EUCLIDEAN EQUIVARIANT PROCESSING AND MACHINE LEARNING ON POSITION-ORIENTATION SPACE

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 woensdag 26 november 2025 om 13:30 uur

door

GIJS BELLAARD

geboren te Geertruidenberg

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

voorzitter:	prof. dr. K.M. van Hee	
promotor:	dr. ir. R. Duits	
co-promotor:	dr. ir. B.M.N. Smets	
leden:	prof. dr. C. Brune	(Universiteit Twente)
	prof. dr. V. Dolean-Maini	
	prof. dr. S.F. Portegies Zwart	(Universiteit Leiden)
	dr. ir. E.J. Bekkers	(Universiteit van Amsterdam)

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

COLOPHON

This document was typeset using Typst 0.13.1 (`https://typst.app`).

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

ISBN: 978-90-386-6537-5.

Printed by ADC Nederland (`https://adc.nl`).

CONTENTS

INTRODUCTION

THIS THESIS explores the interplay between Euclidean equivariant processing, position-orientation space, and machine learning. The research is primarily theoretical, with concrete applications across various domains. However, the main practical focus and experimental efforts of this work lie in the domain of machine learning.

The following introduces and motivates the concepts of Euclidean equivariant processing, position-orientation space, and machine learning.

EUCLIDEAN EQUIVARIANT PROCESSING. Take any object in your hand – say, this thesis – and move and rotate it around. While doing this, the object remains the same. When humans recognize or classify objects, we are not thrown off by where they are or how they are rotated. Our recognition automatically accounts for such changes.

The laws of physics also respect translational and rotational transformations. For example, it doesn't matter how a billiard table is positioned or oriented in a room: as long as the table is level, the billiard balls behave in the same predictable way.

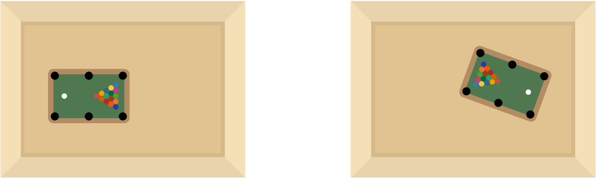
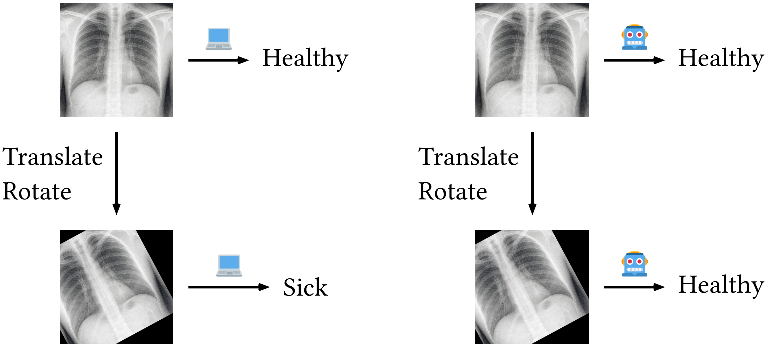


Figure 1: Regardless of the position and orientation of the billiard table, the balls move as expected and the game proceeds as normal.

These observations might seem trivial or superficial, but consider the following. Suppose we are tasked with creating a computer program capable of detecting cancerous cells in a medical image, such as an X-ray of lungs. If a slight shift or rotation of the image caused the program to produce an entirely different diagnosis, the consequences could be disastrous. It is therefore important that the program behaves as expected under translational and rotational transformations, just like how we recognize objects.



A model 💻 whose output varies under transformations of the input can be undesirable.

An equivariant model 🤖 gives the same output, no matter how the input is transformed.

Figure 2: In some tasks it is important that a model respects translational and rotational transformations.

Generally, we say a model or process is *equivariant* if it respects a desired set of transformations. The transformations that are considered in this thesis are rotations, translations, and, occasionally, reflections. These transformation are called *Euclidean transformations*, which all together form the *Euclidean group*. In short, we consider *Euclidean (group) equivariant processing*.

Position-Orientation Space. A lot of things in the world can be satisfactorily described in terms of a position. For example, the location of your home. However, sometimes more information is needed. For instance, when navigating your position is not enough to satisfactorily describe your situation: you also want to know which way you are *oriented*. The collection of all possible position-orientations is called *position-orientation space*.

The location of your home is satisfactorily described with a position, as indicated by the pin.

Where you are during navigation is better described as a position-orientation, as indicated by the arrow.

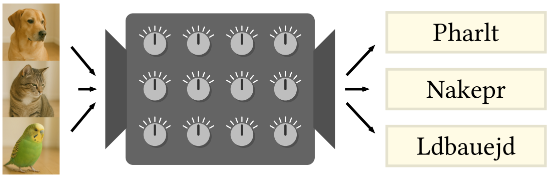
Figure 3: Some things are described using only a position, other things are better described as a position-orientation.

When tackling problems in which directional information is important, it makes sense to work with a space which can represent this information. However, even in problems in which directional information is not of immediate importance, using position-orientation space can still be beneficial.

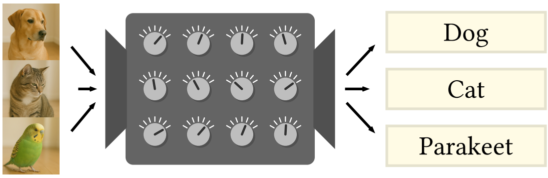
Machine Learning. As of the writing of this thesis, more and more problems are being tackled on computers using *machine learning* (ML), a field concerned with the development and study of models that can "learn" from data and generalize to unseen data.

Machine learning has led to major breakthroughs in tasks previously considered intractable by classical methods. For example, image classification (Krizhevsky, Sutskever and Hinton (2012)), language understanding (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser and Polosukhin (2017)), and protein-structure prediction (Tunyasuvunakool et al. (2021)).

At a high level, *supervised* machine learning can be described as follows. We start with a *black-box* model with a large number – say, hundreds of thousands – of parameters, which are randomly initialized. At initialization, the model outputs random, and most likely wrong, responses to inputs. We give the model a large collection of example input-output pairs, that being the *training data*, and keep on tweaking the parameters such that the model performs better on this data. This process is the “learning” of machine learning. By reasons not yet fully understood, the models we make and train this way also give the correct answers on inputs that are *not* in the training data, that is, they have successfully *generalized*.



An untrained model produces outputs
that are essentially random.



A properly trained model produces
correct outputs for given inputs.

Figure 4: A cartoon illustration of a black-box model. The tunable knobs illustrate the values of the parameters of the model. The training data here consists of images of animals together with a label telling us what is in the image. The parameters of the trained model have been finely tuned, as indicated by the rotated knobs.

Machine learning models that are designed to respect Euclidean transformations generally need less training data and fewer parameters to reach a desired accuracy, in comparison to models that do not (Cohen, Weiler, Kicanaoglu and Welling (2019), Mohamed, Cesa, Cohen and Welling (2020)). For this reason Euclidean equivariant processing is of interest within the context of machine learning.

Generally, the subject of making machine learning models respect a set of transformations is called *(group) equivariant machine learning* (Cohen and Welling (2016), Cohen (2021)) or *geometric (deep) learning* (Bronstein, Bruna, LeCun, Szlam and Vandergheynst

(2017), Bronstein, Bruna, Cohen and Veličković (2021)). Equivariant machine learning methods can be linked to more classical image processing approaches, such as scale-space representations (Witkin (1983), Koenderink (1984), Jacobsen, van Gemert, Lou and Smeulders (2016), Worrall and Welling (2019)), steerable filters (Freeman and Adelson (1991), Cohen and Welling (2017)), and scattering transforms (Mallat (2012), Sifre and Mallat (2013), Bruna and Mallat (2013)). This connection is natural, as these classical methods were also designed to respect the inherent symmetries of images, such as translations, rotations, and scalings.

Concrete Applications

Euclidean equivariant processing, position-orientation space, and machine learning have been explained and motivated at a high level. What follows are concrete examples of works that use, either all or some of, these concepts to their benefit.

In Duits, Smets, Bekkers and Portegies (2021), Smets, Portegies, Bekkers and Duits (2023), Bellaard, Bon, Pai, Smets and Duits (2023), Bellaard, Pai, Oliván Bescós and Duits (2023), and Pai, Bellaard, Smets and Duits (2023), a machine learning framework called *PDE-based neural networks* is proposed and explored. This framework is Euclidean equivariant and uses scalar fields on position-orientation space as its feature maps. It has been used to segment vessels in retinal images and coronary arteries in X-ray angiograms.

In Bellaard, Sakata, Smets and Duits (2025) a specific kind of PDE-based neural network, called a *PDE-CNN*, is explored. The network is Euclidean equivariant and uses semifield valued signals on Euclidean space as its feature maps. It has been used to segment vessels in retinal images.

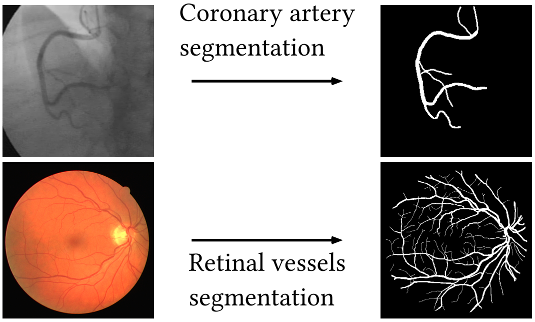


Figure 5: An example of coronary artery and retinal vessel segmentation. Images taken from the DCA1 (Subsection 2.8.2) and DRIVE (Subsection 2.8.1) dataset.

In Bekkers, Vadgama, Hesselink, van der Linden and Romero (2024), Bellaard, Smets and Duits (2025), and Bellaard and Smets (2025), a machine learning model called *PONITA* is introduced and explored. This model is Euclidean equivariant and uses scalar fields on position-orientation space as its feature maps. It has been used to predict the dynamics and chemical properties of molecules and generate new molecules with high atomic and molecular stability.

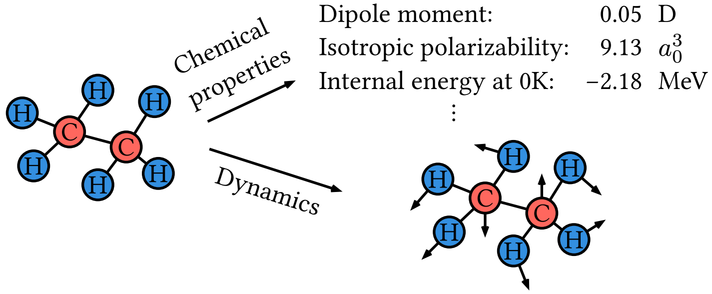


Figure 6: An example of an ethane molecule and its chemical properties and dynamics. The black arrows indicate the forces on each atom resulting from the molecule's deviation from equilibrium.

In Bekkers, Duits, Mashtakov and Sanguinetti (2015), Duits, Meesters, Mirebeau and Portegies (2018), and van den Berg, Smets, Pai, Mirebeau and Duits (2024), data-driven Euclidean equivariant metrics and connections on position-orientation space are used to track vessels in retinal images.

