downarrow 0} u(t) = u_0$. We call $u : [0, \infty) \to H$ the gradient flow of F starting at u_0.

Recall the definition of Ω in (9.1), then the function $\mathrm{TV}_\epsilon : L^2(\Omega) \to [0, \infty]$ is lower-semicontinuous and convex for every $\epsilon \geq 0$. This allows us to generalize solutions to the PDE (9.31) as follows:

Definition 9.9 Let $U \in \Xi := \mathrm{BV}(\Omega) \cap L^2(\Omega)$. We define by $t \mapsto W^\epsilon(\cdot, t)$ the gradient flow of TV_ϵ starting at U.

A functional $\Phi : H \to (-\infty, \infty]$ is said to be λ-convex for some $\lambda \in \mathbb{R}$ if

$$u \mapsto \Phi(u) - \frac{\lambda}{2}\|u\|^2$$

is convex. In that case, the functional

$$u \mapsto \Phi(u) - \frac{\lambda}{2}\|u - v\|^2$$

is convex as well, for arbitrary $v \in H$, because the latter functional deviates from the first by an affine functional.

We first prove a stability estimate for the minimization of $1/\tau$-convex functionals.

Lemma 9.2 (Convex stability) Let $\tau > 0$. If a functional $\Phi : H \to (-\infty, \infty]$ on H is $1/\tau$-convex, and u^* is its unique minimizer, then for all $u \in H$,

$$\frac{1}{2\tau}\|u - u^*\|^2 \leq \Phi(u) - \Phi(u^*).$$

This lemma is an extension of a standard result regarding strongly convex functionals—see, e.g. NESTEROV (2013, Thm. 2.1.7)—but with no assumptions on differentiability.

Proof. The functional $\Psi : H \to (-\infty, \infty]$ given by

$$\Psi(u) := \Phi(u) - \Phi(u^*) - \frac{1}{2\tau}\|u - u^*\|^2$$

is convex. It is sufficient to show that Ψ is nonnegative. If it were not, there would exist a $v \in H$ such that $\Psi(v) < 0$. We will show that then, for t small enough, $\Phi(tv + (1-t)u^*) < \Phi(u^*)$, contradicting that u^* is a minimizer. We first have by definition that, for $t \in (0, 1)$,

$$\begin{aligned}\Phi(tv + (1-t)u^*) - \Phi(u^*) - \frac{t^2}{2\tau}\|v - u^*\|^2 \\ = \Psi(tv + (1-t)u^*).\end{aligned}$$

By the convexity of Ψ,

$$\begin{aligned}\Psi(tv + (1-t)u^*) &\leq t\Psi(v) + (1-t)\Psi(u^*) \\ &= t\Psi(v).\end{aligned}$$

Combining the two inequalities, we find

$$\Phi(tv + (1-t)u^*) - \Phi(u^*) \leq t\Psi(v) + O(t^2),$$

so that indeed, for t small enough, $\Phi(tv + (1-t)u^*) < \Phi(u^*)$, leading to the announced contradiction.

Therefore, Ψ is nonnegative, which means that

$$\frac{1}{2\tau}\|u - u^*\|^2 \leq \Phi(u) - \Phi(u^*)$$

for all $u \in H$. □

For a proper (i.e. not everywhere equal to ∞), lower semicontinuous, convex functional F, and $\tau > 0$, define the so-called (ROCKAFELLAR and WETS, 2009) *proximal* operator $J_\tau^F : H \to H$ by

$$J_\tau^F[u_0] := \operatorname*{arg\,min}_{u \in H} \left(\frac{1}{2\tau}\|u - u_0\|^2 + F(u) \right).$$

LEMMA 9.3 (Gradient flow estimate) Let $\delta > 0$ and let $F, G : H \to [0, \infty]$ be two non-negative, proper, lower semicontinuous, convex functionals on a Hilbert space H, such that for all $u \in H$,

$$F(u) - \delta \leq G(u) \leq F(u) + \delta. \tag{9.33}$$

Let $u_0, v_0 \in H$, such that

$$|\partial F|(u_0) \leq L \quad \text{and} \quad |\partial G|(v_0) \leq L. \tag{9.34}$$

Then, we have the following estimate for the gradient flow $u : [0, \infty) \to H$ of F starting at u_0 and the gradient flow $v : [0, \infty) \to H$ of G starting at v_0:

$$\|u(t) - v(t)\| \leq \begin{cases} 4\sqrt{\delta t} + \|u_0 - v_0\| & \text{for } 0 \leq t \leq \frac{\delta}{L^2} \\ 8\sqrt[3]{L\delta t^2} + \|u_0 - v_0\| & \text{for } t > \frac{\delta}{L^2}. \end{cases}$$

Proof. Let $\tau > 0$ and let $u_1^F := J_\tau^F[u_0]$ and $v_1^G := J_\tau^G[v_0]$. Set also $v_1^F := J_\tau^F[v_0]$ and $u_1^G := J_\tau^G[u_0]$. Then, using the definition of v_1^F in the second inequality below, we find

$$\begin{aligned} &\frac{1}{2\tau}\|v_1^F - v_0\|^2 + G(v_1^F) \\ &\overset{(9.33)}{\leq} \frac{1}{2\tau}\|v_1^F - v_0\|^2 + F(v_1^F) + \delta \\ &\leq \frac{1}{2\tau}\|v_1^G - v_0\|^2 + F(v_1^G) + \delta \\ &\overset{(9.33)}{\leq} \frac{1}{2\tau}\|v_1^G - v_0\|^2 + G(v_1^G) + 2\delta. \end{aligned}$$

Because the functional

$$v \mapsto \frac{1}{2\tau}\|v - v_0\|^2 + G(v)$$

is $1/\tau$-convex, it follows by Lemma 9.2 that

$$\frac{1}{2\tau}\|v_1^F - v_1^G\|^2 \leq 2\delta.$$

Now we use that J_τ^F is non-expansive (AMBROSIO, GIGLI, and SAVARÉ, 2005, Eq. 4.0.2), so

$$\|u_1^F - v_1^F\| = \|J_\tau^F(u_0) - J_\tau^F(v_0)\| \leq \|u_0 - v_0\|.$$

We conclude that

$$\|u_1^F - v_1^G\| \le \|u_0 - v_0\| + 2\sqrt{\delta\tau}.$$

By iterating this estimate, we derive

$$\|(J_\tau^F)^n[u_0] - (J_\tau^G)^n[v_0]\| \le \|u_0 - v_0\| + 2n\sqrt{\delta\tau}. \tag{9.35}$$

The a priori estimate (AMBROSIO, GIGLI, and SAVARÉ, 2005, Thm. 4.0.4(v)) yields that the gradient flows u and v of F and G respectively are approximated well by $(J_{t/n}^F)^n[u_0]$ and $(J_{t/n}^G)^n[v_0]$. More precisely, for $t > 0$ and $n > 0$, the a priori estimate gives

$$\left\|u(t) - (J_\tau^F)^n\right\| \le \frac{Lt}{\sqrt{2}n} \quad \text{and} \quad \left\|v(t) - (J_\tau^G)^n\right\| \le \frac{Lt}{\sqrt{2}n}.$$

By these a priori estimates and the estimate for the discrete flows (9.35), we see that

$$\begin{aligned}\|u(t) - v(t)\| &\le \|u(t) - (J_{t/n}^F)^n[u_0]\| + \|v(t) - (J_{t/n}^F)^n[v_0]\| \\ &\quad + \|(J_{t/n}^F)^n[u_0] - (J_{t/n}^G)^n[v_0]\| \\ &\le \sqrt{2}L\frac{t}{n} + 2n\sqrt{\frac{\delta t}{n}} + \|u_0 - v_0\|.\end{aligned}$$

To derive the final estimates, we need to make good choices for n. If $0 \le t \le \delta/L^2$, we take $n = 1$ and obtain

$$\begin{aligned}\|u(t) - v(t)\| &\le \sqrt{2}Lt + 2\sqrt{\delta t} + \|u_0 - v_0\| \\ &\le 4\sqrt{\delta t} + \|u_0 - v_0\|.\end{aligned}$$

If $t > \delta/L^2$, we choose $n = \lceil L^{2/3}(t/\delta)^{1/3}\rceil$, which is larger than or equal to 2. In that case,

$$n/2 \le n - 1 < L^{2/3}(t/\delta)^{1/3} \le n.$$

We then obtain

$$\|u(t) - v(t)\| \le 8L^{1/3}\delta^{1/3}t^{2/3} + \|u_0 - v_0\|.$$

□

We now know that the gradient flows of F and G are close when the slopes $|\partial F|(u_0)$ and $|\partial G|(v_0)$ are bounded. This assumption can be rather stringent. We will relax it, and merely require that $F(u_0)$ and $G(v_0)$ are bounded by some constant $E > 0$, in exchange for a bound between gradient flows that is slightly worse. Our approach will be to run the gradient flow for a small time s from u_0 and v_0, and use the regularizing property of the gradient flow to conclude a slope bound. On the other hand, if s is small, $u(s)$ and $v(s)$ will be close to u_0 and v_0. We will then choose s (almost) optimally to derive a bound between the gradient flows.

9.9 STRONG L^2-CONVERGENCE OF TV-FLOWS

We prove the convergence, stability and accuracy of TV-flows by considering them as the gradient flows of the family of functionals TV_ε. The theory of contraction semigroups (AMBROSIO, GIGLI, and SAVARÉ, 2005, Ch. 4) will allow us to show that as $\varepsilon \to 0$ the gradient flow of TV_ε converges to the gradient flow of TV_0 in the L^2 sense.

THEOREM 9.4 (Gradient flow convergence) Let $F : H \to [0, \infty]$ and $G : H \to [0, \infty]$ be two proper, lower semicontinuous, convex functionals on a Hilbert space H, such that

$$F(u) - \delta \leq G(u) \leq F(u) + \delta$$

for all $u \in H$. Let $u_0, v_0 \in H$ be such that $F(u_0) \leq E$ and $G(v_0) \leq E$ and $\|u_0 - u^*\| \leq M$ and $\|v_0 - v^*\| \leq M$, for some constants $E, M > 0$, where u^* and v^* minimize F and G respectively. Then the gradient flow $u : [0, \infty) \to H$ of F starting at u_0, and the gradient flow $v : [0, \infty) \to H$ of G starting at v_0 satisfy

$$\|u(t) - v(t)\| \leq 16(ME\delta t^2)^{1/5} + \|u_0 - v_0\|$$

for all $0 \leq t < E^6 M^6 / \delta^9$.

Proof. By the *evolution variational inequality* (AMBROSIO, GIGLI, and SAVARÉ, 2005, Thm. 4.0.4(iii)), we know that for all $s > 0$

$$\|u(s) - u_0\| \leq \sqrt{2sF(u_0)}$$

and

$$\|v(s) - v_0\| \le \sqrt{2sG(v_0)}.$$

By the regularizing property (AMBROSIO, GIGLI, and SAVARÉ, 2005, Thm. 4.0.4(ii)),

$$|\partial F|(u(s)) \le \frac{1}{s}\|u_0 - u^*\| \le \frac{M}{s} \tag{9.37a}$$

and

$$|\partial G|(v(s)) \le \frac{1}{s}\|v_0 - v^*\| \le \frac{M}{s} \tag{9.37b}$$

where u^* minimizes F and v^* minimizes G.

Because the gradient flow is a non-expansive semigroup (AMBROSIO, GIGLI, and SAVARÉ, 2005, Thm. 4.0.4(iv)), we obtain

$$\begin{aligned}\|u(t) - v(t)\| &\le \|u(t+s) - v(t+s)\| + \|u(t+s) - u(t)\| \\ &\quad + \|v(t+s) - v(t)\| \\ &\le \|u(t+s) - v(t+s)\| + \|u(s) - u_0\| \\ &\quad + \|v(s) - v_0\|.\end{aligned}$$

Now assume $t < E^6M^6/\delta^9$. We will want to choose s (almost) optimally, depending on t. We choose

$$s = \frac{M^{2/5}\delta^{2/5}t^{4/5}}{E^{3/5}}$$

and note that with $L := M/s$, we have

$$t \ge \frac{\delta}{L^2}.$$

By the slope estimates (9.37) we can apply Lemma 9.3 to the gradient flows starting at $u(s)$ and $v(s)$, to obtain

$$\begin{aligned}\|u(t) - v(t)\| &\le 8M^{1/3}s^{-1/3}\delta^{1/3}t^{2/3} + \|u(s) - v(s)\| \\ &\quad + \|u(s) - u_0\| + \|v(s) - v_0\| \\ &\le 8M^{1/3}s^{-1/3}\delta^{1/3}t^{2/3} + 2\|u(s) - u_0\| \\ &\quad + 2\|v(s) - v_0\| + \|u_0 - v_0\| \\ &\overset{(9.36)}{\le} 8M^{1/3}s^{-1/3}\delta^{1/3}t^{2/3} + \sqrt{32sE} + \|u_0 - v_0\| \\ &= 16M^{1/5}\delta^{1/5}t^{2/5}E^{1/5} + \|u_0 - v_0\|.\end{aligned}$$

□

If, for the general result of Theorem 9.4, we take $F = \mathrm{TV}_0$, $G = \mathrm{TV}_\varepsilon$ and $\delta = \varepsilon\,|\Omega|$ we obtain the following result.

COROLLARY 9.5 (Strong L^2-convergence, stability and accuracy of TV-flows) Let $U \in L^2(\Omega)$ and let W^ε be the gradient flow of TV_ε starting at U and $\varepsilon, \mathfrak{e} \geq 0$. Let $t \geq 0$. Let $\delta = \varepsilon|\Omega|$, then

$$\lim_{\varepsilon \downarrow 0} W^\varepsilon(\cdot, t) = W^0(\cdot, t) \quad \text{in} \quad L^2(\Omega).$$

More precisely, for all $U \in \mathrm{BV}(\Omega)$, $E \geq \mathrm{TV}_0(U) + \delta$, $M \geq \|U\|$ and $0 \leq t < E^6 M^6/\delta^9$ we have that

$$\|W^\varepsilon(\cdot, t) - W^0(\cdot, t)\|_{L^2(\Omega)} \leq 16\Big(ME\delta t^2\Big)^{\frac{1}{5}}.$$

9.10 EXPERIMENTS

In our experiments, we aim to enhance contour and fiber trajectories in medical images and to remove noise. Lifting the image $f : \mathbb{R}^d \to \mathbb{R}$ towards its orientation lift $U : \mathbb{M}_d \to \mathbb{R}$ defined on the space of positions and orientations $\mathbb{M} = \mathbb{R}^d \rtimes S^{d-1}$ preserves crossings (FRANKEN and DUITS, 2009) and avoids leakage of wavefronts (DUITS, MEESTERS, MIREBEAU, and PORTEGIES, 2018).