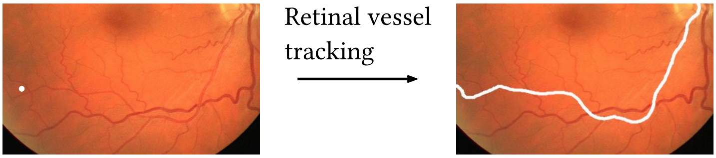


Figure 7: An example of retinal vessel tracking. Starting from the white dot, the goal is to follow the vessel towards to optic disc (top right). Image taken from the DRIVE (Subsection 2.8.1) dataset.

In Bon, Pai, Bellaard, Mula and Duits (2025) and Pai, Bellaard, Sengers, Florack and Duits (2025), optimal transport (Peyré and Cuturi (2020), Santambrogio (2015)) is extended to position-orientation space in a Euclidean equivariant manner. With this extension they are able to perform more meaningful interpolation of images that contain well-defined contours.

In Duits and Franken (2010), Duits and Franken (2011), Portegies, Fick, Sanguinetti, Meesters, Girard and Duits (2015), Portegies, Sanguinetti, Meesters and Duits (2015), Portegies (2018), Duits, St-Onge, Portegies and Smets (2019), Chambolle and Pock (2019), and Smets, Portegies, St-Onge and Duits (2021) PDEs on position-orientation space are used for crossing-aware denoising and enhancement of data, all in a Euclidean equivariant manner.

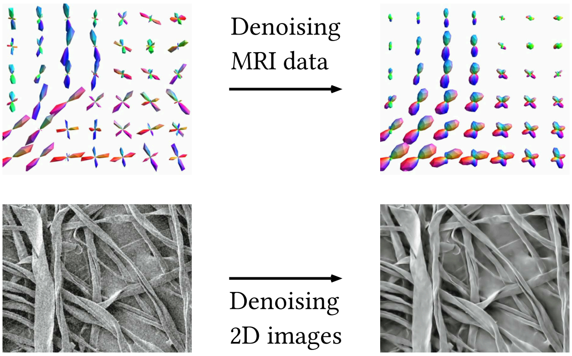


Figure 8: Two examples of denoising/enhancing data. The MRI data (more accurately, the *fiber orientation density*) is visualized as a *glyph field* where each glyph is a *spherical polar plot* (Tuch et al. (2002), Eq.4, Margulies et al. (2013), Fig.2f).

OUTLINE

The thesis is structured as follows. Chapters 1 and 2 are prerequisite to all other chapters. Chapters 3 to 7 are independent and can be read in any order.

- In Chapter 1 we present the necessary mathematical preliminaries. The main purpose of this chapter is to establish notation and conventions.
- In Chapter 2 we review the relevant background, that being Euclidean groups, position-orientation space, scale-space representations, and machine learning. We also provide a more detailed discussion of two specific machine learning architectures – PDE-based neural networks and PONITA – as these serve as the primary "experimental subjects" of this thesis.
- In Chapter 3 we cover BELLAARD, BON, PAI, SMETS AND DUITS (2023) and examine approximations of the Riemannian distance of a Euclidean equivariant metric on two-dimensional position-orientation. We analyze if these approximations can improve the accuracy of PDE-G-CNNs when segmenting coronary arteries and performing contour completion.
- In Chapter 4 we cover BELLAARD AND SMETS (2025) and classify and parametrize all Euclidean invariant Riemannian metrics on three-dimensional position-orientation space. We describe the *mav distance* as a computationally efficient alternative to the Riemannian distance. We analyze if the mav distance can improve the accuracy of PONITA when predicting molecular properties.
- In Chapter 5 we cover BELLAARD, SMETS AND DUITS (2025) and rigorously describe an independent and universal collection of Euclidean invariants between pairs of three-dimensional position-orientations. We analyze if this collection of invariants can improve the accuracy of PONITA when predicting molecular properties.
- In Chapter 6 we cover BELLAARD, PAI, OLIVÁN BESCÓS AND DUITS (2023) and experimentally investigate two geometric adaptations of PDE-G-CNNs on two-dimensional position-orientation space. First,

we analyze if a *fixed lifting layer* using *cake wavelets* can perform just as well as a trained lifting layer. Second, we implement a more general class of Riemannian metrics within the framework.

- In Chapter 7 we cover Bellaard, Sakata, Smets and Duits (2025) and generalize scale-space theory on two-dimensional Euclidean space to semifields. Our theory reveals new scale-spaces to use within PDE-based neural networks and we experimentally examine what impact these have on its accuracy when segmenting vessels in retinal images.

Outtakes

The following works are not the focus of this thesis, as I was not the primary author.

- Pai, Bellaard, Smets and Duits (2023) examine PDE-G-CNNs by theoretically describing their geometric and algebraic symmetries, experimentally confirming their data efficiency, and showing that one can create an equivariant U-Net (Ronneberger, Fischer and Brox (2015)) variant of it.
- Bon, Pai, Bellaard, Mula and Duits (2025) extend optimal transport to position-orientation space.
- Pai, Bellaard, Sengers, Florack and Duits (2025) propose using data-driven metrics on position-orientation space to perform optimal transport.
- Duits, Bellaard and Tumpach (2025) analyze different sections of the special Euclidean group $\mathrm{SE}(3)$ when viewed as a circle bundle over position-orientation space $\mathbb{M}_3$.

PRELIMINARIES

1

In this chapter, we recapitulate some common terminology, notation and definitions used throughout the thesis. Outside of these preliminaries, we assume the reader is familiar with basic (multi)linear algebra, calculus, topology, measure theory, and group theory.

In this thesis, whenever the regularity of a mathematical object (function, manifold, section, etc.) is relevant, we assume it to be smooth, i.e. infinitely differentiable. Consequently, we will usually omit the adjective "smooth". Similarly, whenever a group acts on another mathematical object *unrelated* to the group, the action is taken to be on the *left.* Hence, we will generally omit the adjective "left" when discussing group actions and related notions (invariance, equvariance, etc.). On groups themselves, however, we *will* distinguish between left and right actions.

1.1 GROUP ACTIONS, INVARIANCE & EQUIVARIANCE

A central topic of this thesis is invariance and equivariance. In this section we recall what this means in the context of group theory.

Consider a (left) group action $\triangleright : G \times X \to X$ of a group G on a set X. The *orbit* Gx of an element $x \in X$ is defined as the set $Gx := \{g \triangleright x \mid g \in G\}$. The *stabilizer* $\mathrm{stab}(x)$ of an element $x \in X$ is defined as the set $\mathrm{stab}(x) := \{g \in G \mid g \triangleright x = x\}$. Define $\sim$ as the equivalence relation on X indicating if two elements are in the same orbit, that is $x_1 \sim x_2 \Leftrightarrow \exists g : g \triangleright x_1 = x_2$.

A group action is *transitive* if for every two elements $x_1, x_2 \in X$ there exists a $g \in G$ such that $g \triangleright x_1 = x_2$. An action is *faithful* if for all $g \in G \setminus \{e\}$ there exists a $x \in X$ with $g \triangleright x \neq x$. An action is *free* if for all $g \in G \setminus \{e\}$ and $x \in X$ we have $g \triangleright x \neq x$.

Every group action G on a set X induces a natural action on functions f on X. Namely, we define $(g \triangleright f)(x) := f(g^{-1} \triangleright x)$, also called *translation of functions.*

An *invariant element* $x \in X$ is an element for which $g \triangleright x = x$ for all $g \in G$. An *invariant function* f on X is a function that satisfies $f(g \triangleright x) = f(x)$ for all $g \in G$ and $x \in X$. Using the orbit equivalence relation $\sim$ one can equivalently write $x_1 \sim x_2 \Rightarrow f(x_1) = f(x_2)$ for all $x_1, x_2 \in X$. Using the action on functions one can also write $f = g \triangleright f$ for all $g \in G$. In that sense, an invariant function can also be understood as an invariant element of a function space.

Consider a group G acting $\triangleright$ on two sets X and Y (we use the same symbol $\triangleright$ for both the action on X and Y). An *equivariant function* $f : X \to Y$ is a function for which $f(g \triangleright x) = g \triangleright f(x)$ for all $g \in G$ and $x \in X$. The action of G on Y can be the trivial one, that is, $g \triangleright y = y$ for all $g \in G$ and $y \in Y$. In this sense, every invariant function can also be understood as an equivariant function.

1.2 DIFFERENTIAL GEOMETRY

The Euclidean groups and position-orientation space, objects central to this thesis, are examples of smooth manifolds. This section reviews differential geometric concepts necessary for their analysis. See LEE (2012) for an introduction to differential geometry.

A *coordinate chart* $x : U \to (\operatorname{im} x \subseteq \mathbb{R}^n)$ is a homeomorphism from an open subset $U \subseteq X$ of a topological space X to an open subset $\operatorname{im} x$ of $\mathbb{R}^n$. Given two charts x, y that overlap, define the *transition maps* $x \circ y^{-1}$ and $y \circ x^{-1}$. An *atlas* A is a collection of charts on X that together cover X, such that all transition maps between overlapping charts are smooth.

A *manifold* M is a Hausdorff and second countable topological space, together with an atlas on M.

A *diffeomorphism* $\varphi : M \to N$ is a bijective map between two manifolds M and N such that both φ and φ^{-1} are smooth. Two manifolds are called *diffeomorphic* if there exists a diffeomorphism between them.

A *fiber bundle* (E, M, F, π) consists of three manifolds: the *total space* E, the *base space* M, and the *model fiber* F, together with a surjective smooth map $\pi : E \to M$, called the *projection*, such that for all $p \in M$ there is a neighborhood $p \in U \subseteq M$ such that $\pi^{-1}(U)$ is diffeomorphic to $U \times F$. The preimage $\pi^{-1}(p) \subseteq E$ is called the *fiber* over p, and is diffeomorphic to the model fiber F. A *section* $\Gamma : M \to E$ of a fiber bundle is a smooth map satisfying $\pi \circ \Gamma = \mathrm{id}$, that being a smooth choice of a point in each fiber. The set of all sections is denoted by $\Gamma(E)$.

A *scalar field* $f : M \to \mathbb{R}$ on a manifold M is a smooth map from M to the scalars $\mathbb{R}$. We call it a *field* because we want to emphasize that one should imagine a scalar $f(p)$ attached at every point $p \in M$. The set of all smooth scalar fields on M is denoted with $C^\infty(M)$.

A *curve* $\gamma : I \to M$ on a manifold M is a (piecewise) continuously differentiable map from an interval $I \subseteq \mathbb{R}$ to the manifold M. If $I = [a, b]$ is a closed interval we call $\gamma(a)$ the *start* of the curve and $\gamma(b)$ the *end*.

A *tangent vector* v at a point $p \in M$ on manifold M is an equivalence class of curves $\gamma : I \to M$ through $\gamma(0) = p$, where two curves γ_1, γ_2 are equivalent if $(f \circ \gamma_1)'(0) = (f \circ \gamma_{2'})(0)$ for all scalar fields $f \in C^\infty(M)$. For any tangent vector v and scalar field $f \in C^\infty(M)$ we define the natural "application" $v(f) := (f \circ \gamma)'(0)$ where γ is any curve in the equivalence class v. That is, a tangent vector v at p can be understood as a directional derivative at p. The *tangent space* T_pM is the vector space of all tangent vectors at p. The *tangent bundle* TM is the collection of all of the tangent spaces T_pM. The tangent bundle naturally inherits a smooth structure, making it a manifold. A *vector field* $X \in \Gamma(TM)$ is a section of the tangent bundle. The application of tangent vectors to scalar fields $f \in C^\infty(M)$ extends naturally to vector fields $X \in \Gamma(TM)$ by defining pointwise $(Xf)(p) := X_p f$.

A *cotangent vector* $v^* : T_pM \to \mathbb{R}$ at a point $p \in M$ on a manifold M is a linear map from the tangent space T_pM at p to the real numbers. The *cotangent space* T_p^*M is the vector space of all cotangent vectors at p. The *cotangent bundle* T^*M is the collection of all of the cotangent spaces T_p^*M. The cotangent bundle naturally inherits a smooth structure, making it a manifold in its own right. A *covector field* $X^* \in \Gamma(T^*M)$ is a section of the cotangent bundle.

The *canonical pairing* $\langle \cdot, \cdot \rangle : T_p^*M \times T_pM$ at a point $p \in M$ on a manifold M is defined as $\langle v^*, v \rangle := v^*(v)$ where v^* is a cotangent vector and v a tangent vector at p.

A *tensor* at a point $p \in M$ on a manifold M is an arbitrary tensor product $\otimes$ of tangent vectors and cotangent vectors at p. The *tensor bundle* $T^{m,n}M$ of type $(m, n) \in \mathbb{N} \times \mathbb{N}$ is defined as the tensor product bundle $T^{m,n}M := (TM)^{\otimes m} \otimes (T^*M)^{\otimes n}$. A *tensor field* $T \in \Gamma(T^{m,n}M)$ is a section of a tensor bundle (of any type).

A *differential k-form* $\omega \in \Gamma\left(\bigwedge^k T^*M\right)$ on a manifold M is a section of the k-th exterior power of the cotangent bundle. That is, it assigns to each point $p \in M$ an alternating multilinear map $\omega_p : \left(T_pM\right)^k \to \mathbb{R}$.

The *differential* $\mathrm{d}f : TM \to \mathbb{R}$ of a scalar field $f \in C^\infty(M)$ on a manifold M is defined by $\mathrm{d}f(v) = v(f)$ for all tangent vectors $v \in TM$. We can also understand the differential as a covector field $\mathrm{d}f \in \Gamma(T^*M)$.

Consider coordinates $x : U \to \mathbb{R}^n$ on an open subset U of a manifold M. Define the *coordinate tangent vectors* $\left(\partial x^i\right)_p \in T_pU$ at a point $p \in M$ by $\left(\partial x^i\right)_p := \dot{\gamma}(0)$, where $\gamma(t) = x^{-1}(x(p) + te_i)$ is the curve through $\gamma(0) = p$ that moves purely in the x^i direction. The vector fields $\partial x^i \in \Gamma(TU)$ are the *coordinate vector fields.* Define the *coordinate covector fields* $\mathrm{d}x^i \in \Gamma(T^*U)$ as the differential of the coordinate fields $x^i : U \to \mathbb{R}$, and *coordinate cotangent vectors* as $\mathrm{d}x^i_p$.

The *pushforward of vectors* $\varphi_* : TM \to TN$ of a smooth map $\varphi : M \to N$ between two manifolds M and N is the linear map defined by $\varphi_*(\dot{\gamma}(0)) := (\varphi \circ \gamma)'(0)$ for all curves $\gamma : I \to M$. The *pullback of* *covectors* $\varphi^* : T^*N \to T^*M$ is the linear map defined by $\langle \varphi^* v^*, u \rangle = \langle v^*, \varphi_* u \rangle$ where $u \in T_p M$ and $v^* \in T^*_{\varphi(p)} N$. If the map $\varphi : M \to N$ is a diffeomorphism, we also define the *pullback of vectors* $\varphi^* : TN \to TM$ and the *pushforward of covectors* $\varphi_* : T^*N \to T^*M$ by considering the inverse map φ^{-1}. Through this we can define the pushforward $\varphi_* : T^{m,n}M \to T^{m,n}N$ and pullback $\varphi^* : T^{m,n}N \to T^{m,n}M$ of arbitrary tensors, as long as φ is a diffeomorphism.

An *immersion* $\varphi : M \to N$ is a smooth map between manifolds of which the pushforward $\varphi_* : TM \to TN$ is injective at all $p \in M$. An *immersed submanifold* is the image of an immersion. A *submersion* is a smooth map between manifolds of which the pushforward is surjective everywhere. An *embedding* is an immersion that is also a homeomorphism onto its image. An *embedded submanifold* is the image of an embedding.

The *flow* $\Phi^t : M \to M$ of a vector field $X \in \Gamma(TM)$ on a manifold M is the unique smooth family of diffeomorphisms that satisfy $\frac{\mathrm{d}}{\mathrm{d}t}\Phi^t(p) = X_{\Phi^t(p)}$ and $\Phi^0 = \mathrm{id}$. The flow map satisfies $\Phi^t \circ \Phi^s = \Phi^{t+s}$. Differentiating both sides of this equation with $\frac{\mathrm{d}}{\mathrm{d}s}|_{s=0}$ yields $\Phi^t_* X_p = X_{\Phi^t(p)}$.

The *Lie derivative* $\mathcal{L}_X T$ of a tensor field $T \in \Gamma(T^{m,n}M)$ with respect to a vector field $X \in \Gamma(TM)$ on a manifold M is defined pointwise as $\mathcal{L}_X T = \frac{\mathrm{d}}{\mathrm{d}t}|_{t=0} \left(\Phi^t\right)^* T$ where Φ^t is any smooth family of diffeomorphism with $\frac{\mathrm{d}}{\mathrm{d}t}|_{t=0} \Phi^t(p) = X_p$.

The *Lie bracket of vector fields* $[\cdot,\cdot] : \Gamma(TM) \times \Gamma(TM) \to \Gamma(TM)$ on a manifold M is defined as $[X, Y] := \mathcal{L}_X Y$. Equivalently, the Lie bracket can also be defined as the unique vector field $[X, Y]$ such that $[X, Y]f = X(Yf) - Y(Xf)$ for all scalar fields $f \in C^\infty(M)$.

1.3 RIEMANNIAN GEOMETRY

Building upon differential geometry, Riemannian geometry introduces the concept of a Riemannian metric, allowing for the measurement of distances, angles, and volumes on manifolds. See LEE (2019) for an introduction to Riemannian geometry.

A *Riemannian metric* $\mathcal{G}$ on a manifold M is a choice of an *inner product* $\mathcal{G}_p : T_pM \times T_pM \to \mathbb{R}$ (a symmetric, positive-definite bilinear map) on the tangent space T_pM at every point $p \in M$. More formally, the metric is understood as a section $\mathcal{G} \in \Gamma(T^{0,2}M)$.

A *Riemannian manifold* $(M, \mathcal{G})$ is a manifold M with a Riemannian metric $\mathcal{G}$.

The Riemannian metric $\mathcal{G}$ on a Riemannian manifold M induces a *Riemannian norm* $\|\cdot\| : TM \to \mathbb{R}_{\geq 0}$ defined by $\|v\|^2 := \mathcal{G}(v, v)$. The *Riemannian length* $L(\gamma)$ of a curve $\gamma : [a, b] \to M$ is defined as $L(\gamma) = \int_a^b \|\dot{\gamma}(t)\| \, \mathrm{d}t$. The *Riemannian distance* $d(p, q)$ between two points $p, q \in M$ is defined as $d(p, q) = \inf_{\gamma \in \Gamma} L(\gamma)$ where Γ is the set of curves starting at p and ending at q.

The *gradient* vector field $\nabla f \in \Gamma(TM)$ of a scalar field $f \in C^\infty(M)$ on a Riemannian manifold $(M, \mathcal{G})$ is defined as the unique field such that $\mathcal{G}(\nabla f, Y) = Yf$ for any vector field $Y \in \Gamma(TM)$.

The *Riemannian volume form* ω on a (orientable) n-dimensional Riemannian manifold M is the (up to sign) unique n-form such that $|\omega(e_1, ..., e_n)| = 1$ for any orthonormal frame $e_i \in T_pM$ at any point $p \in M$. The *Riemannian measure* μ is the unique measure on M defined by $\mu(A) = \left|\int_A \omega\right|$ for any measurable set $A \subseteq M$.

The *divergence* $\nabla \cdot X \in C^\infty(M)$ of a vector field $X \in \Gamma(TM)$ on a Riemannian manifold M is defined as the scalar field such that $(\nabla \cdot X)\omega = \mathcal{L}_X \omega$ where $\mathcal{L}$ is the Lie derivative and ω the Riemannian volume form.

The *Laplacian* or *Laplace–Beltrami* $\Delta f \in C^\infty(M)$ of a scalar field $f \in C^\infty(M)$ on a Riemannian manifold M is defined as the divergence of the gradient $\Delta f = \nabla \cdot \nabla f$.

The concepts above, that being the Riemannian norm, length, distance, gradient, and so on, all depend on the Riemannian metric $\mathcal{G}$ under consideration. Whenever multiple metrics are involved, we explicitly indicate this dependence by including the metric in the subscript. That is, we might write $\|\cdot\|_{\mathcal{G}}$, $L_{\mathcal{G}}$, $d_{\mathcal{G}}$, and so on.

1.4 SUB-RIEMANNIAN GEOMETRY

Sub-Riemannian geometry explores a specialized type of Riemannian manifold where movement is restricted to a subbundle of the tangent space. As we will see later in Section 2.2, position-orientation space can be endowed with a natural subbundle. See AGRACHEV, BARILARI AND BOSCAIN (2019) for an introduction to sub-Riemannian geometry.

A vector subbundle $H \subseteq TM$ on a manifold M is *bracket-generating* (or *completely nonintegrable*, *nonholonomic*, or satisfies the *Hörmander condition*) if for all points $p \in M$, the evaluations at p of all iterated Lie brackets of vector fields in $\Gamma(H)$ span the whole tangent space T_pM.

A *sub-Riemannian manifold* $(M, \mathcal{G}, H)$ is a Riemannian manifold $(M, \mathcal{G})$ with a bracket-generating subbundle $H \subseteq TM$ also called the *horizontal subspace*.

A curve $\gamma : I \to M$ in a sub-Riemannian manifold M is called *horizontal* if $\dot{\gamma}(t) \in H$. Because H is bracket-generating, every two points on the manifold can be connected by a horizontal curve, this is known as the *Chow–Rashevskii theorem*.

The *sub-Riemannian distance* or *Carnot–Carathéodory metric* $d(p, q)$ between two points $p, q \in M$ on a sub-Riemannian manifold M is defined in the same way as on a Riemannian manifold, however here *only* horizontal curves are considered.

1.5 LIE THEORY

The Euclidean groups are examples of Lie groups, those being, intuitively, “continuous” collections of transformations. In this section we recollect standard Lie theoretic concepts. See LEE (2012), Chpt.7 and HALL (2015) for an introduction to Lie theory.

A *Lie group* G is a group and a manifold, with both the group product $\cdot : G \times G \to G$ and group inverse $\cdot^{-1} : G \to G$ being smooth maps.