For our experiments for $d = 3$ the initial condition $U : \mathbb{M}_3 \to \mathbb{R}^+$ is a *fiber orientation density function* (FODF) obtained from DW-MRI data (PORTEGIES, FICK, et al., 2015).

For our experiments for $d = 2$ the initial condition U is an invertible orientation score (OS) that we sampled on 8 equidistant orientations.

For both $d = 2$ and $d = 3$ we show the advantages of TVF and MCF over crossing-preserving diffusion flows (CREUSEN, DUITS, VILANOVA, and FLORACK, 2013; FRANKEN and DUITS, 2009) on $\mathbb{M}_d$.

Finally, we include denoising experiments where we show qualitative and quantitative results where comparison with the well-known denoising technique *BM3D* (DABOV, FOI, KATKOVNIK, and EGIAZARIAN, 2007) show advantages and good results.

IMAGE ENHANCEMENT & DENOISING. In accordance with the workflow in Figure 9.1 we go through the following steps:

$$f \mapsto \mathcal{W}_\psi f \mapsto \Phi_t \circ \mathcal{W}_\psi f \mapsto f_t^a := \mathcal{W}_\psi^* \circ \Phi_t \circ \mathcal{W}_\psi f \approx \int_{S^1} \Phi_t(\mathcal{W}_\psi f)(\cdot, n)\, \mathrm{d}\mu_{S^1}(n). \tag{9.38}$$

for $t \geq 0$. With respect to the final step we recall that we use cake wavelets that allow for sharp approximate reconstruction by integration over angles only. Here $U \mapsto W(\cdot, t) = \Phi_t(U)$ denotes the flow operator on $\mathbb{M}_2$ (9.31). Hence the initial condition for our TVF/MCF-PDE (9.31) is set by an orientation score of image $f : \mathbb{R}^2 \to \mathbb{R}$ given by (9.2).

We refer to the different methods we experimented with, by the following terms.

- *MCF*: we set $(a, b) = (1, 1)$.
- *TVF*: we set $(a, b) = (0, 1)$.
- *Left invariant*: we use the left-invariant geometry as per the first column of Table 9.1.
- *Gauge*: we use the locally-adaptive-frames geometry as per the second column of Table 9.1.
- *Isotropic*: we set $\mathfrak{e} = 1$.
- *Anisotropic*: we set $\mathfrak{e} = 0.25$.
- With *coherence enhancement*: we use the PDE with the E operator as per (9.31).

So that *isotropic gauge TVF with coherence enhancement* for example would equate to setting $(a, b) = (0, 1)$, $\mathfrak{e} = 1$, includes the E operator in the PDE and uses the second column of Table 9.1 to define our geometrical objects.

For quantitative comparison we will look at relative L^1 and L^2 errors, meaning if we have a (clean) source image f_{source} and a denoised image f_t that has been processed to time t we calculate the relative error as:

$$\frac{\|f_t - f_{source}\|}{\|f_{source}\|},$$

with the corresponding L^1 or L^2 norm.

We will test against two types of noise: Gaussian and correlated.

GAUSSIAN NOISE. We apply Gaussian noise with standard deviation 0.2 to our normalized (to $[0, 1]$) source image, the original and noisy images are shown in Figure 9.8a respectively 9.8b.

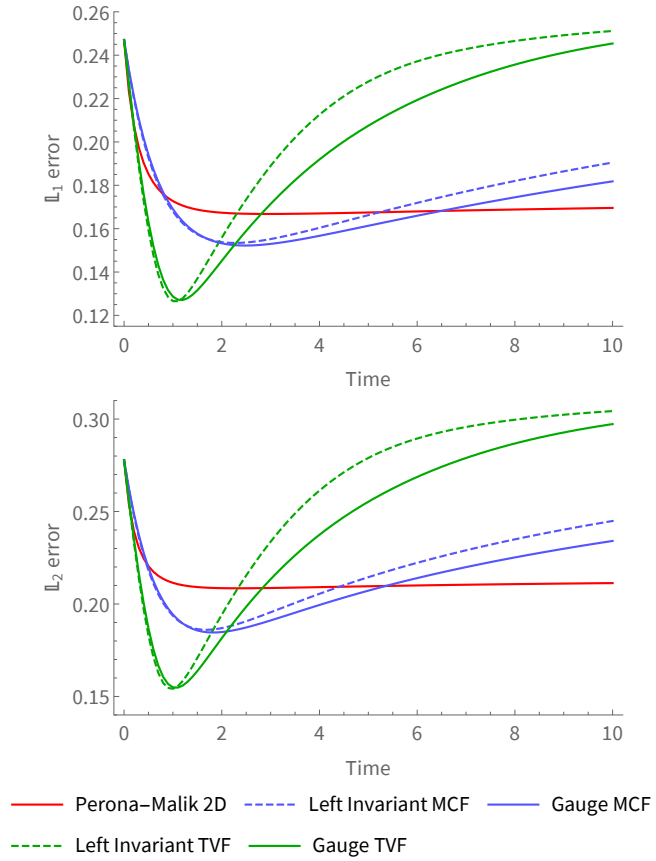


FIGURE 9.5: Isotropic ($\mathfrak{e} = 1$) evolution of relative L^1 and L^2 errors over time of the collagen image with Gaussian noise benchmarked against spatial Perona-Malik. Remaining parameters are $D_S = 1$, $D_A = .01$ and $\varepsilon = .001$

In Figure 9.5 we show how the errors progress with $t \geq 0$ for the isotropic ($\mathfrak{e} = 1$) case without coherence enhancement (i.e. without E). For comparison we plot the same error with spatial Perona-Malik. While Perona-Malik is clearly more stable and resilient to over-smoothing, both our proposed methods have much smaller minimal errors.

TIMESCALE INTERPRETATION. The different methods work on different timescales, we scale these to be able to plot the results together but no meaning should be attributed to one method obtaining a minimum earlier than another. The error graphs just show:

- how large the minimal error is and
- how fast the image deteriorates after this minimum has been reached.

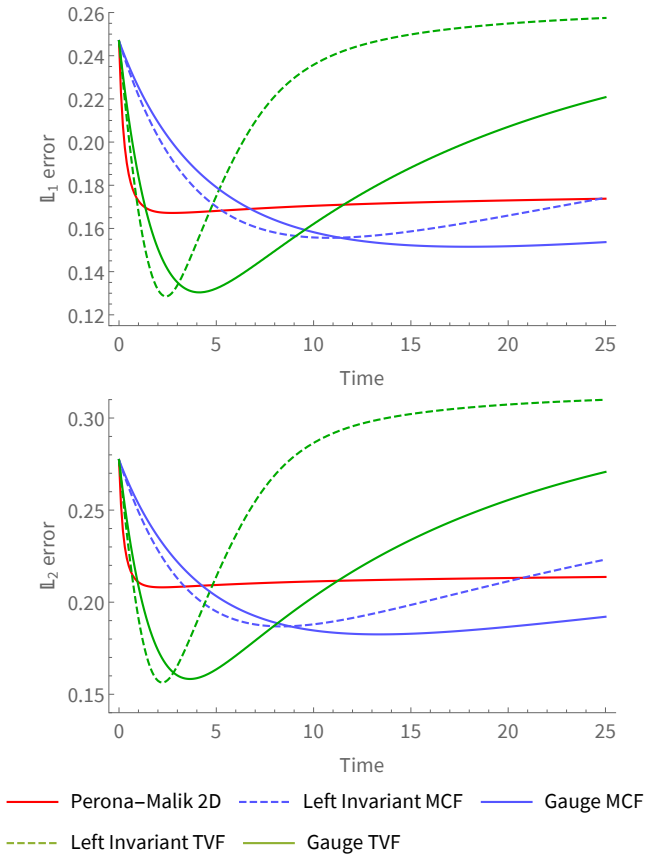


FIGURE 9.6: Anisotropic ($\mathfrak{e} = .25$) evolution of relative L^1 and L^2 errors over time of the collagen image with Gaussian noise benchmarked against spatial Perona-Malik. Run with parameters $D_S = 1$, $D_A = .01$ and $\varepsilon = .001$. The resulting errors are slightly worse than the isotropic setup from Figure 9.5

In our next experiment we increase anisotropy by setting $\mathfrak{e} = .25$, the resulting errors are plotted in Figure 9.6. We gain no improvement in minimal error while requiring more computational cycles to reach the minimum, from which we conclude that for this application isotropic processing is more desirable.

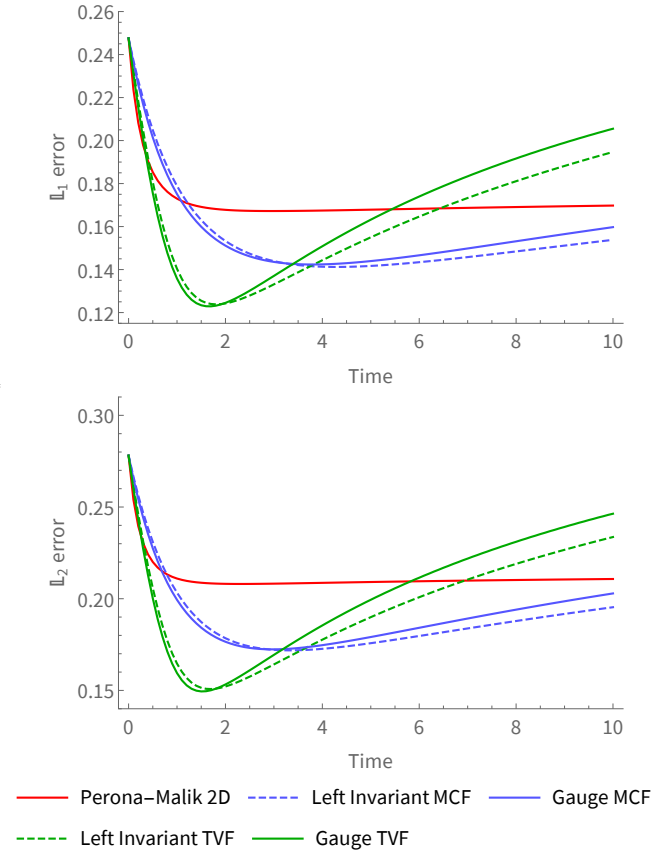


FIGURE 9.7: Isotropic with coherence enhancement evolution of relative L^1 and $L^2$2 errors over time of the collagen image with Gaussian noise benchmarked against spatial Perona-Malik. Run with parameters $D_S = 1$, $D_A = .01$, $\varepsilon = .001$ and $c = .2$

In Figure 9.7 we show the errors for the isotropic setup with coherence enhancement included (for $c = 0.2$). We get a very minor improvement in minimal errors and a decent improvement in over-smoothing stability, although still not on the level of Perona-Malik. It is remarkable that with coherence enhancement included the data adaptive geometry is less stable than the left invariant

geometry, we observe that combining two different methods of adapting to the data is counter-productive in this instance.

For a qualitative comparison of the different isotropic methods with coherence enhancement we over-smooth the collagen image past the time of its lowest L^2 error with a factor of two, the corresponding qualitative results are shown in Figure 9.8.

CORRELATED NOISE. For correlated noise we apply a Gaussian filter with $\sigma = 1.0$ to Gaussian noise with $\sigma = 0.2$. The error evolution for the isotropic methods is plotted in Figure 9.9. We observe that MCF performs worse in this setting against correlated noise, both in minimal error and stability it does not do as well as spatial Perona-Malik. TVF on the other hand has a better minimal error than Perona-Malik at the cost of stability. The stability does somewhat improve if we turn on the locally adaptive frames.

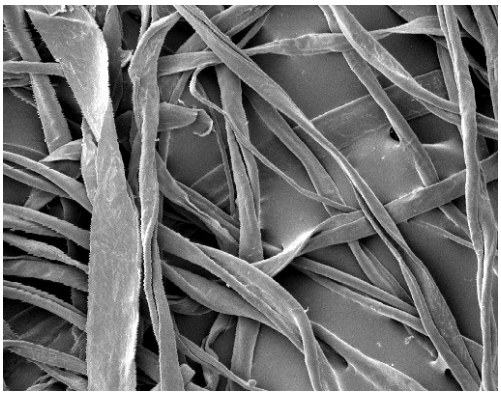
The error evolution of the experiment including the use of coherence enhancement is displayed in Figure 9.10. Overall this improves the results but MCF still exhibits an inferior performance in this setting. TVF sees both an improvement in minimal error and stability. As with the Gaussian noise experiment we see that turning on both coherence enhancement and locally adaptive frames is counterproductive.

A qualitative comparison of the methods against correlated noise is shown in Figure 9.11, where again we smooth twice the optimal time. We observe the same general trend as in Figure 9.8, all methods do a good job of preserving edges but TVF stands out in clearing the plateaus.

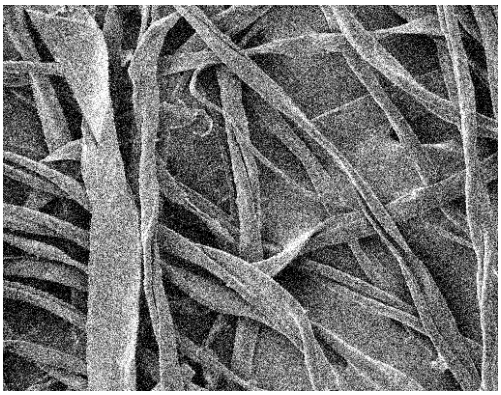
As a final method to compare against we look at BM3D (DABOV, FOI, KATKOVNIK, and EGIAZARIAN, 2007). Both BM3D and our methods share a dependence on some prior knowledge for optimal performance: BM3D requires us to know the standard deviation of the noise and our method requires us to know the optimal evolution time.

We make a qualitative comparison of removing correlated noise between BM3D and left invariant TVF by smoothing 1.5× past the optimal L^2 error: 1.5× the optimal standard deviation in case of BM3D and 1.5× the processing time in case of TVF. The resulting images are shown in Figure 9.12.

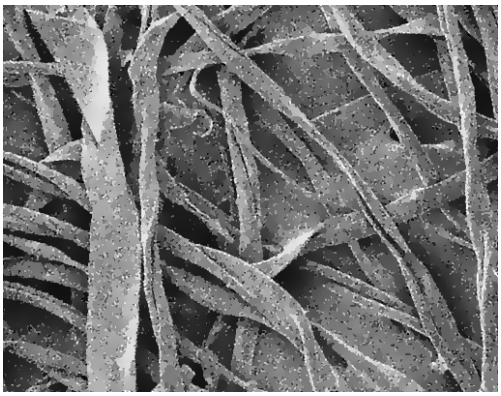
For a broader comparison, we compute *peak signal-to-noise ratios* (PSNR) for the collagen image we already saw and for two addi-



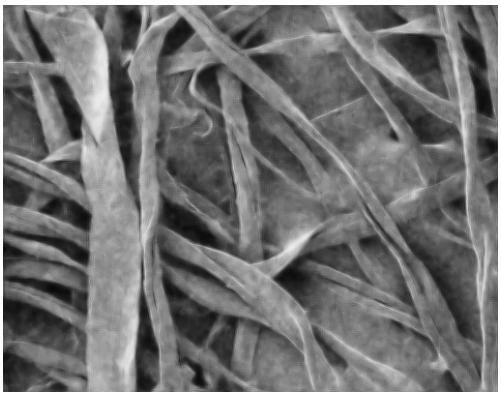
(A) Original



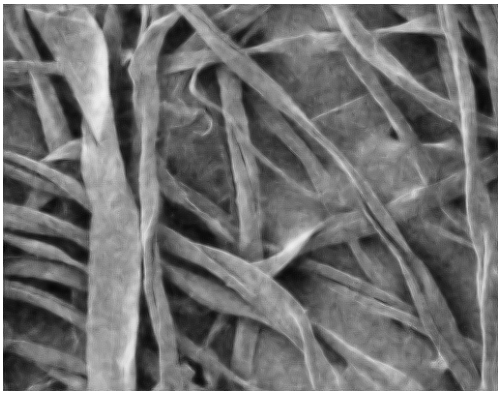
(B) With Gaussian noise



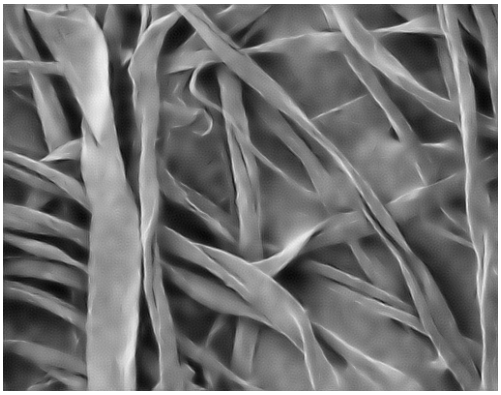
(C) Spatial Perona-Malik



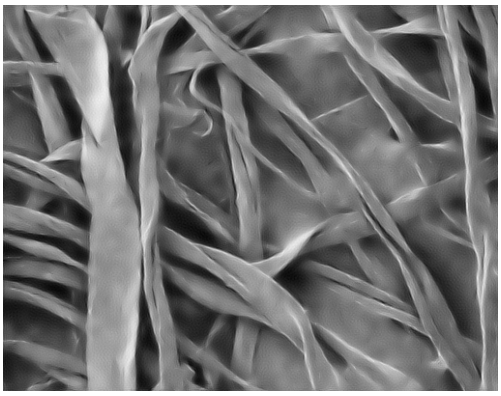
(D) Left invariant MCF



(E) Gauge MCF



(F) Left invariant TVF



(G) Gauge TVF

FIGURE 9.8: Qualitative comparison of over-smoothed collagen images starting from a noisy image with $\sigma = .2$ Gaussian noise. Images are smoothed for twice the time needed to reach their minimal L^2 error per Figure 9.7. We use isotropic processing with coherence enhancement.

tional images of different styles shown in Figure 9.13. Results are summarized in Table 9.4.

As the gauge TVF method compares favorably against correlated noise according to the PSNR value in Table 9.4, we will look at its qualitative result and compare it against BM3D in Figure 9.14.

DENOISING AND FIBER ENHANCEMENT ON FODFS. In DW-MRI image processing one obtains a field of angular diffusivity profiles (orientation density function) of water-molecules. A high diffusivity in particular orientation correlates to biological fibers structure, in brain white matter, along that same direction. Crossing-preserving enhancement of FODF fields $U : \mathbb{M}_3 \to \mathbb{R}^+$ helps to better identify structural pathways in brain white matter, which is relevant for surgery planning, see for example MEESTERS et al. (2017) and PORTEGIES, FICK, et al. (2015).

For a quantitative comparison we applied TVF, MCF and Perona-Malik diffusion (CREUSEN, DUITS, VILANOVA, and FLORACK, 2013) to denoise a popular synthetic FODF $U : \mathbb{M}_3 \to \mathbb{R}^+$ from the *Fiberfox Tractometer challenge* with realistic noise profiles (NEHER, LAUN, STIELTJES, and MAIER-HEIN, 2014). In Figure 9.15, we can

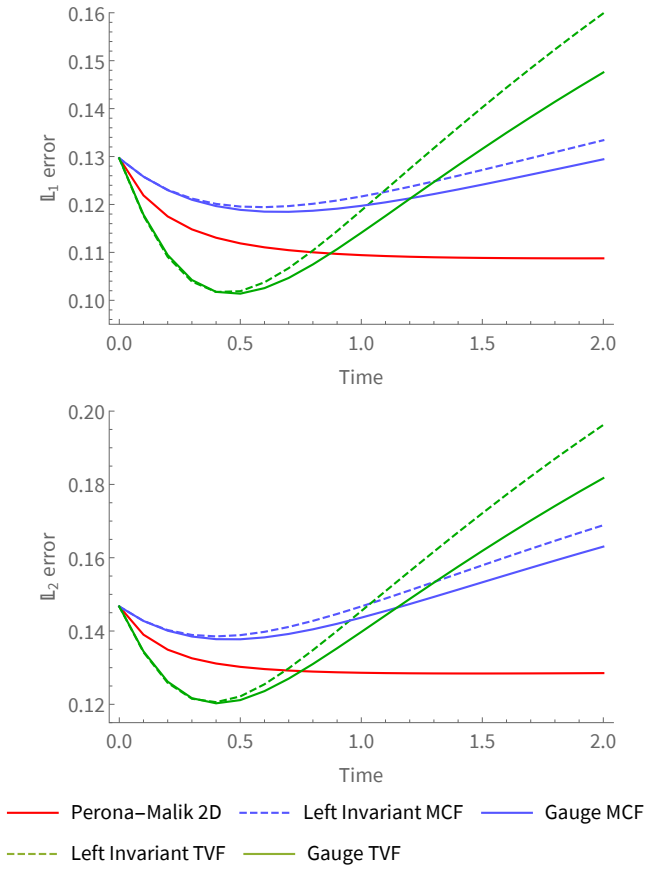


FIGURE 9.9: Isotropic ($\mathfrak{e} = 1$) evolution of relative L^1 and L^2 errors over time of the collagen image with correlated noise ($\sigma = 0.2$ Gaussian noise with $r = 2$ and $\sigma = 1$ Gaussian Filter) benchmarked against spatial Perona-Malik. Remaining parameters are $D_S = 1$, $D_A = .01$ and $\varepsilon = .001$

observe the many crossing fibers in the dataset. Furthermore, we depicted the absolute L^2-error $t \mapsto \|U - \Phi_t(U)\|_{L^2(\mathbb{M}_3)}$ as a function of the evolution parameter t, where $\Phi_t(U) = W_\varepsilon(\cdot, t)$ with optimized $\varepsilon = 0.02$ in the case of TVF (in green), and MCF (in blue), and where Φ_t is the Perona-Malik diffusion evolution on $\mathbb{M}_3$ with optimized PM parameter $K = 0.2$ (in red).

We also depict results for $K = 0.1, 0.4$ (with the dashed lines) and $\varepsilon = 0.01, 0.04$. We see that the other parameter settings pro-

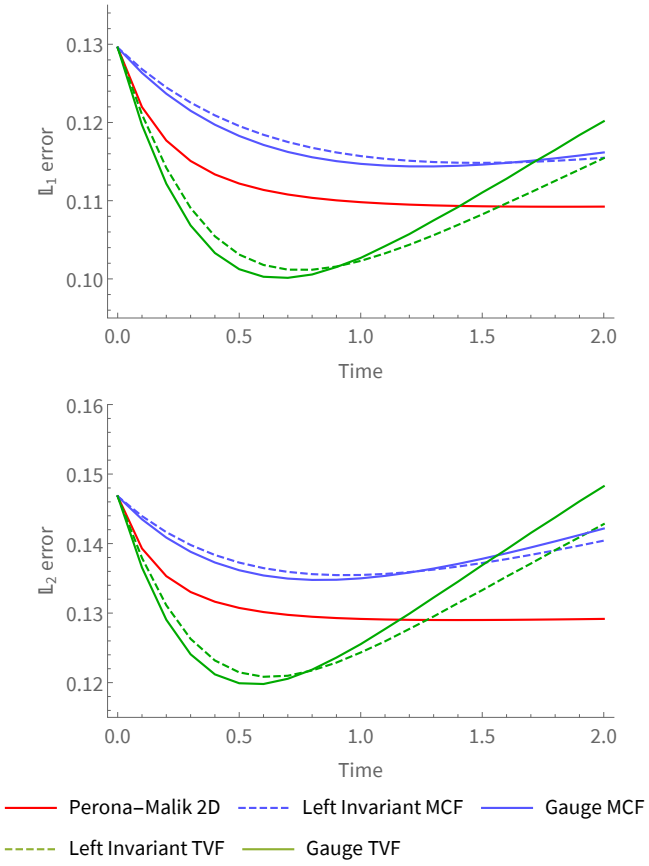
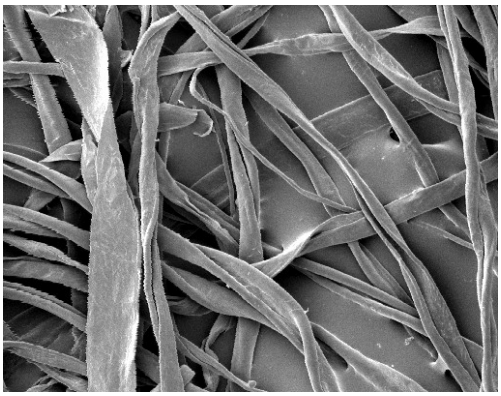


FIGURE 9.10: Isotropic with coherence enhancement evolution of relative L^1 and L^2 errors over time of the collagen image with correlated noise ($\sigma = 0.2$ Gaussian noise with $r = 2$ Gaussian Filter) benchmarked against spatial Perona-Malik. Run with parameters $D_S = 1$, $D_A = .01$, $\varepsilon = .001$ and $c = .2$

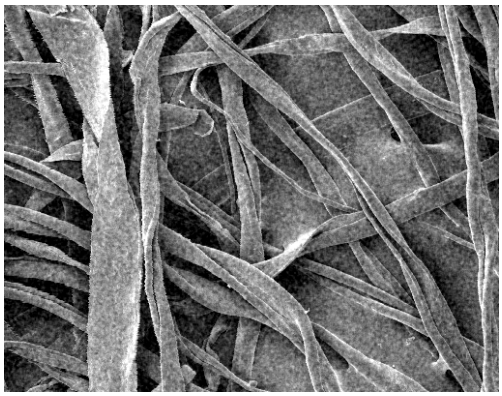
vide on average worse results, justifying our optimized parameter settings. We set $D_S = 1.0$, $D_A = 0.001$, $\Delta t = 0.01$. We observe that:

- TVF can reach lower error values than MCF with adequate $\Delta t = 0.01$,
- MCF provides more stable errors for all $t > 0$ than TVF with respect to $\epsilon > 0$,
- TVF and MCF produce lower error values than Perona-Malik diffusion,

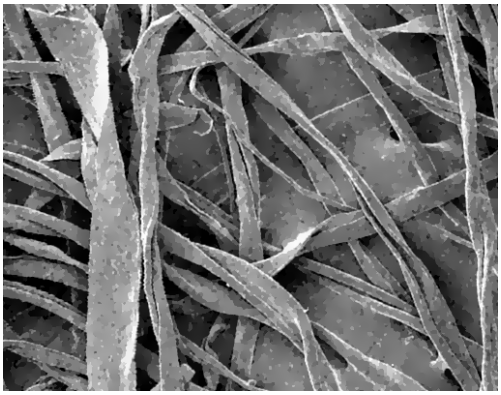
- Perona-Malik diffusion provides the most variable results for all $t > 0$.