The Lie group product $\cdot : G \times G \to G$ induces two actions on tangent vectors. Namely, the *left pushforward* $\cdot : G \times TG \to TG$ defined by $g \cdot \dot{\gamma}(0) := (g \cdot \gamma)'(0)$, and the *right pushforward* $\cdot : TG \times G \to$

TG defined by $\dot{\gamma}(0) \cdot g := (\gamma \cdot g)'(0)$, for all curves $\gamma : I \to G$. We sometimes drop the $\cdot$ for conciseness.

A *Lie group homomorphism* $\varphi : G \to H$ between two Lie groups G and H is a smooth map that is also a group homomorphism. If φ is in addition a diffeomorphism then it is a *Lie group isomorphism.*

An *exponential curve* $\eta : \mathbb{R} \to G$ in a Lie group G is a one-parameter subgroup, that being a curve satisfying $\eta(s)\eta(t) = \eta(s+t)$ for all $s, t \in \mathbb{R}$. Taking the derivatives $\frac{\mathrm{d}}{\mathrm{d}s}|_{s=0}$ and $\frac{\mathrm{d}}{\mathrm{d}t}|_{t=0}$ on both sides of this equation yields $X\eta(t) = \dot{\eta}(t)$ and $\eta(s)X = \dot{\eta}(s)$, where $X = \dot{\eta}(0)$. That is, η follows *both* the left-invariant vector field $g \mapsto gX$ *and* the right-invariant vector field $g \mapsto Xg$.

The tangent space $\mathfrak{g} := T_e G$ at the identity element e of a Lie group G is usually denoted in fraktur font. Elements $X \in \mathfrak{g}$ are also called *generators.* Every generator $X \in \mathfrak{g}$ corresponds one-to-one to an exponential curve η, and vice versa, through $\dot{\eta}(0) = X$. Generators are called such because they *generate* one-parameter subgroups, that being the corresponding exponential curve.

The *exponential map* $\exp : \mathfrak{g} \to (\operatorname{im}\exp \subseteq G)$ is defined by $\exp(X) := \eta_X(1)$ where η_X is the unique corresponding exponential curve with $\dot{\eta}_X(0) = X$. One can verify that $\exp(tX) = \eta_X(t)$ for all $t \in \mathbb{R}$. A *logarithmic map* $\log : \operatorname{im}\exp \to \mathfrak{g}$ is a *right inverse* of the exponential on $\operatorname{im}\exp$, that is $\exp \circ \log = \mathrm{id}$.

A *Lie group representation* $R : G \to \mathrm{GL}(V)$ is a Lie group homomorphism from a Lie group G to the general linear group of a vector space V. This induces a *Lie algebra representation* $r : \mathfrak{g} \to \mathrm{End}(V)$ by defining $r := (R_*)_e$, that being the pushforward of R around the identity e.

The *adjoint representation* $\mathrm{Ad} : G \to \mathrm{GL}(\mathfrak{g})$ of the Lie group G is defined as $\mathrm{Ad}_g(X) := gXg^{-1}$. The adjoint representation $\mathrm{ad} : \mathfrak{g} \to \mathrm{End}(\mathfrak{g})$ of the generators $\mathfrak{g}$ is $\mathrm{ad} := (\mathrm{Ad}_*)_e$. We define the *Lie bracket of generators* $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ through $[X, Y] := \mathrm{ad}_X Y$, turning $\mathfrak{g}$ into a *Lie algebra.*

Consider a Lie group G. The *linear group convolution* $f_1 * f_2$ of two integrable functions f_1, f_2 on G is defined as $(f_1 * f_2)(g) := \int f_1(h^{-1}g) f_2(h)\, \mu(\mathrm{d}h)$, where μ is a left-invariant measure on G. The *infimal group convolution* $f_1 \boxminus f_2$ of two bounded-from-below functions $f_1, f_2 : G \to \mathbb{R}$ is defined as $(f_1 \boxminus f_2)(g) := \inf_{h \in G}\{f_1(h^{-1}g) + f_2(h)\}$. The *supremal group convolution* $f_1 \boxplus f_2$ of two bounded-from-above functions $f_1, f_2 : G \to \mathbb{R}$ is defined as $(f_1 \boxplus f_2)(g) := \sup_{h \in G}\{f_1(h^{-1}g) + f_2(h)\}$. The infimal and supremal convolution are also known as *morphological group convolutions*. We refer to the linear, infimal, and supremal group convolutions collectively as *group convolutions*. The group convolutions $\star \in \{*, \boxminus, \boxplus\}$ are left-equivariant, that is $f_1 \star (g \rhd f_2) = g \rhd (f_1 \star f_2)$ for all $g \in G$. Here the group acts on functions $f : G \to \mathbb{R}$ via left-translation, that is $(g \rhd f)(h) = f(g^{-1}h)$.

A (left) *Lie group action* $\rhd : G \times M \to M$ of a Lie group G on a manifold M is a smooth group action. The Lie group action induces an action on tangent vectors $\rhd : G \times TM \to TM$ by defining $g \rhd \dot{\gamma}(0) := (g \rhd \gamma)'(0)$ for all curves $\gamma : I \to M$. The Lie group action also induces a *Lie algebra action* $\rhd : \mathfrak{g} \times M \to TM$ by defining $X \rhd p = (\gamma \rhd p)'(0)$, where $\gamma : I \to G$ is any curve with $\dot{\gamma}(0) = X$. The *fundamental vector field* $X^{\#} \in \Gamma(TM)$ of a generator X is $X^{\#}_p := X \rhd p$. The $\rhd$ symbol is usually dropped for conciseness. The group multiplication of a Lie group can be understood as a (either left or right) Lie group action on itself.

Consider a Lie group action $\rhd : G \times M \to M$. An *invariant vector field* $X \in \Gamma(TM)$ is a vector field that satisfies $X_{g \rhd p} = g \rhd X_p$ for all $g \in G$ and $p \in M$. An *invariant Riemannian metric* $\mathcal{G}$ is a Riemannian metric that satisfies $\mathcal{G}_p(u, v) = \mathcal{G}_{g \rhd p}(g \rhd u, g \rhd v)$ for all $u, v \in T_pM$, $g \in G$, and $p \in M$. This induces an *invariant Riemannian distance* d, that is $d(p, q) = d(g \rhd p, g \rhd q)$ for all $g \in G$ and $p, q \in M$.

A *homogeneous space* M is a manifold on which a Lie group G acts transitively. A homogeneous space is called *principal* if the Lie group acts freely. In particular, every Lie group can be understood as an principal homogeneous space by letting the group act on itself.

Fix some reference point $p_0 \in M$ in a homogeneous space M with Lie group G. Every other point $p \in M$ can now be associated one-to-one with the set of elements $G_{p_0 \to p} := \{g \in G \mid gp_0 = p\} \subseteq G$ that map p_0 to p. One can verify that these sets partition the group G and can equivalently be described as *left cosets* of $H = \operatorname{stab} p_0$. In other words, the homogeneous space M can also be understood as the *left coset space* or *quotient space* G/H. This perspective will not be used in this thesis, but we include it for the sake of completeness.

Consider a homogeneous space M with Lie group G. Let $k : M \times M \to \mathbb{R}$ be an invariant *kernel*, that is $k(g \triangleright p, g \triangleright q) = k(p, q)$ for all $p, q \in M$ and $g \in G$. A *linear homogeneous convolution* Φ on M is an operator of the form $(\Phi f)(p) = \int k(p, q) f(q) \, \mu(\mathrm{d}q)$, where μ is an invariant measure on M. An *infimal homogeneous convolution* is $(\Phi f)(p) = \inf_q k(p, q) + f(q)$. An *supremal homogeneous convolution* is $(\Phi f)(p) = \sup_q k(p, q) + f(q)$. We refer to the linear, infimal, and supremal homogeneous convolutions collectively as *homogeneous convolutions*. The convolutions Φ are equivariant, that is $\Phi(g \triangleright f) = g \triangleright (\Phi f)$ for all $g \in G$. Here the group acts naturally on functions $f : M \to \mathbb{R}$ via translation, that is $(g \triangleright f)(p) = f(g^{-1} \triangleright p)$.

BACKGROUND 2

In this chapter we cover the relevant background, that being Euclidean groups, position-orientation space, lifting, scale-space representations, and machine learning. We also provide a more detailed discussion of two specific machine learning architectures – PDE-G-CNNs and PONITA – as these serve as the primary "experimental subjects" of this thesis. Finally, this chapter ends with an overview of the datasets that were used in the experimental sections throughout the various chapters.

2.1 THE (SPECIAL) EUCLIDEAN GROUP

Consider a translation vector $t \in \mathbb{R}^d$ and an orthonormal matrix $Q \in \mathbb{R}^{d \times d}$, that being a matrix with $Q^\top Q = I$. We let a *rigid transformation* (t, Q) act $\triangleright$ (faithfully) on $\mathbb{R}^d$ via

$$(t, Q) \triangleright x = t + Qx, \tag{2.1}$$

that is, we first transform with Q around the origin, and then translate with t.

Rigid transformations are the *distance-preserving* transformations, also known as *isometries* of Euclidean space $\mathbb{R}^d$. This means that for all transformations $g = (t, Q)$ we have that $d(x_1, x_2) = d(g \triangleright x_1, g \triangleright x_2)$ for all $x_1, x_2 \in \mathbb{R}^d$, where $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is the standard Euclidean distance $d(x_1, x_2) := \|x_1 - x_2\|$.

Rigid transformations can be subdivided into two classes. Namely, consider the map $\operatorname{sign}(t, Q) = \det Q$. If $\operatorname{sign}(t, Q) = +1$, the rigid transformation (t, Q) is called *orientation-preserving*, if $\operatorname{sign}(t, Q) = -1$, it is called *orientation-reversing*.

Inspecting the action (2.1) we find that:

- $(0, I) \triangleright x = x$,
- $(t_2, Q_2) \triangleright (t_1, Q_1) \triangleright x = (t_2 + Q_2 t_1, Q_2 Q_1) \triangleright x$,

- $(-Q^{-1}t, Q^{-1}) \triangleright (t, Q) \triangleright x = x$.

The set of all rigid transformations forms the Lie group $\mathrm{E}(d)$, known as the *Euclidean group*:

$$\mathrm{E}(d) := \{(t, Q) \in \mathbb{R}^d \times \mathbb{R}^{d \times d} \mid Q^\top Q = I\}, \tag{2.2}$$

with the identity element, group product, and inverse as listed above. The dimension is $\dim \mathrm{E}(d) = d + d(d-1)/2$, that being d dimensions for the translations, and $d(d-1)/2$ dimensions for the orthonormal matrices.

By restricting the orthonormal matrices Q to be rotation matrices, which we generally denote by R, we get *roto-translations* (t, R), which together form the *Special Euclidean group* or *roto-translation group*:

$$\mathrm{SE}(d) := \{(t, R) \in \mathbb{R}^d \times \mathbb{R}^{d \times d} \mid R^\top R = I, \det R = 1\}. \tag{2.3}$$

with the action, identity element, group product, and inverse kept the same as in the Euclidean group. Equivalently, one can succinctly define $\mathrm{SE}(d)$ as the subgroup of orientation-preserving rigid transformations.