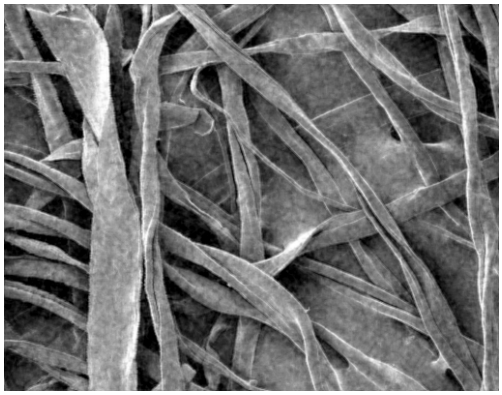
(A) Original



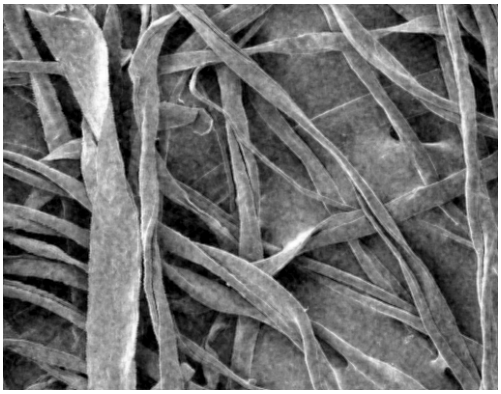
(B) With Gaussian noise



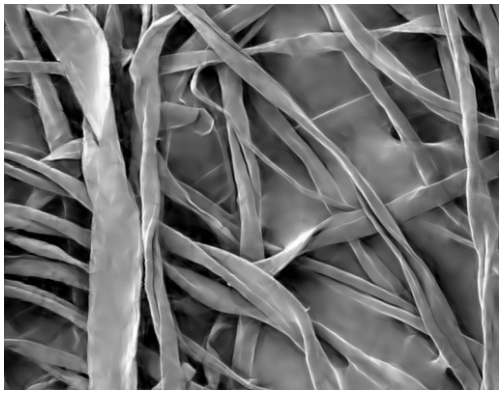
(C) Spatial Perona-Malik



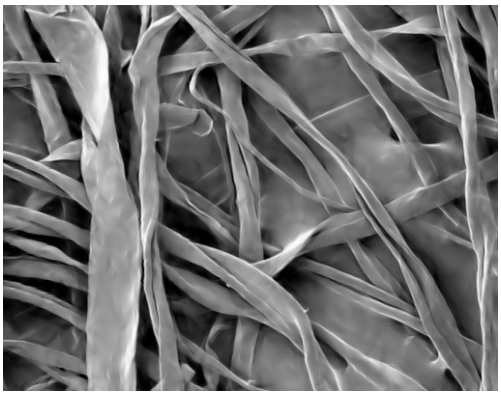
(D) Left invariant MCF



(E) Gauge MCF



(F) Left invariant TVF

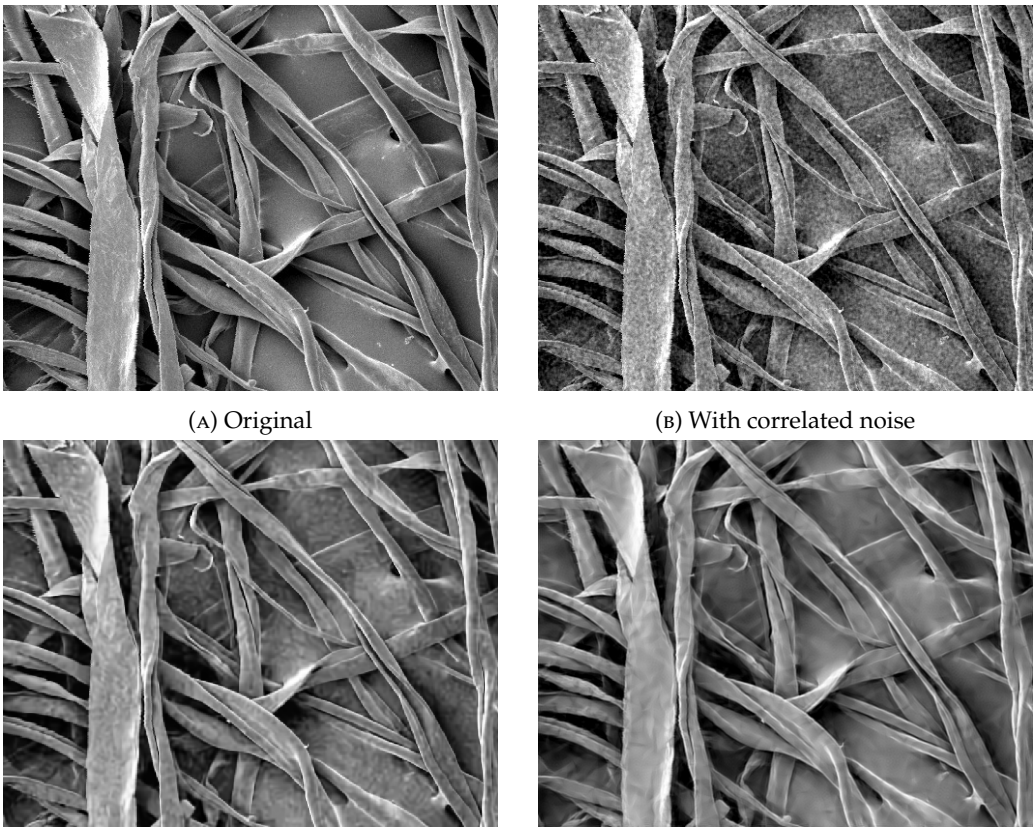


(G) Gauge TVF

FIGURE 9.11: Qualitative comparison of over-smoothed collagen images starting from a noisy image with correlated noise ($\sigma = 0.2$ Gaussian noise with $r = 2$ Gaussian Filter). Images are smoothed for twice the time needed to reach their minimal L^2 error per Figure 9.10. We use isotropic processing with coherence enhancement.

	Gaussian noise			Correlated noise		
Method	Collagen	Spiral	Mona Lisa	Collagen	Spiral	Mona Lisa
Perona-Malik	20.1	19.5	20.5	24.2	25.3	25.1
BM3D	**23.1**	**21.4**	23.9	24.0	21.4	26.3
Left inv. MCF	21.7	19.5	23.3	23.8	24.3	26.2
Gauge MCF	21.7	19.8	23.7	23.9	24.2	26.2
Left inv. TVF	22.4	20.9	26.0	24.7	25.7	26.8
Gauge TVF	23.0	21.2	**26.1**	**24.9**	**25.9**	**26.9**
Noisy image	14.1	15.5	14.1	23.9	25.3	23.9

TABLE 9.4: Comparing Peak-Signal-to-Noise-Ratio (dB) for the coherence enhanced methods against BM3D and spatial Perona-Malik at the minimal L^2 error. Three images were tested, the first being the collagen image we have been using, second a spiral test image and third a grayscale Mona Lisa. The best result for each case is indicated in **purple**, the second best result in green.



(A) Original (B) With correlated noise

(C) BM3D (D) Left invariant TVF

FIGURE 9.12: Comparison of left invariant TVF with coherence enhancement against BM3D. The BM3D image is denoised at $\sigma = 0.12$ (1.5× optimal L^2) and the left invariant TVF denoised for 1.5× optimal L^2 time. Pay particular attention to the diagonal edges where TVF exhibits superior performance.

For a qualitative comparison we applied TVF, MCF, Perona-Malik diffusion and linear diffusion to a FODF field $U : \mathbb{M}_3 \to \mathbb{R}^+$ obtained from a standard DW-MRI dataset (with $b = 1000s/mm^2$, 54 gradient directions) via *constrained spherical deconvolution* (CSD) (DESCOTEAUX, DERICHE, KNOSCHE, and ANWANDER, 2008; TOURNIER, CALAMANTE, and CONNELLY, 2007). See Figure 9.16, where for each method, we used the optimal parameter settings with the artificial dataset. We see that

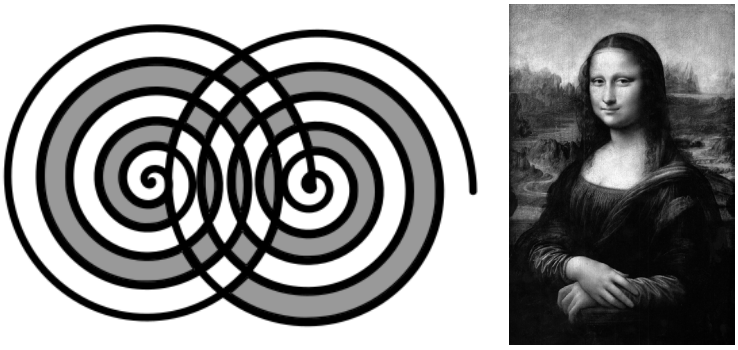


FIGURE 9.13: A spiral test image and a monochrome Mona-Lisa that were used for the PSNR experiments in Table 9.4.

(A) With correlated noise (B) BM3D (C) Gauge TVF with CE

FIGURE 9.14: Comparing Gauge TVF with coherence enhancement and BM3D against correlated noise, the standard deviation for BM3D and the evolution time for TVF were set at 1.5 times the number needed to reach the optimal L^2 error

- all methods perform well on the real datasets. Contextual alignment of the angular profiles better reflects the anatomical fiber bundles,
- MCF and TVF better preserve boundaries and angular sharpness,

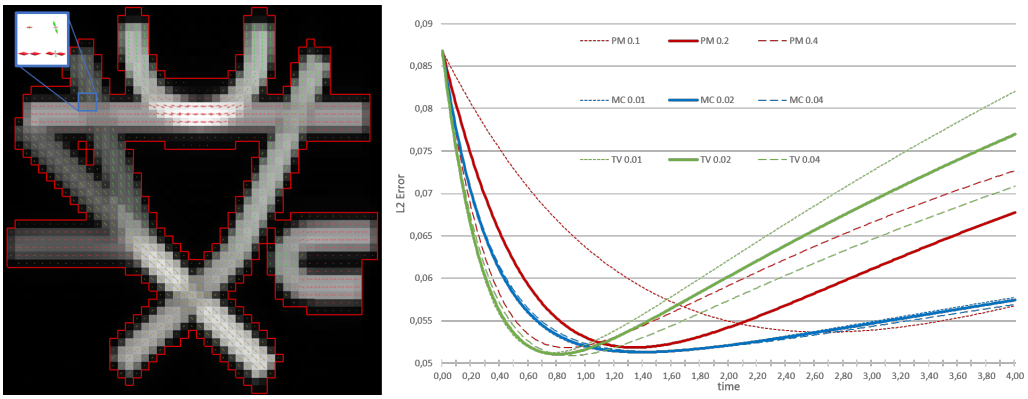


FIGURE 9.15: Quantitative comparison of denoising a fiber orientation density function (FODF) obtained by CSD (DESCOTEAUX, DERICHE, KNOSCHE, and ANWANDER, 2008; TOURNIER, CALAMANTE, and CONNELLY, 2007) from the benchmark DW-MRI dataset Fiberfox (NEHER, LAUN, STIELTJES, and MAIER-HEIN, 2014).

- MCF better preserves the amplitude at crossings at longer times.

9.11 CONCLUSION

We have proposed a PDE system on the homogeneous space $\mathbb{M}_d = \mathbb{R}^d \rtimes S^{d-1}$ of positions and orientations, for crossing-preserving denoising and enhancement of (lifted) images containing both complex elongated structures and plateaus.

It includes TVF, MCF and diffusion flows as special cases, and includes (sub-)Riemannian geometry. Thereby we generalized recent works by CHAMBOLLE and POCK (2019) and CITTI, FRANCESCHIELLO, SANGUINETTI, and SARTI (2016) from 2D to 3D using a different numerical scheme with new convergence results (Theorem 9.5) and stability bounds. We used the divergence and intrinsic gradient on a (sub)-Riemannian manifold above $\mathbb{M}_d$ for a formal weak formulation of total variation flows, which simplifies if the lifted images are differentiable (Lemma 9.1).

For 2D image denoising and enhancement we have shown that in all cases TVF on $\mathbb{M}_2$ has a better minimal error than Perona-Malik and MCF at the cost of being more sensitive to oversmooth-

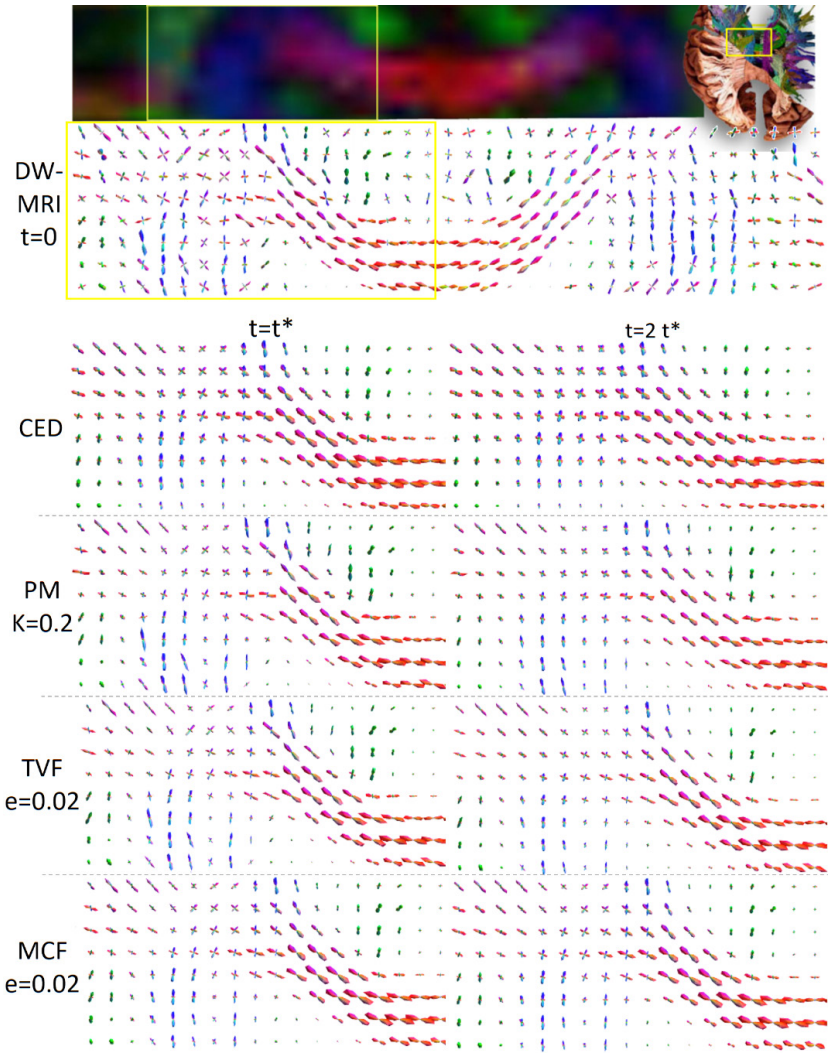


FIGURE 9.16: Qualitative comparison of denoising a FODF obtained by CSD (DESCOTEAUX, DERICHE, KNOSCHE, and ANWANDER, 2008; TOURNIER, CALAMANTE, and CONNELLY, 2007) from a standard DW-MRI dataset (with $b = 1000s/mm^2$ and 54 gradient directions). For the CSD we used up to 8^{th} order spherical harmonics, and the FODF is then spherically sampled on a tessellation of the isocahedron with 162 orientations.

ing, recall Figures 9.5 to 9.11. The L^1, L^2 and PSNR measures indicate the potential of our proposed methods for denoising and

we manage to improve PSNR results against methods such as BM3D against correlated noise on some images, recall Figure 9.12 & 9.14. Qualitatively this is mainly reflected in better clearing of plateaus while still preserving hard edges and crossings.

In 3D we compared to previous nonlinear crossing-preserving diffusion methods on $\mathbb{M}_3$, we showed improvements over Perona-Malik and improvements over contextual fiber enhancement methods in DW-MRI processing (Creusen, Duits, Vilanova, and Florack, 2013; Duits, Dela Haije, Creusen, and Ghosh, 2013) on real medical image data. We observe that crossings and boundaries (of bundles and plateaus) are better preserved over time. We support this quantitatively by a denoising experiment on a benchmark DW-MRI dataset, where MCF performs better than TVF and both perform better than Perona-Malik diffusions, in view of error reduction and stability.

Altogether, we conclude that our TVF and MCF methods on $\mathbb{M}_d$ work well for denoising and enhancement for both $d = 2$ and $d = 3$. In general we see clear benefits of the inclusion of locally adaptive frames and of limited inclusion of coherence enhancement. The code from our experiments is available as a Mathematica notebook at www.bmnsmets.com/files/tvf_mcf_denoising_jmiv.nb.

FUTURE WORK. While we have shown the potential of our PDE system on $\mathbb{M}_d$ as a denoising/enhancement method some challenges remain for future work:

- Determining stopping time; our methods show good minimal errors but are prone to degrading the image if left running for too long. For general applications a robust automatic stopping method would be helpful. Spectral analysis of nonlinear operators (Bungert, Burger, and Tenbrinck, 2019; Cohen, Falik, and Gilboa, 2019) may apply here.
- Coherence enhancement (Weickert, 1999) was not originally conceived for denoising. It is therefore interesting to see how *edge enhancing diffusion* (EED) (Fabbrini, Greco, Messina, and Pinelli, 2013) performs when generalized to $\mathbb{M}_d$, i.e. we would reformulate our enhancement operator E as:

$$E := \int_{S^d} c \otimes c \; e^{\frac{|\nabla U \cdot c|^2}{2\kappa^2}} \, \mathrm{d}\mu(c),$$

and test its performance.

- In this article we obtained convergence results of our PDE solutions for $\epsilon \downarrow 0$ while keeping $\mathfrak{e} > 0$ fixed. It is interesting to study the full limiting case $(\epsilon, \mathfrak{e}) \to (0, 0)$, for the general setting covering total variation flow.

Part IV

APPENDIX

A

LEFT-INVARIANT VECTOR FIELD BASIS FOR SE(3)

Two charts are required to cover SO(3), when using *ZYZ-Euler angles*:

$$g = \left(x,\ y,\ z,\ R_{e_z,\gamma} R_{e_y,\beta} R_{e_z,\alpha}\right), \tag{A.1}$$

where $\beta \in (0, \pi)$ and $\alpha, \gamma \in [0, 2\pi)$, the left invariant vector field basis is given by:

$$\begin{aligned}
(\mathcal{A}_1)_g &= (\cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma)\ (\partial_x)_g \\
&\quad + (\sin\alpha\cos\gamma + \cos\alpha\cos\beta\sin\gamma)\ \left(\partial_y\right)_g \\
&\quad - \cos\alpha\sin\beta\ (\partial_z)_g\,, \\
(\mathcal{A}_2)_g &= (-\sin\alpha\cos\beta\cos\gamma - \cos\alpha\sin\gamma)\ (\partial_x)_g \\
&\quad (\cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma)\ \left(\partial_y\right)_g \\
&\quad + \sin\alpha\sin\beta\ (\partial_z)_g\,, \\
(\mathcal{A}_3)_g &= \sin\beta\cos\gamma\ (\partial_x)_g + \sin\beta\sin\gamma\ \left(\partial_y\right)_g + \cos\beta\ (\partial_z)_g\,,
\end{aligned}$$

$$\begin{aligned}
-(\mathcal{A}_5)_g &= \cos\alpha\cot\beta\ (\partial_\alpha)_g + \sin\alpha\ \left(\partial_\beta\right)_g - \frac{\cos\alpha}{\sin\beta}\ \left(\partial_\gamma\right)_g\,, \\
(\mathcal{A}_4)_g &= -\sin\alpha\cot\beta\ (\partial_\alpha)_g + \cos\alpha\ \left(\partial_\beta\right)_g + \frac{\sin\alpha}{\sin\beta}\ \left(\partial_\gamma\right)_g\,, \\
(\mathcal{A}_6)_g &= (\partial_\alpha)_g\,.
\end{aligned}$$

Recall that in this chapter we take the convention that A_{d+i} is the in planar rotation from axis $a = (0, 0, 1)$ to A_i for $i = 1, \ldots d-1$. Thereby $-\mathcal{A}_5$ is the counter clockwise rotation about axis $\mathcal{A}_1$, whereas $\mathcal{A}_4$ is the counter-clockwise rotation about axis $\mathcal{A}_2$. Recall Fig. 9.3.

The above set of expressions are not valid for $\beta = 0$ or $\beta = \pi$, in that case we can switch to XYZ-Euler angles:

$$g = \left(x,\, y,\, z,\, R_{e_x,\gamma'} R_{e_y,\beta'} R_{e_z,\alpha'}\right), \tag{A.2}$$

with $\alpha' \in [0, 2\pi)$, $\beta' \in [-\pi, \pi)$ and $\gamma' \in (-\pi/2, \pi/2)$. The basis vector fields are then given by:

$$\begin{aligned}
(\mathcal{A}_1)_g &= \cos\alpha' \cos\beta' \; (\partial_x)_g \\
&\quad + (\cos\gamma' \sin\alpha' + \cos\alpha' \sin\beta' \sin\gamma') \; \left(\partial_y\right)_g \\
&\quad + (\sin\alpha' \sin\gamma' - \cos\alpha' \sin\beta' \cos\gamma') \; (\partial_z)_g \,, \\
(\mathcal{A}_2)_g &= -\sin\alpha' \cos\beta' \; (\partial_x)_g \\
&\quad + (\cos\alpha' \cos\gamma' - \sin\alpha' \sin\beta' \sin\gamma') \; \left(\partial_y\right)_g \\
&\quad + (\sin\alpha' \sin\beta' \cos\gamma' + \cos\alpha' \sin\gamma') \; (\partial_z)_g \,, \\
(\mathcal{A}_3)_g &= \sin\beta' \, \partial_x - \cos\beta' \sin\gamma' \, \partial_y + \cos\beta' \cos\gamma' \; (\partial_z)_g \,,
\end{aligned}$$

$$\begin{aligned}
-(\mathcal{A}_5)_g &= -\cos\alpha' \tan\beta' \; (\partial_{\alpha'})_g + \sin\alpha' \; \left(\partial_{\beta'}\right)_g + \frac{\cos\alpha'}{\cos\beta'} \; \left(\partial_{\gamma'}\right)_g \,, \\
(\mathcal{A}_4)_g &= \sin\alpha' \tan\beta' \; (\partial_{\alpha'})_g + \cos\alpha' \; \left(\partial_{\beta'}\right)_g - \frac{\sin\alpha'}{\cos\beta'} \; \left(\partial_{\gamma'}\right)_g \,, \\
(\mathcal{A}_6)_g &= (\partial_{\alpha'})_g \,,
\end{aligned}$$

for $|\beta'| \neq \pi/2$.

BIBLIOGRAPHY

Akian, Marianne, Jean-Pierre Quadrat, and Michel Viot (1994). "Bellman processes." In: 11th International Conference on Analysis and Optimization of Systems Discrete Event Systems: Sophia-Antipolis, June 15–16–17, 1994, Springer, pp. 302–311 (cit. on p. 96).

Ali, Syed Twareque, Jean-Pierre Antoine, Jean-Pierre Gazeau, et al. (2000). Coherent states, wavelets and their generalizations, vol. 3. Springer (cit. on p. 95).

Alt, Tobias, Pascal Peter, Joachim Weickert, and Karl Schrader (2021). "Translating numerical concepts for PDEs into neural architectures." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 294–306 (cit. on p. 97).

Ambrosio, Luigi, Roberta Ghezzi, and Valentino Magnani (2015). "BV functions and sets of finite perimeter in sub-Riemannian manifolds." In: Annales de l'Institut Henri Poincaré C, 32.3, pp. 489–517 (cit. on p. 202).

Ambrosio, Luigi, Nicola Gigli, and Giuseppe Savaré (2005). Gradient flows: in metric spaces and in the space of probability measures, Springer Science & Business Media (cit. on pp. 205, 208, 210–213).

Anderson, Edgar (1936). "The species problem in Iris." In: Annals of the Missouri Botanical Garden, 23.3, pp. 457–509 (cit. on p. 170).

Apicella, Andrea, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete (2021). "A survey on modern trainable activation functions." In: Neural Networks, 138, pp. 14–32 (cit. on pp. 154–156).

Aquinas, St. Thomas (1269). "Summa Contra Gentiles, Book III, c. 25." In: trans. by James F. Anderson. University of Notre Dame Press, pp. 97–103 (cit. on p. 271).

Azagra, Daniel, Juan Ferrera, and Fernando López-Mesas (2005). "Nonsmooth analysis and Hamilton–Jacobi equations on Riemannian manifolds." In: Journal of Functional Analysis, 220.2, pp. 304–361 (cit. on p. 130).

Balogh, Zoltán M, Alexandre Engulatov, Lars Hunziker, and Outi Elina Maasalo (2012). "Functional inequalities and Hamilton–Jacobi equations in geodesic spaces." In: Potential analysis, 36, pp. 317–337 (cit. on pp. 128–130).

Barbieri, Davide, Giovanna Citti, Giacomo Cocci, and Alessandro Sarti (2014). "A cortical-inspired geometry for contour perception and motion integration." In: Journal of mathematical imaging and vision, 49, pp. 511–529 (cit. on p. 95).

Baspinar, Emre (2018). "Minimal surfaces in sub-Riemannian structures and functional geometry of the visual cortex." PhD thesis. Università di Bologna. doi: 10.6092/unibo/amsdottorato/8661 (cit. on pp. 95, 205).

Baspinar, Emre, Giovanna Citti, and Alessandro Sarti (2018). "A geometric model of multi-scale orientation preference maps via Gabor functions." In: Journal of Mathematical Imaging and Vision, 60, pp. 900–912 (cit. on p. 184).

Baweja, Chaitanya (2018). RotNIST, https://github.com/ChaitanyaBaweja/RotNIST (cit. on pp. 144, 147).

Bekkers, Erik J, Remco Duits, Alexey Mashtakov, and Gonzalo R Sanguinetti (2015). "A PDE approach to data-driven sub-Riemannian geodesics in SE (2)." In: SIAM Journal on Imaging Sciences, 8.4, pp. 2740–2770 (cit. on pp. 112, 130, 182).

Bekkers, Erik J. (2017). "Retinal image analysis using sub-Riemannian geometry in SE(2)." PhD thesis. Eindhoven University of Technology (cit. on pp. 66, 182, 183, 186).

Bekkers, Erik J. (2019). "B-spline CNNs on Lie groups." In: arXiv preprint arXiv:1909.12057, URL: https://arxiv.org/abs/1909.12057 (cit. on pp. 53, 73).

Bekkers, Erik J., Maxime W. Lafarge, Mitko Veta, Koen A.J. Eppenhof, Josien P.W. Pluim, and Remco Duits (2018). "Roto-translation covariant convolutional networks for medical image analysis." In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I, Springer, pp. 440–448 (cit. on pp. 73, 81).

Bellaard, Gijs, Daan L.J. Bon, Gautam Pai, Bart M.N. Smets, and Remco Duits (2023). "Analysis of (sub-) Riemannian PDE-G-CNNs." In: Journal of Mathematical Imaging and Vision, pp. 1–25. DOI: 10.1007/s10851-023-01147-w (cit. on pp. 3, 4, 93, 98, 115).

Bellaard, Gijs, Gautam Pai, Javier Olivan Bescos, and Remco Duits (2023). "Geometric adaptations of PDE-G-CNNs." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 538–550 (cit. on pp. 87, 93).

Bellaard, Gijs, Sei Sakata, Bart M. N. Smets, and Remco Duits (2024). "PDE-CNNs: Axiomatic Derivations and Applications." In: arXiv: 2403.15182 [cs.LG] (cit. on p. 5).

Bengio, Yoshua, Ian Goodfellow, and Aaron Courville (2017). Deep learning, vol. 1. MIT press Cambridge, MA, USA (cit. on p. 75).

VAN DEN BERG, Nicky J., Bart M.N. SMETS, Gautam PAI, Jean-Marie MIREBEAU, and Remco DUITS (2024). "Geodesic tracking via new data-driven connections of Cartan type for vascular tree tracking." In: Journal of Mathematical Imaging and Vision, 66, pp. 198–230. DOI: 10.1007/s10851-023-01170-x (cit. on p. 5).

VAN DEN BERG, Nicky J., Shuhe ZHANG, Bart M.N. SMETS, Tos T.J.M. BERENDSCHOT, and Remco DUITS (2023). "Geodesic tracking of retinal vascular trees with optical and TV-flow enhancement in SE(2)." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 525–537. DOI: 10.1007/978-3-031-31975-4_40 (cit. on p. 5).

BERTALMÍO, Marcelo, Luca CALATRONI, Valentina FRANCESCHI, Benedetta FRANCESCHIELLO, and Dario PRANDI (2019). "A cortical-inspired model for orientation-dependent contrast perception: a link with Wilson-Cowan equations." In: Scale Space and Variational Methods in Computer Vision: 7th International Conference, SSVM 2019, Hofgeismar, Germany, June 30–July 4, 2019, Proceedings 7, Springer, pp. 472–484 (cit. on p. 182).

BOLELLI, Maria V. (2023). "Neurogeometry of stereo vision." PhD thesis. Università di Bologna. DOI: 10.48676/unibo/amsdottorato/10714 (cit. on p. 95).

VAN DEN BOOMGAARD, Rein and Arnold SMEULDERS (1994). "The morphological structure of images: The differential equations of morphological scale-space." In: IEEE transactions on pattern analysis and machine intelligence, 16.11, pp. 1101–1113 (cit. on pp. 95, 123).

BOSCAIN, Ugo V., Roman CHERTOVSKIH, Jean-Paul GAUTHIER, Dario PRANDI, and Alexey REMIZOV (2018). "Highly corrupted image inpainting through hypoelliptic diffusion." In: Journal of Mathematical Imaging and Vision, 60, pp. 1231–1245 (cit. on p. 182).

Breuß, Michael and Joachim Weickert (2009). "Highly accurate PDE-based morphology for general structuring elements." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 758–769 (cit. on p. 95).