The Lie algebra of both $\mathrm{E}(d)$ and $\mathrm{SE}(d)$ is

$$\mathfrak{e}(d) = \mathfrak{se}(d) = \{(v, \omega) \in \mathbb{R}^d \times \mathbb{R}^{d \times d} \mid \omega^\top + \omega = 0\}. \tag{2.4}$$

We call v the *translation velocity* vector and ω the *angular velocity tensor*. The total *angular velocity* $\|\omega\|$ of an angular velocity tensor ω is defined as

$$\|\omega\| := \frac{1}{\sqrt{2}} \|\omega\|_F \tag{2.5}$$

where $\|\cdot\|_F$ is the *Frobenius norm* $\|A\|_F^2 := \operatorname{tr} A^\top A$.

Remark 2.1: It may seem surprising that two distinct Lie groups can share the same Lie algebra. However, remember that the Lie algebra is fundamentally a local object: it is defined as the tangent space at the identity. It is possible for two globally different objects to look the same locally. As such, the Lie algebras of two globally distinct Lie groups can be identical. In our case, there are no “infinitesimal” rigid transformations that generate reflections, making the Lie algebras of $\mathrm{E}(d)$ and $\mathrm{SE}(d)$ equal.

In three dimensions, the angular velocity tensor ω can be associated with a pseudovector $\vec{\omega}$ called the *angular velocity vector*:

$$\mathbb{R}^{3\times 3} \ni \omega = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \leftrightarrow \vec{\omega} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \in \mathbb{R}^3, \qquad (2.6)$$

with the property that $\omega x = \vec{\omega} \times x$ for all $x \in \mathbb{R}^3$, where on the left we have matrix-vector multiplication, and on the right the cross product $\times$ on $\mathbb{R}^3$. In this case, the angular velocity (2.5) is

$$\|\omega\|^2 = \|\vec{\omega}\|^2 = \omega_1^2 + \omega_2^2 + \omega_3^2. \qquad (2.7)$$

An extremely useful representation of $\mathrm{E}(d)$ is

$$\mathrm{E}(d) \ni (t, Q) \leftrightarrow \begin{pmatrix} Q & t \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}, \qquad (2.8)$$

which induces the Lie algebra representation:

$$\mathfrak{e}(d) \ni (v, \omega) \leftrightarrow \begin{pmatrix} \omega & v \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}. \qquad (2.9)$$

With these representations, the Lie group exponential, logarithm, and Lie algebra bracket correspond respectively to the matrix exponential, matrix logarithm, and matrix commutator, respectively. These operations are available in essentially all *computer algebra systems* (Mathematica, SciPy, SageMath, Maple, etc…), which make it possible to work with $\mathrm{E}(d)$ in a numerical and practical manner.

2.1.1 Two-Dimensional Special Euclidean Group

For the Lie algebra $\mathfrak{se}(2)$ of the two-dimensional special Euclidean group SE(2) we choose the following basis of generators:

$$
\begin{aligned}
A_1 &= \left(\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0 & 0\\0 & 0\end{pmatrix}\right),\\
A_2 &= \left(\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0 & 0\\0 & 0\end{pmatrix}\right),\\
A_3 &= \left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0 & -1\\1 & 0\end{pmatrix}\right).
\end{aligned}
\tag{2.10}
$$

In Figure 2.1 one can see a visualization of these generators. We see that A_1 generates translations to the right, A_2 translations upwards, and A_3 counterclockwise rotations.

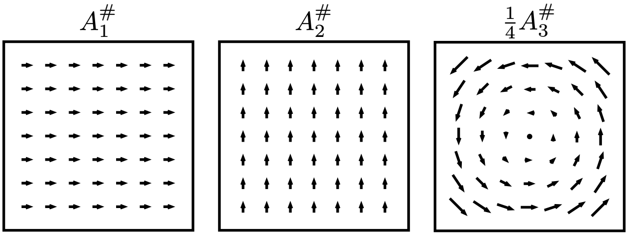


Figure 2.1: Visualization of the generators A_i by plotting their fundamental vector fields $A_i^\#$ on $\mathbb{R}^2$.

In two dimensions, we can parametrize all rotation matrices R by a rotation angle $\theta \in \mathbb{R}/(2\pi\mathbb{Z})$:

$$
R = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}. \tag{2.11}
$$

This allows us to efficiently parametrize SE(2) using three numbers:

$$
\mathbb{R}^2 \times [0, 2\pi) \ni (x, y, \theta) \leftrightarrow \left(\begin{pmatrix}x\\y\end{pmatrix}, \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}\right) \in \mathrm{SE}(2). \tag{2.12}
$$

So, we have that x and y represent the translation and θ the rotation. In these coordinates the group multiplication is

$$\begin{aligned}(x_2, y_2, \theta_2) \cdot (x_1, y_1, \theta_1) = (&x_2 + x_1 \cos \theta_2 - y_1 \sin \theta_2, \\ &y_2 + x_1 \sin \theta_2 + y_1 \cos \theta_2, \\ &\theta_2 + \theta_1 \bmod 2\pi).\end{aligned} \tag{2.13}$$

Remark 2.2: The *universal cover* Lie group $G = \widetilde{\mathrm{SE}(2)} = \{(x, y, \theta) \in \mathbb{R}^3\}$ of the Lie group SE(2) is a nice example of a Lie group that is diffeomorphic to three-dimensional Euclidean space that does *not* have a surjective exponential $\exp : \mathfrak{g} \to G$. Its group product is identical to (2.13) but *without* the modulo operation in the angular part. An element is *not* reachable by an exponential curve *if and only if* it is of the form $(x, y, 2\pi k)$ with $0 \neq k \in \mathbb{Z}$ and $(x, y) \neq 0$.

By considering derivatives of (2.12), we see that we can reexpress the basis of generators $A_i \in \mathfrak{se}(2)$ in terms of the (x, y, θ) coordinates, namely:

$$A_1 = \partial x|_e, \quad A_2 = \partial y|_e, \quad A_3 = \partial \theta|_e. \tag{2.14}$$

The Lie group exponential $\exp : \mathfrak{se}(2) \to \mathrm{SE}(2)$ in terms of the A_i basis and (x, y, θ) coordinates is

$$\exp(c^1 A_1 + c^2 A_2 + c^3 A_3) = (x, y, \theta) \tag{2.15}$$

where

$$\begin{aligned} x &= \left(c^1 \cos \frac{c^3}{2} - c^2 \sin \frac{c^3}{2}\right) \operatorname{sinc} \frac{c^3}{2}, \\ y &= \left(c^1 \sin \frac{c^3}{2} + c^2 \cos \frac{c^3}{2}\right) \operatorname{sinc} \frac{c^3}{2}, \\ \theta &= c^3 \bmod 2\pi, \end{aligned} \tag{2.16}$$

and $\operatorname{sinc}(\varphi) = \sin(\varphi)/\varphi$ (with $\operatorname{sinc}(0) = 1$) is the *unnormalized sinc function*.

The Lie group logarithm $\log : \mathrm{SE}(2) \to \mathfrak{se}(2)$ in this basis is

$$\log(x, y, \theta) = c^1 A_1 + c^2 A_2 + c^3 A_3 \tag{2.17}$$

where

$$\begin{aligned} c^1 &= \left(x \cos\frac{\theta}{2} + y \sin\frac{\theta}{2} \right) / \operatorname{sinc}\frac{\theta}{2}, \\ c^2 &= \left(-x \sin\frac{\theta}{2} + y \cos\frac{\theta}{2} \right) / \operatorname{sinc}\frac{\theta}{2}, \\ c^3 &= \theta. \end{aligned} \tag{2.18}$$

In this calculation it is important to use the small-angle identification $\mathbb{R}/(2\pi\mathbb{Z}) \equiv [-\pi, \pi)$ for θ.

2.2 POSITION-ORIENTATION SPACE

The *space of positions and orientations* $\mathbb{M}_d$ is defined as the manifold

$$\mathbb{M}_d = \{p = (x, n) \in \mathbb{R}^d \times \mathbb{R}^d \mid \|n\| = 1\}, \tag{2.19}$$

where x refers to the position and n to the orientation. The dimension is $\dim \mathbb{M}_d = d + (d - 1)$, that being d dimensions for the positions, and $d - 1$ dimensions for the orientations.

Remark 2.3: Position-orientation space $\mathbb{M}_d$ can also be defined as the *unit tangent bundle* or *unit sphere bundle* $UT\mathbb{R}^d$ of $\mathbb{R}^d$:

$$\mathbb{M}_d = UT\mathbb{R}^d := \{v \in T\mathbb{R}^d \mid \|v\| = 1\} \tag{2.20}$$

where we have equipped $\mathbb{R}^d$ with its natural Euclidean metric.

The tangent space $T_{(x,n)}\mathbb{M}_d$ at a position-orientation $(x, n) \in \mathbb{M}_d$ is

$$T_{(x,n)}\mathbb{M}_d = \{\dot{p} = (\dot{x}, \dot{n}) \in \mathbb{R}^d \times \mathbb{R}^d \mid \dot{n} \cdot n = 0\}, \tag{2.21}$$

and the complete tangent bundle $T\mathbb{M}_d$ is

$$T\mathbb{M}_d = \left\{(p, \dot{p}) \mid p \in \mathbb{M}_d, \dot{p} \in T_p\mathbb{M}_d\right\}. \tag{2.22}$$

The Euclidean group acts $\triangleright\colon \mathrm{E}(d) \times \mathbb{M}_d \to \mathbb{M}_d$ naturally on position-orientation space:

$$(t, Q) \rhd (x, n) = (t + Qx, Qn). \tag{2.23}$$

This action is transitive, thus turning position-orientation space into a homogeneous space. In fact, the special Euclidean group $\mathrm{SE}(d)$ already acts transitively as well.

By considering push-forwards of the action we get the following derived action $\rhd\colon \mathrm{E}(d) \times T\mathbb{M}_d \to T\mathbb{M}_d$:

$$(t, Q) \rhd ((x, n), (\dot{x}, \dot{n})) = ((t + Qx, Qn), (Q\dot{x}, Q\dot{n})). \tag{2.24}$$

The Lie group action also induces a Lie algebra action $\rhd\colon \mathfrak{e}(d) \times \mathbb{M}_d \to T\mathbb{M}_d$ given by

$$(v, \omega) \rhd (x, n) = ((x, n), (v + \omega x, \omega n)). \tag{2.25}$$

There is a natural $\mathrm{E}(d)$ invariant subbundle $H \subset T\mathbb{M}_d$ of the tangent bundle on position-orientation space. Namely, we define:

$$H \coloneqq \{((x, n), (\dot{x}, \dot{n})) \in T\mathbb{M}_d \mid \dot{x} \propto n\}. \tag{2.26}$$

That is, the subbundle H restricts spatial movement $\dot{x}$ to forward and backward with respect to the orientation n. This subbundle is bracket-generating and can be used to turn $\mathbb{M}_d$ into a sub-Riemannian manifold.