Brézis, H (1973). Operateurs maximeaux monotones et semi-gropes de contractions dans les espaces de Hilbert, (cit. on p. 205).

Bronstein, Michael M, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst (2017). "Geometric deep learning: going beyond euclidean data." In: IEEE Signal Processing Magazine, 34.4, pp. 18–42 (cit. on p. 2).

Bungert, Leon, Martin Burger, and Daniel Tenbrinck (2019). "Computing nonlinear eigenfunctions via gradient flow extinction." In: Scale Space and Variational Methods in Computer Vision: 7th International Conference, SSVM 2019, Hofgeismar, Germany, June 30–July 4, 2019, Proceedings 7, Springer, pp. 291–302 (cit. on p. 230).

Burger, Martin, Alexander Sawatzky, and Gabriele Steidl (2016). First order algorithms in variational image processing, Springer (cit. on p. 97).

Cajori, Florian (1928). "The early history of partial differential equations and of partial differentiation and integration." In: The American Mathematical Monthly, 35.9, pp. 459–467. doi: 10.1080/00029890.1928.11986877 (cit. on p. 1).

Chambolle, Antonin and Thomas Pock (2019). "Total roto-translational variation." In: Numerische Mathematik, 142, pp. 611–666 (cit. on pp. 181–184, 204, 228).

Chen, Tian Qi, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud (2018). "Neural ordinary differential equations." In: Advances in Neural Information Processing Systems, pp. 6571–6583 (cit. on pp. 95, 96).

Chen, Yunjin, Wei Yu, and Thomas Pock (2015). "On learning optimized reaction diffusion processes for effective image restoration." In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5261–5269 (cit. on p. 97).

Chirikjian, Gregory S. and Alexander B. Kyatkin (2000). "An operational calculus for the Euclidean motion group with applications in robotics and polymer science." In: Journal of Fourier Analysis and Applications, 6, pp. 583–606 (cit. on p. 74).

Citti, Giovanna, Benedetta Franceschiello, Gonzalo Sanguinetti, and Alessandro Sarti (2016). "Sub-Riemannian mean curvature flow for image processing." In: SIAM Journal on Imaging Sciences, 9.1, pp. 212–237 (cit. on pp. 3, 181, 182, 184, 204, 205, 228).

Citti, Giovanna and Alessandro Sarti (2006). "A cortical based model of perceptual completion in the roto-translation space." In: Journal of Mathematical Imaging and Vision, 24, pp. 307–326 (cit. on pp. 95, 182).

Citti, Giovanna and Alessandro Sarti, eds. (2014). Neuromathematics of Vision, Lecture Notes in Morphogenesis. Berlin, Heidelberg: Springer. ISBN: 978-3-642-34443-5. DOI: 10.1007/978-3-642-34444-2 (cit. on p. 2).

Cohen, E., T. Deffieux, C. Demené, L.D. Cohen, and M. Tanter (2018). "3D vessel extraction in the rat brain from ultrasensitive Doppler images." In: Computer Methods in Biomechanics and Biomedical Engineering: Proceedings of the 14th International Symposium CMBBE, Tel Aviv, Israel, 2016, Springer, pp. 81–91 (cit. on p. 184).

Cohen, Ido, Adi Falik, and Guy Gilboa (2019). "Stable explicit p-Laplacian flows based on nonlinear eigenvalue analysis." In: Scale Space and Variational Methods in Computer Vision: 7th International Conference, SSVM 2019, Hofgeismar, Germany, June 30–July 4, 2019, Proceedings 7, Springer, pp. 315–327 (cit. on p. 230).

Cohen, Taco (2021). "Equivariant convolutional networks." PhD thesis. University of Amsterdam (cit. on pp. 49, 53, 73, 88, 89).

Cohen, Taco, Mario Geiger, and Maurice Weiler (2020). A General Theory of Equivariant CNNs on Homogeneous Spaces, arXiv: 1811.02017 [cs.LG] (cit. on p. 73).

Cohen, Taco and Max Welling (2016). "Group equivariant convolutional networks." In: International conference on machine learning, PMLR, pp. 2990–2999 (cit. on pp. 73, 89).

Creusen, Eric, Remco Duits, Anna Vilanova, and Luc Florack (2013). "Numerical schemes for linear and non-linear enhancement of DW-MRI." In: Numerical Mathematics: Theory, Methods and Applications, 6.1, pp. 138–168 (cit. on pp. 183, 185, 206, 214, 221, 230).

Dabov, Kostadin, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian (2007). "Image denoising by sparse 3-D transform-domain collaborative filtering." In: IEEE Transactions on image processing, 16.8, pp. 2080–2095 (cit. on pp. 181, 214, 219).

Descoteaux, Maxime, Rachid Deriche, Thomas R Knosche, and Alfred Anwander (2008). "Deterministic and probabilistic tractography based on complex fibre orientation distributions." In: IEEE transactions on medical imaging, 28.2, pp. 269–286 (cit. on pp. 226, 228, 229).

Dhilleswararao, Pudi, Srinivas Boppu, M Sabarimalai Manikandan, and Linga Reddy Cenkeramaddi (2022). "Efficient hardware architectures for accelerating deep neural networks: Survey." In: IEEE access, 10, pp. 131788–131828 (cit. on p. 178).

Dieleman, Sander, Jeffrey De Fauw, and Koray Kavukcuoglu (2016). "Exploiting cyclic symmetry in convolutional neural networks." In: International conference on machine learning, PMLR, pp. 1889–1898 (cit. on p. 73).

Dieleman, Sander, Kyle W Willett, and Joni Dambre (2015). "Rotation-invariant convolutional neural networks for galaxy morphology prediction." In: Monthly notices of the royal astronomical society, 450.2, pp. 1441–1459 (cit. on p. 73).

Dragoni, Federica (2007). "Metric Hopf-Lax formula with semi-continuous data." In: Discrete and Continuous Dynamical Systems, 17.4, p. 713 (cit. on pp. 130, 131).

Dubrovina-Karni, Anastasia, Guy Rosman, and Ron Kimmel (2014). "Multi-region active contours with a single level set function." In: IEEE transactions on pattern analysis and machine intelligence, 37.8, pp. 1585–1601 (cit. on p. 97).

Duits, Remco (2005). "Perceptual organization in image analysis: a mathematical approach based on scale, orientation and curvature." PhD thesis. Eindhoven University of Technology. DOI: 10.6100/IR595308 (cit. on pp. 73, 95, 182–184, 186, 187).

Duits, Remco, Erik J. Bekkers, and Alexey Mashtakov (2019). "Fourier transform on the homogeneous space of 3D positions and orientations for exact solutions to linear PDEs." In: Entropy, 21.1, p. 38 (cit. on pp. 74, 96).

Duits, Remco and Bernhard Burgeth (2007). "Scale spaces on Lie groups." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 300–312 (cit. on pp. 3, 73, 97).

Duits, Remco, Tom Dela Haije, Eric Creusen, and Arpan Ghosh (2013). "Morphological and linear scale spaces for fiber enhancement in DW-MRI." In: Journal of mathematical imaging and vision, 46, pp. 326–368 (cit. on pp. 183, 230).

Duits, Remco, Maurice Duits, Markus van Almsick, and Bart ter Haar Romeny (2007). "Invertible orientation scores as an application of generalized wavelet theory." In: Pattern Recognition and Image Analysis, 17.1, pp. 42–75 (cit. on p. 4).

Duits, Remco, Maurice Duits, Markus A. van Almsick, and Luc M.J. Florack (2004). "A new reconstruction from orientation bundle functions as an application of generalized wavelet theory." In: Early Cognitive Vision Workshop 2004, (cit. on p. 95).

Duits, Remco and Erik M. Franken (2010). "Left invariant parabolic evolution equations on SE (2) and contour enhancement via invertible orientation scores, part I: linear left-invariant diffusion equations on SE (2)." In: Appl. Math. AMS, 68, pp. 255–292. doi: 10.1090/S0033-569X-10-01172-0 (cit. on p. 184).

Duits, Remco, Hartmut Führ, Bart Janssen, Mark Bruurmijn, Luc Florack, and Hans van Assen (2013). "Evolution equations on Gabor transforms and their applications." In: Applied and Computational Harmonic Analysis, 35.3, pp. 483–526 (cit. on p. 95).

Duits, Remco, Michiel H.J. Janssen, Julius Hannink, and Gonzalo R. Sanguinetti (2016). "Locally adaptive frames in the roto-translation group and their applications in medical imaging." In: Journal of Mathematical Imaging and Vision, 56, pp. 367–402 (cit. on pp. 183, 192, 193, 196, 197).

Duits, Remco, Stephan P.L. Meesters, J.-M. Mirebeau, and Jorg M. Portegies (2018). "Optimal paths for variants of the 2D and 3D Reeds–Shepp car with applications in image analysis." In: Journal of Mathematical Imaging and Vision, 60, pp. 816–848 (cit. on pp. 182, 183, 205, 214).

Duits, Remco, Etienne St-Onge, Jim Portegies, and Bart M.N. Smets (2019). "Total variation and mean curvature PDEs on the space of positions and orientations." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 211–223. doi: 10.1007/978-3-030-22368-7_17 (cit. on pp. 6, 181).

Duits, Remco, Bart M.N. Smets, Erik J. Bekkers, and Jim Portegies (2021). "Equivariant deep learning via morphological and linear scale space PDEs on the space of positions and orientations." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 27–39. doi: 10.1007/978-3-030-75549-2_3 (cit. on p. 5).

Duits, Remco, Bart M.N. Smets, A.J. Wemmenhove, Jim Portegies, and Erik J. Bekkers (2021). "Recent geometric flows in multi-orientation image processing via a Cartan connection." In: Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision, pp. 1–60. doi: 10.1007/978-3-030-03009-4_101-1 (cit. on p. 4).

Duits, Remco and Markus Van Almsick (2008). "The explicit solutions of linear left-invariant second order stochastic evolution equations on the 2D Euclidean motion group." In: Quarterly of Applied Mathematics, 66.1, pp. 27–67 (cit. on p. 116).

Duran, Joan, Michael Moeller, Catalina Sbert, and Daniel Cremers (2016). "Collaborative total variation: A general framework for vectorial TV models." In: SIAM Journal on Imaging Sciences, 9.1, pp. 116–151 (cit. on p. 97).

Egmont-Petersen, Michael, Dick de Ridder, and Heinz Handels (2002). "Image processing with neural networks—a review." In: Pattern recognition, 35.10, pp. 2279–2301 (cit. on p. 2).

Elfwing, Stefan, Eiji Uchibe, and Kenji Doya (2018). "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning." In: Neural Networks, 107, pp. 3–11 (cit. on p. 155).

ter Elst, A.F.M. and Derek W. Robinson (1998). "Weighted subcoercive operators on Lie groups." In: Journal of Functional Analysis, 157.1, pp. 88–163 (cit. on p. 111).

Esteves, Carlos, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis (2018). "Learning SO(3) equivariant representations with spherical cnns." In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 52–68 (cit. on p. 73).

Evans, Lawrence C and Joel Spruck (1991). "Motion of level sets by mean curvature. I." In: Fundamental Contributions to the Continuum Theory of Evolving Phase Interfaces in Solids: A Collection of Reprints of 14 Seminal Papers, Springer, pp. 328–374 (cit. on p. 205).

Evans, Lawrence C. (2022). Partial differential equations, vol. 19. American Mathematical Society. ISBN: 978-1-4704-6942-9 (cit. on pp. 94, 98, 102, 130).

Fabbrini, Luca, Mario Greco, Marco Messina, and Gianpaolo Pinelli (2013). "Improved edge enhancing diffusion filter for speckle-corrupted images." In: IEEE Geoscience and Remote Sensing Letters, 11.1, pp. 99–103 (cit. on p. 230).

Federer, Herbert (2014). Geometric measure theory, Springer. ISBN: 978-3-540-60656-7. DOI: 10.1007/978-3-642-62010-2 (cit. on pp. 49, 52).

Felsberg, Michael, P-E Forssén, and Hanno Scharr (2005). "Channel smoothing: Efficient robust smoothing of low-level signal features." In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 28.2, pp. 209–222 (cit. on p. 184).

Finzi, Marc, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson (2020). "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data." In: International Conference on Machine Learning, PMLR, pp. 3165–3176 (cit. on p. 73).

Finzi, Marc Anton, Roberto Bondesan, and Max Welling (2020). "Probabilistic Numeric Convolutional Neural Networks." In: International Conference on Learning Representations, (cit. on p. 96).

Franken, Erik, Markus van Almsick, Peter Rongen, Luc Florack, and Bart ter Haar Romeny (2006). "An efficient method for tensor voting using steerable filters." In: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV 9, Springer, pp. 228–240 (cit. on p. 74).

Franken, Erik and Remco Duits (2009). "Crossing-preserving coherence-enhancing diffusion on invertible orientation scores." In: International Journal of Computer Vision, 85, pp. 253–278 (cit. on pp. 3, 182, 183, 185, 199, 206, 214).

Franken, Erik M. (2008). "Enhancement of crossing elongated structures in images." PhD thesis. Eindhoven University of Technology (cit. on p. 185).

Gaveau, Bernard (1977). "Principe de moindre action, propagation de la chaleur et estimees sous elliptiques sur certains groupes nilpotents." In: Acta Mathematica, 139, pp. 95–153 (cit. on p. 116).

Ghimpeţeanu, Gabriela, Thomas Batard, Marcelo Bertalmío, and Stacey Levine (2015). "A decomposition framework for image denoising algorithms." In: IEEE transactions on Image Processing, 25.1, pp. 388–399 (cit. on p. 182).

Giga, Yoshikazu and Moto-Hiko Sato (1991). "Generalized interface evolution with the Neumann boundary condition." In: (cit. on p. 205).

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks." In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, pp. 249–256 (cit. on pp. 164, 165, 167).

Goodfellow, Ian, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio (2013). "Maxout networks." In: International conference on machine learning, PMLR, pp. 1319–1327 (cit. on p. 155).

Grigor'yan, Alexander (2009). Heat kernel and analysis on manifolds, vol. 47. American Mathematical Society (cit. on pp. 115, 117).

Grigor'yan, Alexander (2006). "Heat kernels on weighted manifolds and applications." In: Cont. Math, 398.2006, pp. 93–191 (cit. on p. 117).

Grigor'Yan, Alexander, Jiaxin Hu, and Ka-Sing Lau (2009). "Heat kernels on metric spaces with doubling measure." In: Fractal geometry and stochastics IV, Springer, pp. 3–44 (cit. on pp. 117, 120).