A commonly used and natural $\mathrm{E}(d)$ invariant Riemannian metric $\mathcal{G}$ on $\mathbb{M}_d$ has a corresponding norm $\|\cdot\|_{\mathcal{G}} : T\mathbb{M}_d \to \mathbb{R}_{\geq 0}$ of the form

$$\|(p, \dot{p})\|_{\mathcal{G}}^2 = (w_1 |\dot{x} \cdot n|)^2 + (w_2 \|\dot{x} \wedge n\|)^2 + (w_3 \|\dot{n}\|)^2, \tag{2.27}$$

where $p = (x, n) \in \mathbb{M}_d$, $\dot{p} = (\dot{x}, \dot{n}) \in T_p\mathbb{M}_d$, and $w_1, w_2, w_3 > 0$ are three positive numbers, called the *metric parameters*. Intuitively, the metric parameters w_1, w_2, and w_3 correspond to the cost of forwards/backwards spatial motion, sideways spatial movement, and changing the orientation, respectively.

The norm $\|\cdot\|_{\mathcal{G}}$ at every point can be verified to satisfy the *parallelogram law*:

$$\|v + w\|_{\mathcal{G}}^2 + \|v - w\|_{\mathcal{G}}^2 = 2\|v\|_{\mathcal{G}}^2 + 2\|w\|_{\mathcal{G}}^2. \tag{2.28}$$

This implies, as it should, that it is induced by an inner product. The corresponding Riemannian metric $\mathcal{G}$ can then be recovered using the *polarization identity*:

$$\mathcal{G}(v, w) = \frac{1}{4}\left(\|v + w\|_{\mathcal{G}}^2 - \|v - w\|_{\mathcal{G}}^2\right). \tag{2.29}$$

The metric (2.27) is called *spatially isotropic* if $w_1 = w_2 = w$, and *spatially anisotropic* otherwise. If it is spatially isotropic the norm can be equivalently written as

$$\|(p, \dot{p})\|_{\mathcal{G}}^2 = (w\|\dot{x}\|)^2 + (w_3\|\dot{n}\|)^2. \tag{2.30}$$

In the spatially isotropic case, the Riemannian distance can be expressed in closed form:

$$d((x_1, n_1), (x_2, n_2))^2 = (w\|x_2 - x_1\|)^2 + (w_3 \operatorname{acos} n_2 \cdot n_1)^2. \tag{2.31}$$

2.2.1 Two-Dimensional Position-Orientation Space

There is a natural $\mathrm{SE}(2)$ invariant frame $\mathcal{A}_i \in \Gamma(T\mathbb{M}_2)$ of vector fields on $\mathbb{M}_2$. Namely, we define

$$\begin{aligned}
(\mathcal{A}_1)_{(x,n)} &= (n, 0), \\
(\mathcal{A}_2)_{(x,n)} &= (n^\perp, 0), \\
(\mathcal{A}_3)_{(x,n)} &= (0, n^\perp),
\end{aligned} \tag{2.32}$$

where $n^\perp = \binom{-n_2}{n_1}$ is the ninety-degree counterclockwise rotation of the orientation $n = \binom{n_1}{n_2}$. Intuitively, $\mathcal{A}_1$ corresponds to moving x forwards along n, $\mathcal{A}_2$ to moving sideways to the left, and $\mathcal{A}_3$ to rotating n counterclockwise around x. In Figure 2.2 this frame is visualized.

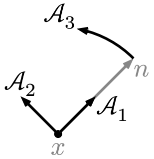


Figure 2.2: Visualization of the natural SE(2) invariant frame $\mathcal{A}_i$ at a position-orientation $(x, n) \in \mathbb{M}_2$.

The previously mentioned natural subbundle H (2.26) can be written in this frame as

$$H = \operatorname{span}\{\mathcal{A}_1, \mathcal{A}_3\}. \tag{2.33}$$

This should make sense: the subbundle H restricts spatial movement to forwards and backwards, that being the span of $\mathcal{A}_1$, and leaves orientation motion untouched, that being the span of $\mathcal{A}_3$.

One can restate the $\mathrm{E}(2)$ invariant Riemannian metric $\mathcal{G}$ we saw earlier (2.27) in terms of the frame $\mathcal{A}_i$. Namely, we have:

$$\mathcal{G}(\mathcal{A}_i, \mathcal{A}_j) = \begin{pmatrix} w_1^2 & 0 & 0 \\ 0 & w_2^2 & 0 \\ 0 & 0 & w_3^2 \end{pmatrix}_{ij}. \tag{2.34}$$

In two dimensions, we can parametrize all orientations $n \in S^1$ using an angle $\theta \in \mathbb{R}/(2\pi\mathbb{Z})$:

$$n = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}. \tag{2.35}$$

This allows us to efficiently parametrize $\mathbb{M}_2$ using three coordinates:

$$\mathbb{R}^2 \times [0, 2\pi) \ni (x, y, \theta) \leftrightarrow \left(\begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \right) \in \mathbb{M}_2. \tag{2.36}$$

So, we have that x and y represent the position and θ the orientation.

By considering derivatives of (2.36), we see that the coordinate vector fields ∂x, ∂y and $\partial\theta$ are

$$\begin{aligned}
\partial x_{(x,y,\theta)} &= \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, 0 \right), \\
\partial y_{(x,y,\theta)} &= \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, 0 \right), \\
\partial \theta_{(x,y,\theta)} &= \left(0, \begin{pmatrix} -\sin\theta \\ +\cos\theta \end{pmatrix} \right)
\end{aligned} \tag{2.37}$$

With this, we can equivalently write the natural frame $\mathcal{A}_i$ as:

$$\begin{aligned}
\mathcal{A}_1 &= +\cos(\theta)\partial x + \sin(\theta)\partial y, \\
\mathcal{A}_2 &= -\sin(\theta)\partial x + \cos(\theta)\partial y, \\
\mathcal{A}_3 &= \partial\theta,
\end{aligned} \tag{2.38}$$

which is a more common way to introduce this frame on $\mathbb{M}_2$.

The Riemannian distance of the spatially isotropic metric (2.31) on $\mathbb{M}_2$ can also be reexpressed using (x, y, θ) coordinates, namely:

$$\begin{aligned}
&d((x_1, y_1, \theta_1), (x_2, y_2, \theta_2))^2 = \\
&(w|x_2 - x_1|)^2 + (w|y_2 - y_1|)^2 + (w_3|\theta_2 - \theta_1 \bmod 2\pi|)^2,
\end{aligned} \tag{2.39}$$

where one has to make sure to use the small-angle identification $\mathbb{R}/(2\pi\mathbb{Z}) \equiv [-\pi, \pi)$ before calculating the absolute value of the angle difference.

Using (x, y, θ) coordinates we can visualize $\mathbb{M}_2$ in a new way. We imagine $\mathbb{R}^2 \times [0, 2\pi)$ as a three-dimensional volume, with the vertical axis being the θ coordinate. This is visualized in Figure 2.3. Although θ is plotted “linearly”, it is inherently periodic, so the top and bottom of the volume should be identified with each other. This visualization of $\mathbb{M}_2$ will be used extensively in Chapter 3.

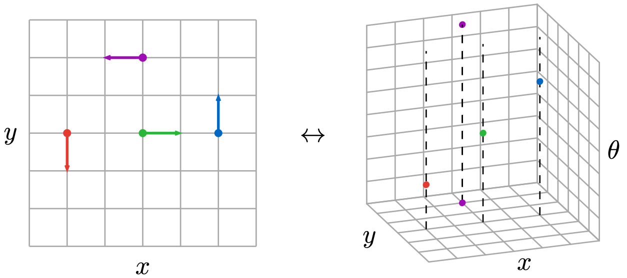


Figure 2.3: We can visualize position-orientations in two ways: 1) as arrows in the two-dimensional plane, 2) as points in a three-dimensional volume that is periodic vertically.

2.3 Lifting and Projecting

Data is usually defined on a low-dimensional space M, typically Euclidean space $\mathbb{R}^d$. However, geometric processing methods shine when the data is defined on high-dimensional spaces with which we can encode directional information, for example $\mathbb{M}_d$. The process of transferring data from a low-dimensional space to a higher dimensional one in an equivariant manner is called *lifting*, and the reverse process is called *projecting*.

In the specific case of lifting a scalar field on $\mathbb{R}^d$ to one on $\mathbb{M}_d$, the lifting can be done through what is called the *orientation score transform*. We imagine designing a *filter* or *kernel* $\psi : \mathbb{R}^d \to \mathbb{R}$ that can "pick up" an arbitrary local orientation $n_0 \in S^{d-1}$ at an arbitrary position $x_0 \in \mathbb{R}^d$ within a signal $f : \mathbb{R}^d \to \mathbb{R}$. We collect the position x_0 and orientation n_0 into a combined position-orientation $p_0 = (x_0, n_0) \in \mathbb{M}_d$. By "picking up" we mean that, if we calculate

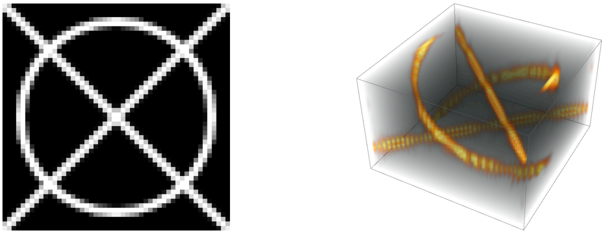
$$\langle \psi, f \rangle := \int_{\mathbb{R}^d} \psi(x) \cdot f(x) \, \mathrm{d}x, \tag{2.40}$$

we obtain (in absolute value) a large response if f indeed contains a local orientation in the direction of n_0 at x_0.

We then extend this to the whole of $\mathbb{M}_d$ in the following way:

$$(\mathcal{W}_\psi f)(g \triangleright p_0) := \langle g \triangleright \psi, f \rangle \text{ for all } g \in \mathrm{SE}(d). \tag{2.41}$$

This operator $\mathcal{W}_\psi$ is the orientation score transform, and $\mathcal{W}_\psi f : \mathbb{M}_d \to \mathbb{R}$ is called the *orientation score* of f. Intuitively, $\mathcal{W}_\psi$ takes the filter ψ, roto-translates it across the input f, and “saves” the response as a function on $\mathbb{M}_d$. in Figure 2.4 an image together with its orientation score is visualized.



An example input image $f : \mathbb{R}^2 \to \mathbb{R}$ with clearly defined local orientations.

Its orientation score $f : \mathbb{M}_2 \to \mathbb{R}$. Notice how the crossings are disentangled.

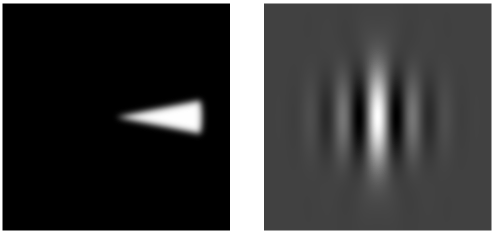
Figure 2.4: An example of a two-dimensional image together with its orientation score. We visualize $\mathbb{M}_2$ as a three-dimensional volume, as explained in Figure 2.3. The orientation scores of higher-dimensional images are harder to visualize.

One can imagine wanting the orientation score transform $\mathcal{W}_\psi$ to be (easily) *invertible* and *stable* (in the sense that the condition number of the operator is close to 1). Here, the trick is choosing the right function spaces and filters ψ, of which the design and analysis can get quite involved (Duits (2005), Duits, Duits, Almsick and ter Haar Romeny (2007), Duits, Felsberg, Granlund and ter Haar Romeny (2007)).

Historically, an early invertible orientation score transform (under the name *invertible orientation bundles*) using the so-called *Kalitzin kernel*, appears in Kalitzin, ter Haar Romeny and Viergever (1997) and Kalitzin, ter Haar Romeny and Viergever (1999).

Instead of considering all orientations on the (hyper)sphere S^{d-1}, one may restrict to a discrete set of orientations. In that case, a natural class of filters to use are the *cake wavelets*, originally proposed for $d = 2$ in

Duits (2005) and extended to $d = 3$ in Janssen, Duits and Breeuwer (2015) and Janssen, Janssen, Bekkers, Bescós and Duits (2018). In Figure 2.5 one such cake wavelet is visualized. More recently, Sherry, van de Geijn, Bekkers and Duits (2025) showed that cake wavelets provide a good approximation to the "optimal" wavelet, that being the one that minimizes uncertainty in both position and orientation.



Cake wavelet in the frequency domain.

Cake wavelet in the spatial domain.

Figure 2.5: Visualization of a cake wavelet that filters for vertical edges. In the frequency domain it becomes clear why cake wavelets are called such: they look like pieces of a round cake.

Projecting a scalar signal $f : \mathbb{M}_d \to \mathbb{R}$ on position-orientation space back to Euclidean space $\mathcal{P}f : \mathbb{R}^d \to \mathbb{R}$ can be done in multiple straightforward ways, generally based on *pooling* orientational information at every position. For example, one approach is *sum-pooling*, defined as

$$(\mathcal{P}f)(x) := \int_{S^{d-1}} f(x, n)\, \mu(\mathrm{d}n), \tag{2.42}$$

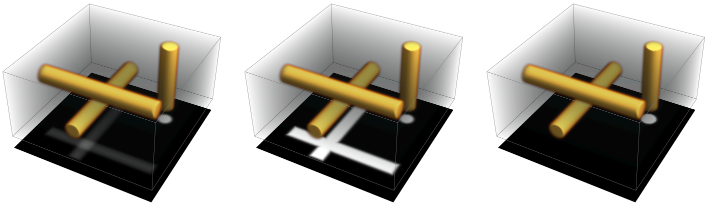
where μ is the standard measure on the (hyper)sphere S^{d-1}. Alternatively, there is *max-pooling*

$$(\mathcal{P}f)(x) := \max_{n \in S^{d-1}} f(x, n), \tag{2.43}$$

and, analogously, *min-pooling*

$$(\mathcal{P}f)(x) := \min_{n \in S^{d-1}} f(x, n). \tag{2.44}$$

In Figure 2.6 these pooling operations are visualized on $\mathbb{M}_2$.



Sum-pooling Max-pooling Min-pooling

Figure 2.6: Example of sum-, max-, and min-pooling over orientations for a synthetic scalar signal on $\mathbb{M}_2$. We visualize $\mathbb{M}_2$ as a three-dimensional volume, as explained in Figure 2.3. The result of the projection is visualized below the volume.

Remark 2.4: If a sum-pooling projection (2.42) after an orientation score transform lift is approximately (a scalar multiple of) the identity, that is

$$f(x) \approx \int_{S^{d-1}} (W_\psi f)(x, n) \mu(\mathrm{d}n), \tag{2.45}$$

one speaks of an orientation score transform with the *fast reconstruction* property (Duits (2005)).

We have discussed how to lift and project scalar signals from and to $\mathbb{R}^d$ and $\mathbb{M}_d$. However, sometimes data is vector-valued. In such cases, we can also naturally lift and project such data to and from scalar fields on $\mathbb{M}_d$.

For example, we can lift a vector field $f : \mathbb{R}^d \to \mathbb{R}^d$ to a scalar field $\mathcal{L}f : \mathbb{M}_d \to \mathbb{R}$ by

$$(\mathcal{L}f)(x, n) := f(x) \cdot n, \tag{2.46}$$

and project a scalar signal $f : \mathbb{M}_d \to \mathbb{R}$ back to a vector field $\mathcal{P}f : \mathbb{R}^d \to \mathbb{R}^d$ via

$$(\mathcal{P}f)(x) := \int_{S^{d-1}} f(x, n) n \, \mu(\mathrm{d}n), \tag{2.47}$$

where μ is the standard measure on the (hyper)sphere S^{d-1}. Furthermore, the operators $\mathcal{L}$ and $\mathcal{P}$ are each other's adjoint (after equipping the spaces of square-integrable scalar- and vector-valued functions on $\mathbb{R}^d$ with their standard inner products), and we have $\mathcal{P} \circ \mathcal{L} = \mu\left(S^{d-1}\right)/d$.

So, we can lift and project both scalar and vector-valued signals on $\mathbb{R}^d$ to and from scalar fields on $\mathbb{M}_d$. More generally, this idea extends to arbitrary fields $f : \mathbb{R}^d \to V$, where V is any finite-dimensional vector space transforming under a representation $\rho : \mathrm{O}(d) \to \mathrm{GL}(V)$, as also realized by Gasteiger, Becker and Günnemann (2021).

Namely, representation theory of $\mathrm{O}(d)$ implies that any such vector space V can be associated with finite-dimensional space of functions $\mathcal{H} \subset C^\infty\left(S^{d-1}, \mathbb{R}^m\right)$ on the (hyper)sphere S^{d-1}, for some $m \in \mathbb{N}$. Here $\mathrm{O}(d)$ acts naturally on $\mathcal{H}$ through translation of functions. This means that there exists an invertible equivariant linear map $\Phi : V \to \mathcal{H}$ with which we can lift a field $f : \mathbb{R}^d \to V$ to the scalar field(s) $\mathcal{L}f : \mathbb{M}_d \to \mathbb{R}^m$:

$$(\mathcal{L}f)(x, n) = (\Phi f(x))(n), \tag{2.48}$$

and, conversely, project scalar fields $f : \mathbb{M}_d \to \mathbb{R}^m$ back to $\mathcal{P}f : \mathbb{R}^d \to V$ via

$$(\mathcal{P}f)(x) = \Phi^{-1}(f(x, \cdot)). \tag{2.49}$$

2.4 SCALE-SPACE REPRESENTATION

Real world scenes contain many different objects at different scales. When a computer is tasked with analyzing an image of a scene there is no way for it to know beforehand at which scale(s) the interesting structures live. One way to tackle this problem is to analyze the image of interest at *all* scales.

In broad terms, a *scale-space representation*, or simply *scale-space*, of a 2D grayscale image $f_0 : \mathbb{R}^2 \to \mathbb{R}$ is an ordered collection of images f_t where each successive image contains less and less detail; that is the smaller scales have been processed away. The collection of images is usually indexed by the *scale-parameter* $t \geq 0$ with $t = 0$ being the original image. In the abstract ideal the scale parameter t is continuous and the scale-space ranges all the way from scale $t = 0$ to scales that are arbitrarily large. In practice a discrete set of scales t is chosen, usually in an exponential manner such as $t = 1, 4, 16, 64, 256$ (ter Haar Romeny (2003)).

Scale-space representations are natural objects in computer vision solutions (either neural networks or classical methods) as they respect the inherent symmetries of images, that being translations, rotational, and scaling symmetries. What we mean by this mathematically is that, for example, the scale-space g_t of a translated image $g_0 = T_v f_0$, is equal to the translated scale-space of the original image: $g_t = T_v f_t$. Here T_v is the translation operator defined by $(T_v f)(x) = f(x - v)$. Analogous statements hold for the rotation, reflection, and scaling symmetries. We say that creating the scale-space representation of an image is *equivariant* with respect to translations, rotations and scalings.

The prototypical, and most likely first (Weickert, Ishikawa and Imiya (1999)), example of a scale-space is the *Gaussian scale-space* (Iijima (1959), Witkin (1983), Koenderink (1984)) made by successive *diffusing* (that is *blurring* or *smoothing*) of the original image. The Gaussian scale-space f_t of a two-dimensional image $f_0 : \mathbb{R}^2 \to \mathbb{R}$ can be written as a linear convolution with a Gaussian kernel:

$$f_t(x) = \int_{\mathbb{R}^2} \frac{1}{2\pi t} \exp\left(-\frac{\|x - y\|^2}{2t} \right) f_0(y) \, \mathrm{d}y \tag{2.50}$$

The Gaussian scale-space corresponds to the *diffusion* or *heat* PDE

$$\frac{\partial f}{\partial t} = \frac{1}{2} \Delta f \tag{2.51}$$

Two other examples are the *morphological scale-space* representations (Brockett and Maragos (1992)) made by successively *dilating* or *eroding* the original image. The α*-dilation scale-space* can be written as a supremal convolution:

$$f_t(x) = \sup_{y \in \mathbb{R}^2} -\frac{t}{\beta}\left(\frac{\|x-y\|}{t}\right)^{\beta} + f_0(y) \tag{2.52}$$

where β is such that $1/\alpha + 1/\beta = 1$. It corresponds to viscosity solutions (Evans (2010), Chpt.10) of the α-*dilation* PDE:

$$\frac{\partial f}{\partial t} = +\frac{1}{\alpha}\|\nabla f\|^{\alpha}. \tag{2.53}$$

The α-*erosion scale-space* is created using an infimal convolution:

$$f_t(x) = \inf_{y \in \mathbb{R}^2} \frac{t}{\beta}\left(\frac{\|x-y\|}{t}\right)^{\beta} + f_0(y) \tag{2.54}$$

It corresponds to viscosity solutions of α-*erosion* PDE:

$$\frac{\partial f}{\partial t} = -\frac{1}{\alpha}\|\nabla f\|^{\alpha}. \tag{2.55}$$

In Figure 2.7 the Gaussian, quadratic ($\alpha = 2$) dilation, and quadratic erosion scale-spaces representations are visualized of an image of the DRIVE dataset (Subsection 2.8.1). In Figure 2.8 the α-erosion scale-space representation with $\alpha = \frac{8}{7}$ is shown. In the Gaussian scale-space both white and black features fade away towards a uniform image. In the dilation scale-space the black details (low values), such as the vessels, vanish at bigger scales. In the erosion scale-space the white details (high values), such as the space between vessels, are removed at higher scales

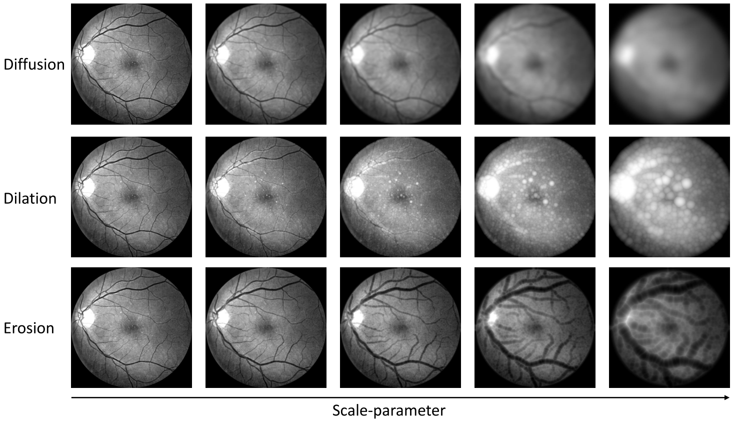


Figure 2.7: The Gaussian (2.50), quadratic ($\alpha = 2$) dilation (2.52), and quadratic erosion (2.54) scale-space representations of a grayscale image of the interior surface of the eye at various scale-parameters.