Grigor'yan, Alexander, Jiaxin Hu, and Ka-Sing Lau (2014). "Heat kernels on metric measure spaces." In: Geometry and Analysis of Fractals: Hong Kong, December 2012, Springer, pp. 147–207 (cit. on pp. 98, 115).

Haber, Eldad and Lars Ruthotto (2017). "Stable architectures for deep neural networks." In: Inverse problems, 34.1, p. 014004 (cit. on p. 97).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." In: Proceedings of the IEEE international conference on computer vision, pp. 1026–1034 (cit. on pp. 164, 165).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition." In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (cit. on pp. 95, 96).

Janssen, Michiel H.J., Tom C.J. Dela Haije, Frank C. Martin, Erik J. Bekkers, and Remco Duits (2017). "The Hessian of axially symmetric functions on SE (3) and application in 3D image analysis." In: Scale Space and Variational Methods in Computer Vision: 6th International Conference, SSVM 2017, Kolding, Denmark, June 4-8, 2017, Proceedings 6, Springer, pp. 643–655 (cit. on p. 185).

Janssen, Michiel H.J., Augustus J.E.M. Janssen, Erik J. Bekkers, J Oliván Bescós, and Remco Duits (2018). "Design and processing of invertible orientation scores of 3D images." In: Journal of mathematical imaging and vision, 60, pp. 1427–1458 (cit. on pp. 183, 184, 186, 188).

Jenner, Erik and Maurice Weiler (2021). "Steerable Partial Differential Operators for Equivariant Neural Networks." In: International Conference on Learning Representations, (cit. on p. 97).

Kidger, Patrick (2022). "On Neural Differential Equations." PhD thesis. Oxford University. arXiv: 2202.02435 [cs.LG] (cit. on p. 95).

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization." In: arXiv preprint arXiv:1412.6980, (cit. on pp. 144, 145).

Koda, Takashi (2009). "An introduction to the geometry of homogeneous spaces." In: Proceedings of the 13th International Workshop on Differential Geometry and Related Fields, Natl. Inst. Math. Sci.(NIMS), Taejon, pp. 121–144. URL: http://webbuild.knu.ac.kr/~yjsuh/proceedings/13th/%5B10%5D09Prowork_Koda.pdf (cit. on pp. 35, 40).

Kondor, Risi and Shubhendu Trivedi (2018). "On the generalization of equivariance and convolution in neural networks to the action of compact groups." In: International Conference on Machine Learning, PMLR, pp. 2747–2755 (cit. on p. 73).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "Imagenet classification with deep convolutional neural networks." In: Advances in neural information processing systems, 25 (cit. on p. 74).

Lafarge, Maxime W, Erik J. Bekkers, Josien PW Pluim, Remco Duits, and Mitko Veta (2021). "Roto-translation equivariant convolutional networks: Application to histopathology image analysis." In: Medical Image Analysis, 68, p. 101849 (cit. on pp. 73, 81).

LeCun, Yann, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel (1989). "Backpropagation applied to handwritten zip code recognition." In: Neural computation, 1.4, pp. 541–551 (cit. on pp. 74, 148).

Lee, John M. (2012). Smooth manifolds, 2nd ed. Springer. doi: 10.1007/978-1-4419-9982-5 (cit. on pp. 9, 12, 17, 20, 35, 39, 42, 110).

Lee, John M. (2018). Introduction to Riemannian manifolds, 2nd ed. Springer. doi: 10.1007/978-3-319-91755-9 (cit. on pp. 9, 44, 49).

Liu, Zhuang, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie (2022). A ConvNet for the 2020s, arXiv: 2201.03545 [cs.CV] (cit. on pp. 95, 174, 175).

Long, Zichao, Yiping Lu, Xianzhong Ma, and Bin Dong (2018). "Pde-net: Learning pdes from data." In: International conference on machine learning, PMLR, pp. 3208–3216 (cit. on p. 97).

Loshchilov, Ilya and Frank Hutter (2019). Decoupled Weight Decay Regularization, arXiv: 1711.05101 [cs.LG] (cit. on p. 172).

Lu, Yiping, Aoxiao Zhong, Quanzheng Li, and Bin Dong (2018). "Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations." In: International Conference on Machine Learning, PMLR, pp. 3276–3285 (cit. on p. 96).

Maheux, Patrick (1998). "Estimations du noyau de la chaleur sur les espaces homogenes." In: The Journal of Geometric Analysis, 8, pp. 65–96 (cit. on pp. 117, 118, 120, 128).

Manfredi, Juan J and Bianca Stroffolini (2002). "A version of the Hopf-Lax formula in the Heisenberg group." In: Communications in Partial Differential Equations, (cit. on p. 130).

Maragos, Petros (2005). "PDEs for morphological scale-spaces and eikonal applications." In: The Image and Video Processing Handbook, pp. 587–612 (cit. on pp. 1, 95).

Markidis, Stefano, Steven Wei Der Chien, Erwin Laure, Ivy Bo Peng, and Jeffrey S. Vetter (2018). "Nvidia tensor core programmability, performance & precision." In: 2018 IEEE international parallel and distributed processing symposium workshops (IPDPSW), IEEE, pp. 522–531 (cit. on p. 178).

Martin, Frank, Erik J. Bekkers, and Remco Duits (2017). LieAnalysis: a Mathematica package, url: www.lieanalysis.nl (cit. on pp. 185, 186, 206).

Masci, Jonathan, Jesús Angulo, and Jürgen Schmidhuber (2013). "A learning framework for morphological operators using counter–harmonic mean." In: Mathematical Morphology and Its Applications to Signal and Image Processing: 11th International Symposium, ISMM 2013, Uppsala, Sweden, May 27-29, 2013. Proceedings 11, Springer, pp. 329–340 (cit. on p. 156).

Meesters, Stephan, Pauly Ossenblok, Louis Wagner, Olaf Schijns, Paul Boon, Luc Florack, Anna Vilanova, and Remco Duits (2017). "Stability metrics for optic radiation tractography: towards damage prediction after resective surgery." In: Journal of neuroscience methods, 288, pp. 34–44 (cit. on p. 221).

Mellouli, Dorra, Tarek M Hamdani, Mounir Ben Ayed, and Adel M Alimi (2017). "Morph-CNN: A morphological convolutional neural network for image classification." In: Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24, Springer, pp. 110–117 (cit. on p. 156).

Montobbio, Noemi, Laurent Bonnasse-Gahot, Giovanna Citti, and Alessandro Sarti (2019). "Kercnns: biologically inspired lateral connections for classification of corrupted images." In: arXiv preprint arXiv:1910.08336, (cit. on p. 96).

Montobbio, Noemi, Alessandro Sarti, and Giovanna Citti (2020). "A metric model for the functional architecture of the visual cortex." In: SIAM Journal on Applied Mathematics, 80.2, pp. 1057–1081 (cit. on p. 96).

Morel, Jean-Michel and Sergio Solimini (2012). Variational methods in image segmentation: with seven image processing experiments, vol. 14. Springer Science & Business Media (cit. on p. 97).

Mustapha, Sami (2000). "Gaussian estimates for heat kernels on Lie groups." In: Mathematical Proceedings of the Cambridge Philosophical Society, vol. 128. 1. Cambridge University Press, pp. 45–64 (cit. on p. 117).

Naitzat, Gregory, Andrey Zhitnikov, and Lek-Heng Lim (2020). "Topology of deep neural networks." In: The Journal of Machine Learning Research, 21.1, pp. 7503–7542 (cit. on p. 170).

Neher, Peter F, Frederik B Laun, Bram Stieltjes, and Klaus H Maier-Hein (2014). "Fiberfox: facilitating the creation of realistic white matter software phantoms." In: Magnetic resonance in medicine, 72.5, pp. 1460–1470 (cit. on pp. 221, 228).

Nesterov, Yurii (2013). Introductory lectures on convex optimization: A basic course, vol. 87. Springer Science & Business Media (cit. on p. 209).

Oyallon, Edouard and Stéphane Mallat (2015). "Deep roto-translation scattering for object classification." In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2865–2873 (cit. on p. 73).

Pai, Gautam, Gijs Bellaard, Bart M.N. Smets, and Remco Duits (2023). "Functional properties of PDE-based group equivariant convolutional neural networks." In: International Conference on Geometric Science of Information, Springer, pp. 63–72. doi: 10.1007/978-3-031-38271-0_7 (cit. on pp. 3, 5, 93).

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. (2019). "Pytorch: An imperative style, high-performance deep learning library." In: Advances in neural information processing systems, 32 (cit. on pp. 4, 143).

Perona, Pietro and Jitendra Malik (1990). "Scale-space and edge detection using anisotropic diffusion." In: IEEE Transactions on pattern analysis and machine intelligence, 12.7, pp. 629–639 (cit. on pp. 181, 184).

Petitot, Jean (2017). Elements of neurogeometry, Springer. isbn: 978-3-319-65589-5 (cit. on p. 2).

Peyré, Gabriel, Mickael Péchaud, Renaud Keriven, Laurent D. Cohen, et al. (2010). "Geodesic methods in computer vision and graphics." In: Foundations and Trends® in Computer Graphics and Vision, 5.3–4, pp. 197–397 (cit. on p. 97).

Pinheiro, José C. and Douglas M. Bates (1996). "Unconstrained parametrizations for variance-covariance matrices." In: Statistics and computing, 6, pp. 289–296 (cit. on pp. 99, 140).

Podlubny, Igor (1998). Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications, elsevier (cit. on p. 102).

Portegies, Jorg, Gonzalo Sanguinetti, Stephan Meesters, and Remco Duits (2015). "New approximation of a scale space kernel on SE (3) and applications in neuroimaging." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 40–52 (cit. on p. 111).

Portegies, Jorg M. (2018). "PDEs on the Lie group SE(3) and their application in diffusion-weighted MRI." PhD thesis. Technische Universiteit Eindhoven (cit. on pp. 95, 183).

Portegies, Jorg M. and Remco Duits (2017). "New exact and numerical solutions of the (convection–) diffusion kernels on SE (3)." In: Differential Geometry and its Applications, 53, pp. 182–219 (cit. on p. 205).

Portegies, Jorg M., Rutger Henri Jacques Fick, Gonzalo R. Sanguinetti, Stephan P.L. Meesters, Gabriel Girard, and Remco Duits (2015). "Improving fiber alignment in HARDI by combining contextual PDE flow with constrained spherical deconvolution." In: PloS one, 10.10, e0138122 (cit. on pp. 184, 214, 221).

Ramachandran, Prajit, Barret Zoph, and Quoc V Le (2017). "Searching for activation functions." In: arXiv preprint arXiv:1710.05941, (cit. on p. 154).

Reisert, Marco (2008). "Group integration techniques in pattern analysis–A kernel view." PhD thesis. Albert-Ludwigs-Universität Freiburg (cit. on p. 74).

Reisert, Marco and Hans Burkhardt (2008). "Efficient tensor voting with 3d tensorial harmonics." In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE, pp. 1–7 (cit. on p. 183).

Ritter, Gerhard X and Peter Sussner (1996). "An introduction to morphological neural networks." In: Proceedings of 13th International Conference on Pattern Recognition, vol. 4. IEEE, pp. 709–717 (cit. on p. 155).

Rockafellar, R Tyrrell and Roger J-B Wets (2009). Variational analysis, vol. 317. Springer Science & Business Media (cit. on p. 209).

Rund, Hanno (1966). The Hamilton-Jacobi theory in the calculus of variations: its role in mathematics and physics, Krieger Publishing Company (cit. on p. 130).

Ruthotto, Lars and Eldad Haber (2020). "Deep neural networks motivated by partial differential equations." In: Journal of Mathematical Imaging and Vision, 62, pp. 352–364 (cit. on p. 97).

Sander, Michael, Pierre Ablin, and Gabriel Peyré (2022). "Do residual neural networks discretize neural ordinary differential equations?" In: Advances in Neural Information Processing Systems, 35, pp. 36520–36532 (cit. on p. 95).

Sapiro, Guillermo (2006). Geometric partial differential equations and image analysis, Cambridge university press. doi: 10.1017/CBO9780511626319 (cit. on pp. 1, 97, 184, 205).

Sato, Moto-Hiko (1992). "Interface evolution with Neumann boundary condition." In: Hokkaido University Preprint Series in Mathematics, 147, pp. 2–16 (cit. on p. 205).

Schmidt, Martin and Joachim Weickert (2016). "Morphological counterparts of linear shift-invariant scale-spaces." In: Journal of Mathematical Imaging and Vision, 56.2, pp. 352–366 (cit. on pp. 123, 182).

Serra, Jean (1982). Image analysis and mathematical morphology, Academic press (cit. on p. 156).

Serra, Jean and Luc Vincent (1992). "An overview of morphological filtering." In: Circuits, Systems and Signal Processing, 11, pp. 47–108 (cit. on p. 94).

Sethian, James A. et al. (1999). Level set methods and fast marching methods, vol. 98. 2. Cambridge Cambridge UP (cit. on p. 97).

Shen, Yucong, Xin Zhong, and Frank Y. Shih (2019). Deep Morphological Neural Networks, arXiv: 1909.01532 [cs.CV] (cit. on p. 156).

Shen, Zhengyang, Lingshen He, Zhouchen Lin, and Jinwen Ma (2020). "Pdo-econvs: Partial differential operator based equivariant convolutions." In: International Conference on Machine Learning, PMLR, pp. 8697–8706 (cit. on p. 97).

Shih, Frank Y, Yucong Shen, and Xin Zhong (2019). "Development of deep learning framework for mathematical morphology." In: International Journal of Pattern Recognition and Artificial Intelligence, 33.06, p. 1954024 (cit. on p. 156).

Sifre, Laurent (2014). "Rigid-motion scattering for image classification." PhD thesis. Ecole Polytechnique Paris (cit. on pp. 73, 95).

Sifre, Laurent and Stéphane Mallat (2013). "Rotation, scaling and deformation invariant scattering for texture discrimination." In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1233–1240 (cit. on p. 73).

Sifre, Laurent and Stéphane Mallat (2014). "Rigid-Motion Scattering for Texture Classification." In: arXiv: 1403.1687 [cs.CV] (cit. on pp. 73, 95).

Simmons II, Brandon (2021). "Investigating Heart Disease Datasets and Building Predictive Models." PhD thesis. Elizabeth City State University (cit. on p. 170).

Smets, Bart M.N. (2023). Mathematics of Neural Networks, Lecture notes 2MMA80 (2021-2023). arXiv: 2403.04807 [cs.LG] (cit. on pp. 5, 49).

Smets, Bart M.N., Peter D. Donker, Jim Portegies, and Remco Duits (2024). "Semiring Activation in Neural Networks." In: arXiv: 2405.18805 [cs.LG] (cit. on pp. 5, 154).

Smets, Bart M.N., Jim Portegies, Erik J. Bekkers, and Remco Duits (2022). "PDE-based group equivariant convolutional neural networks." In: Journal of Mathematical Imaging and Vision, 65.1, pp. 209–239. doi: 10.1007/s10851-022-01114-x (cit. on pp. 3, 5, 93, 98, 114, 115, 143).

Smets, Bart M.N., Jim Portegies, Etienne St-Onge, and Remco Duits (2021). "Total variation and mean curvature PDEs on the homogeneous space of positions and orientations." In: Journal of Mathematical Imaging and Vision, 63, pp. 237–262. doi: 10.1007/s10851-020-00991-4 (cit. on pp. 6, 181).

Smith, Leslie N and Nicholay Topin (2019). "Super-convergence: Very fast training of neural networks using large learning rates." In: Artificial intelligence and machine learning for multi-domain operations applications, vol. 11006. SPIE, pp. 369–386 (cit. on p. 172).

Soille, Pierre et al. (1999). Morphological image analysis: principles and applications, vol. 2. 3. Springer (cit. on p. 95).

Staal, Joes, Michael D Abràmoff, Meindert Niemeijer, Max A Viergever, and Bram Van Ginneken (2004). "Ridge-based vessel segmentation in color images of the retina." In: IEEE transactions on medical imaging, 23.4, pp. 501–509. url: https://www.isi.uu.nl/Research/Databases/DRIVE/ (cit. on pp. 144, 146).

Strömberg, Thomas (2002). "The Hopf-Lax formula gives the unique viscosity solution." In: Differential and Integral Equations, 15.1, pp. 47–52. DOI: 10.57262/die/1356060882 (cit. on p. 98).

Sütfeld, Leon René, Flemming Brieger, Holger Finger, Sonja Füllhase, and Gordon Pipa (2018). Adaptive Blending Units: Trainable Activation Functions for Deep Neural Networks, arXiv: 1806.10064 [cs.LG] (cit. on p. 155).

Tao, Terence (2010). "Differential forms and integration." In: The Princeton companion to mathematics, ed. by Timothy Gowers, June Barrow-Green, and Imre Leader. Princeton University Press. Chap. III.16, pp. 175–185 (cit. on p. 25).

Tournier, J-Donald, Fernando Calamante, and Alan Connelly (2007). "Robust determination of the fibre orientation distribution in diffusion MRI: non-negativity constrained super-resolved spherical deconvolution." In: Neuroimage, 35.4, pp. 1459–1472 (cit. on pp. 226, 228, 229).

Varadhan, Sathamangalam R Srinivasa (1967). "On the behavior of the fundamental solution of the heat equation with variable coefficients." In: Communications on Pure and Applied Mathematics, 20.2, pp. 431–455 (cit. on p. 117).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." In: Advances in neural information processing systems, 30 (cit. on p. 154).

Vázquez, Juan Luis (2017). The mathematical theories of diffusion. Nonlinear and fractional diffusion, arXiv: 1706.08241 [math.AP] (cit. on p. 115).

Vogel, Curtis R and Mary E Oman (1996). "Iterative methods for total variation denoising." In: SIAM Journal on Scientific Computing, 17.1, pp. 227–238 (cit. on p. 4).

Vogt, Thomas and Jan Lellmann (2018). "Measure-valued variational models with applications to diffusion-weighted imaging." In: Journal of Mathematical Imaging and Vision, 60, pp. 1482–1502 (cit. on pp. 182, 183).

Weickert, Joachim (1996). Theoretical foundations of anisotropic diffusion in image processing, Springer (cit. on p. 97).

Weickert, Joachim (1999). "Coherence-enhancing diffusion filtering." In: International journal of computer vision, 31, pp. 111–127 (cit. on pp. 183, 184, 197, 230).

Weickert, Joachim, Sven Grewenig, Christopher Schroers, and Andrés Bruhn (2016). "Cyclic schemes for PDE-based image analysis." In: International Journal of Computer Vision, 118, pp. 275–299 (cit. on p. 97).

Weiler, Maurice and Gabriele Cesa (2019). "General E(2)-equivariant steerable cnns." In: Advances in neural information processing systems, 32 (cit. on p. 73).

Weiler, Maurice, Patrick Forré, Erik Verlinde, and Max Welling (2021). "Coordinate Independent Convolutional Networks–Isometry and Gauge Equivariant Convolutions on Riemannian Manifolds." In: arXiv preprint arXiv:2106.06020, (cit. on p. 97).

Weiler, Maurice, Fred A Hamprecht, and Martin Storath (2018). "Learning steerable filters for rotation equivariant CNNs." In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 849–858 (cit. on p. 73).

Weinan, E (2017). "A proposal on machine learning via dynamical systems." In: Communications in Mathematics and Statistics, 1.5, pp. 1–11 (cit. on p. 96).

Welk, Martin and Joachim Weickert (2019). "PDE evolutions for M-smoothers: from common myths to robust numerics." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 236–248 (cit. on p. 97).

Wightman, Ross (2019). PyTorch Image Models, https://github.com/rwightman/pytorch-image-models. DOI: 10.5281/zenodo.4414861 (cit. on p. 174).

Winkels, Marysia and Taco S. Cohen (2018). "3D G-CNNs for pulmonary nodule detection." In: arXiv preprint arXiv:1804.04656, (cit. on p. 73).

Woo, Sanghyun, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie (2023). "Convnext v2: Co-designing and scaling convnets with masked autoencoders." In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16133–16142 (cit. on p. 95).

Worrall, Daniel and Gabriel Brostow (2018). "Cubenet: Equivariance to 3d rotation and translation." In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 567–584 (cit. on p. 73).

Worrall, Daniel E., Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow (2017). "Harmonic networks: Deep translation and rotation equivariance." In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5028–5037 (cit. on p. 74).

Xiao, Han, Kashif Rasul, and Roland Vollgraf (Aug. 28, 2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, arXiv: cs.LG/1708.07747 [cs.LG] (cit. on p. 170).

Yosida, Kosaku (1968). Functional Analysis, Springer. DOI: 10.1007/978-3-662-11791-0 (cit. on pp. 121, 122).

Zhu, Wei and Tony Chan (2012). "Image denoising using mean curvature of image surface." In: SIAM Journal on Imaging Sciences, 5.1, pp. 1–32 (cit. on p. 4).

LIST OF PUBLICATIONS

Duits, Remco, Etienne St-Onge, Jim Portegies, and **Bart M.N. Smets** (2019). "Total variation and mean curvature PDEs on the space of positions and orientations." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 211–223. doi: 10.1007/978-3-030-22368-7_17.

Duits, Remco, **Bart M.N. Smets**, Erik J. Bekkers, and Jim Portegies (2021). "Equivariant deep learning via morphological and linear scale space PDEs on the space of positions and orientations." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 27–39. doi: 10.1007/978-3-030-75549-2_3.

Duits, Remco, **Bart M.N. Smets**, A.J. Wemmenhove, Jim Portegies, and Erik J. Bekkers (2021). "Recent geometric flows in multi-orientation image processing via a Cartan connection." In: Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision, pp. 1–60. doi: 10.1007/978-3-030-03009-4_101-1.

Smets, Bart M.N., Jim Portegies, Etienne St-Onge, and Remco Duits (2021). "Total variation and mean curvature PDEs on the homogeneous space of positions and orientations." In: Journal of Mathematical Imaging and Vision, 63, pp. 237–262. doi: 10.1007/s10851-020-00991-4.

Smets, Bart M.N., Jim Portegies, Erik J. Bekkers, and Remco Duits (2022). "PDE-based group equivariant convolutional neural networks." In: Journal of Mathematical Imaging and Vision, 65.1, pp. 209–239. doi: 10.1007/s10851-022-01114-x.

Bellaard, Gijs, Daan L.J. Bon, Gautam Pai, **Bart M.N. Smets**, and Remco Duits (2023). "Analysis of (sub-) Riemannian PDE-G-CNNs." In: Journal of Mathematical Imaging and Vision, pp. 1–25. DOI: 10.1007/s10851-023-01147-w.

van den Berg, Nicky J., Shuhe Zhang, **Bart M.N. Smets**, Tos T.J.M. Berendschot, and Remco Duits (2023). "Geodesic tracking of retinal vascular trees with optical and TV-flow enhancement in SE(2)." In: International Conference on Scale Space and Variational Methods in Computer Vision, Springer, pp. 525–537. DOI: 10.1007/978-3-031-31975-4_40.

Pai, Gautam, Gijs Bellaard, **Bart M.N. Smets**, and Remco Duits (2023). "Functional properties of PDE-based group equivariant convolutional neural networks." In: International Conference on Geometric Science of Information, Springer, pp. 63–72. DOI: 10.1007/978-3-031-38271-0_7.

Smets, Bart M.N. (2023). Mathematics of Neural Networks, Lecture notes 2MMA80 (2021-2023). arXiv: 2403.04807 [cs.LG].

Bellaard, Gijs, Sei Sakata, **Bart M. N. Smets**, and Remco Duits (2024). "PDE-CNNs: Axiomatic Derivations and Applications." In: arXiv: 2403.15182 [cs.LG].

van den Berg, Nicky J., **Bart M.N. Smets**, Gautam Pai, Jean-Marie Mirebeau, and Remco Duits (2024). "Geodesic tracking via new data-driven connections of Cartan type for vascular tree tracking." In: Journal of Mathematical Imaging and Vision, 66, pp. 198–230. DOI: 10.1007/s10851-023-01170-x.

Smets, Bart M.N., Peter D. Donker, Jim Portegies, and Remco Duits (2024). "Semiring Activation in Neural Networks." In: arXiv: 2405.18805 [cs.LG].

INDEX

Symbols

A

B

C

T

V

ABOUT THE AUTHOR

Bart Smets was born on the 5th of October, 1984, in Neerpelt, Belgium. He finished his secondary school education at Salvator College in Hamont-Achel in 2002. After a number of years working in transport and logistics he decided on a career switch and enrolled at the TU/e to study applied mathematics in 2014. He obtained his bachelor's degree in 2017 and his master's degree in 2019, specializing in analysis and geometry. In the same year he started as a PhD-candidate and teaching assistant, again in Eindhoven. His research focuses on the interface between *geometry* and *deep learning*, with an eye towards applications in *image processing*. The results of this research are presented in this dissertation.

> *"Since all creatures are ordered to God as to an ultimate end, all achieve this end to the extent that they participate somewhat in His likeness. Intellectual creatures attain it in a special way, that is, through their proper operation of understanding Him. Hence, this must be the end of the intellectual creature, namely, to understand God."*
>
> — *Saint Thomas Aquinas (1269)*

ACKNOWLEDGMENTS

At the time of this writing I have been at the TU/e for a decade. This period has been a wonderful time in my life, made possible by the many people that supported me in one way or another. I am grateful to everyone who has surrounded me these past years.

My *maternal grandparents* in particular deserve mention for being supportive of my choice to go to university again. They must have had concerns and doubts about this course of action knowing that the first time did not end so well. So I am grateful that they still got to see me reach this point and find a calling in life.

Thanks to my sister *Caroline*, her husband *Giovanni* and sons *Julian* & *Jake* and my brother *Philippe*, his wife *Petra* and daughter *Nina-Marie* for the many happy times together. I adore my niece and nephews and I am grateful to be part of their lives.

I am also grateful for my friends: *Joeri*, *Claudine*, *Ellen*, *An*, *Tine*, *Jan*, *Judith*, *Thijs*, *Marco* & *Ymke*. It is remarkable to think about how long we have been friends already, over two-thirds of our lives for most of us. You have all been a stable presence for so long now that I can not imagine life without you. I look forward to our next cycling trip . . . that I will do by car of course.

Going to work the last few years has been a pleasure mainly due to my immediate colleagues: *Gijs*, *Nicky*, *Gautam*, *Finn*, *Rick*, *Sjors* & *Lars*. It is a privilege to be able to talk to people this bright on a daily basis and I have enjoyed our many conversations over lunch and 'working class' meetings.

More broadly, CASA has been a great environment to work in with many great colleagues to interact with during varied activities. Particular thanks to our secretaries *Enna*, *Diane* & *Gea* for keeping the whole organization running smoothly.

Many thanks to my committee members prof.dr. *Christoph Brune*, prof.dr. *Giovanna Citti*, prof.dr. *Christoph Schnörr* and prof.dr.ir. *Barry Koren* for taking the time to read my thesis and participating in my defense.

Special mention is merited for *Alessandro Di Bucchianico*, who I got to know as a great lecturer when I was a student. Later, when I was a fledgling PhD candidate, he offered me the chance to set up and teach my own course. This would become the course *Mathematics of Neural Networks* that I got to lecture for two years. Making a PhD candidate responsible for a MSc-level course seems like a risky plan, but with Alessandro's support and advice everything turned out well. During this experience I learned that I quite like teaching, something I did not think I would. This realization has contributed to my choice to pursue an academic career, so I am grateful to have been given this opportunity. Alessandro was also supposed to be part of my promotion committee, but sadly he passed away recently after a sudden and brief illness. I kept his name on the list of committee members at the start of this thesis as a tribute. He will be missed.

I was lucky to have *Jim* as co-promoter. Jim has a reputation for caring greatly for his students and I can confirm that this is well-deserved. He takes his time to think deeply about questions posed to him so as to give the most helpful answer possible. Once he answers he always manages to give just the right insight to move our work along. Jim: thank you for your time, effort and care.

Finally, the person I owe the most gratitude to is *Remco*. I met Remco for the first time while I was looking for a topic for a bachelor project in 2017. Our initial meeting lasted only a few minutes, little did I know what an inflection point in my life that meeting turned out to be. Remco: thank you for being my mentor for these past 7 years, it has been an incredible journey.

— Bart M.N. Smets
Overpelt, August 2024

"We have to remember these days, because there's no guarantee that they will last forever. Enjoy them as long as they last!"

— *Sebastian Vettel*