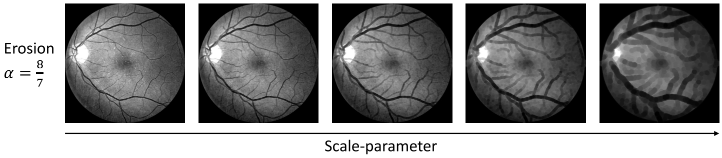


Figure 2.8: The α-erosion scale-space representation (2.54) with $\alpha = \frac{8}{7}$. This can be compared with the quadratic ($\alpha = 2$) erosion scale-space seen in Figure 2.7.

2.5 Machine Learning

Supervised Learning aims to find a function $m : X \to Y$, also called a *model*, that maps inputs to outputs in accordance with a finite set of example input-output pairs $\mathcal{T} \subset X \times Y$, known as the *training dataset*. Here, X is called the *input space* and Y the *output space*. For instance, x could be an image and y a label describing its content, so that X is the set of all images and Y the set of possible labels.

To measure the quality of a model m, we introduce a *loss function* $\ell : Y \times Y \to \mathbb{R}_{\geq 0}$, quantifying the discrepancy between predicted and true outputs. The *loss* of a model m on a dataset $\mathcal{D} \subset X \times Y$ is defined as:

$$L(m, \mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \ell(m(x), y). \tag{2.56}$$

The loss on the training dataset $L(m, \mathcal{T})$, called the *train loss*, is what we aim to minimize.

Remark 2.5: To keep the exposition simple, we deliberately avoid the probabilistic/statistical formulation of supervised learning here.

We restrict our search to a predesigned set of functions M, called the *hypothesis space*. The “best” model $m^* \in M$ is the one that has the lowest train loss:

$$m^* = \arg\min_{m \in M} L(m, \mathcal{T}). \tag{2.57}$$

Mind however, that the goal is not merely to fit the training dataset $\mathcal{T}$, but to find a model m that generalizes to inputs *not* in the training data. To check if a model has successfully generalized, its loss is checked on a separate dataset, the *test dataset*, which is held out entirely during training.

Typically, the hypothesis space is parametrized through a finite set of parameters $\theta \in \Theta$ restricted to some subset $\Theta \subseteq \mathbb{R}^p$ called *parameter space*. We denote the function corresponding to the parameters θ by m_θ. With this notation the train loss becomes a function of the parameters and we can rewrite the above optimization as:

$$\theta^* = \arg\min_{\theta \in \Theta} L(m_\theta, \mathcal{T}), \tag{2.58}$$

and the best model m^* is equal to m_{θ^*}. Thus, instead of considering an optimization over hypothesis space M, we consider an optimization over parameter space Θ.

How exactly the train loss is minimized over the parameter space, also referred to as *training*, falls outside the scope of this thesis. As of the time of writing, *stochastic gradient descent* (SGD) is the most popular way of training models. Moreover, not all parameters can be optimized via SGD, most notably when they are non-differentiable. Typically, any parameter left untouched by SGD are referred to as *hyperparameters* and need to be tuned “by hand”.

Loss functions. In the case of the output space being $Y = \mathbb{R}^n$, one popular and simple loss function is the *mean squared error* (MSE) $\ell(\hat{y}, y) := \frac{1}{n}\|\hat{y} - y\|^2$. Its derivative with respect to $\hat{y}$, which is necessary to perform SGD, is particularly simple: $\frac{\partial \ell}{\partial \hat{y}} = \frac{2}{m}(\hat{y} - y)$.

Another loss function we will see in Chapter 3 and Chapter 7 is the *Dice loss*

$$\ell(\hat{y}, y) = 1 - \operatorname{dice}(\hat{y}, y), \tag{2.59}$$

that being one minus the *Dice-Sørensen coefficient*, or just *Dice coefficient*, defined as:

$$\operatorname{dice}(\hat{y}, y) := \frac{2\hat{y} \cdot y}{\hat{y} \cdot \hat{y} + y \cdot y}. \tag{2.60}$$

In this case the output space is discrete binary vectors $Y = \{0, 1\}^m$, and the Dice coefficient measures how similar two binary vectors ($\hat{y}$ and y) are by comparing the size of their overlap ($\hat{y} \cdot y$) to the total number of elements in both vectors ($\hat{y} \cdot \hat{y} + y \cdot y$).

Equation (2.60), and therefore in turn (2.59), generalizes directly to vectors attaining values in the continuous interval $[0, 1]$, that is $Y = [0, 1]^m$. This turns the Dice loss into a differentiable function, which is necessary to perform SGD.

The Dice loss is a common loss function for image segmentation tasks (like in DRIVE, DCA1, and Lines datasets Section 2.8) because it is more effective than alternatives like MSE loss when the target segmentation is a small fraction of the total image.

Architectures in machine learning refer to the design of the hypothesis space M of models. Typical architectures are made through the composition of multiple *layers* l^i, each with their own parameters. This means that models $m \in M$ take the form:

$$m = l^L \circ l^{L-1} \circ \cdots \circ l^1. \tag{2.61}$$

The idea is that every layer transforms its input into an output that represents progressively more abstract information.

For instance, when the input space X consists of images, the first layer l^1 might extract low-level features such as edges, corners, or line segments. The second layer l^2 can then use these to detect more complex patterns, such as textures or object parts, and this process continues through the subsequent layers.

The intermediate representations passed between layers are called *features* or *feature maps*. Typically, there are many feature maps passed between layers, called *channels*, where each channel encodes a distinct learned feature.

Convolutional Neural Networks (CNNs) are equivariant architectures primarily built upon convolutions on a homogeneous space M on which a Lie group G acts. We adopt a broader perspective than is typical in the literature, where CNNs usually refer to the special case of linear convolutions on $M = G = \mathbb{R}^d$. Here, we generalize to arbitrary homogeneous spaces and Lie groups, and include not only linear but also morphological convolutions.

This architecture is also called a *homogeneous G-CNN* in Cohen (2021), Chpt.9, as it is a direct generalization of the *group equivariant CNN* (G-CNN) architecture introduced in Cohen and Welling (2016). Furthermore, we restrict our setting to scalar fields on homogeneous spaces and do not consider more general vector- or tensor-valued fields, as in, for example, Cohen, Geiger and Weiler (2019).

A layer in a convolutional network takes in C_{in} feature maps $f_i : M \to \mathbb{R}$ on M, and outputs C_{out} feature maps $g_j : M \to \mathbb{R}$ in an equivariant manner. The layer can involve linear, supremal, or infimal homogeneous convolutions, pointwise function application, and channel mixing. An example layer is

$$g_j(p) = b_j + \sum_{i=1}^{C_{\text{in}}} A_{ij}\, \sigma\left(\int_M k_i(p,q) f_i(q)\, \mu(\mathrm{d}q) \right), \tag{2.62}$$

where $k_i : M \times M \to \mathbb{R}$ are invariant kernels, σ a nonlinear *activation function*, $A \in \mathbb{R}^{C_{\text{in}} \times C_{\text{out}}}$ a *weight matrix*, and $b \in \mathbb{R}^{C_{\text{out}}}$ a *bias vector*. Typically, the weight matrix A, the bias vector b, and the (parametrized) kernels k_i are all learnable.

We can divide this layer into three distinct operations: 1) we perform C_{in} homogeneous convolutions with kernels k_i, 2) we apply a pointwise nonlinearity with the activation function σ, 3) we combine the channels pointwise in an affine manner with the matrix A and vector b. Figure 2.9 illustrates this layer.

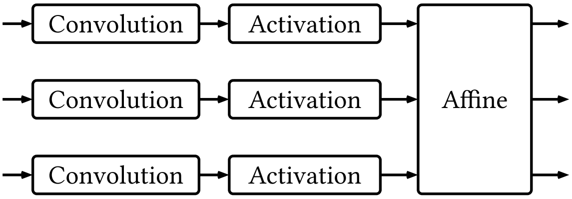


Figure 2.9: Diagram of a depthwise separable convolutional layer (2.62). The vertical direction represents the channels. The arrows represent the "flow" of the feature maps.

Because we convolve each input channel with a distinct kernel k_i, the convolutional part is known as a *depthwise convolution* in the machine learning literature. The affine operation is also called *pointwise convolution* or *channel mixing*. The full layer is referred to as *depthwise separable* (Chollet (2017)) as it performs a depthwise convolution followed by the affine mixing of channels (here, however, "interrupted" by an activation in between).

Depthwise separable should not be confused with *spatially separable*, which refers to a kernel $k : (M_1 \times M_2) \times (M_1 \times M_2) \to \mathbb{R}$, where M_1 and M_2 are two homogeneous spaces, which can be decomposed as

$$k(p_1, p_2, q_1, q_2) = k_1(p_1, q_1) k_2(p_2, q_2), \tag{2.63}$$

with $k_1 : M_1 \times M_1 \to \mathbb{R}$ and $k_2 : M_2 \times M_2 \to \mathbb{R}$.

2.6 PDE-based Neural Networks

In Smets, Portegies, Bekkers and Duits (2023) *PDE-based group equivariant convolution neural networks* (PDE-G-CNNs) were introduced. It is shown in Pai, Bellaard, Smets and Duits (2023), Bellaard, Bon, Pai, Smets and Duits (2023), Bellaard, Pai, Oliván Bescós and Duits (2023), and Bellaard, Sakata, Smets and Duits (2025) that, for medical image analysis problems, PDE-G-CNNs — in addition to being inherently equivariant — require fewer parameters, achieve higher accuracy, and are more data-efficient, in comparison to classical CNNs. From this perspective, PDE-G-CNNs can be preferable over other architectures in the aforementioned image processing tasks.

PDE-based networks build upon solvers of evolution PDEs defined on a homogeneous space M on which a Lie group G acts. The behavior of the PDEs depends on the Riemannian metric tensor field $\mathcal{G}$ defined on the space M. The Riemannian metrics $\mathcal{G}$ are designed to be invariant to the Lie group G, resulting in equivariant evolution. The parameters that determine the invariant metrics $\mathcal{G}$ are learned during training.

The evolution PDEs used are

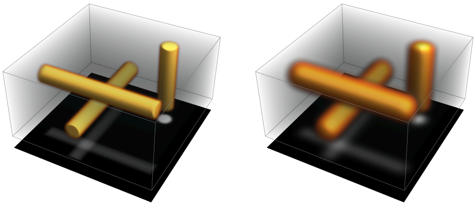
$$\text{Convection} \qquad \frac{\partial f}{\partial t} = -c \cdot \nabla f \tag{2.64}$$

$$\alpha\text{-Diffusion} \qquad \frac{\partial f}{\partial t} = -\frac{1}{\alpha}(-\Delta f)^{\frac{\alpha}{2}} \tag{2.65}$$

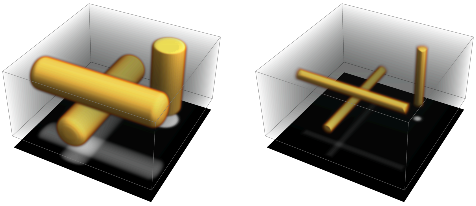
$$\alpha\text{-Dilation} \qquad \frac{\partial f}{\partial t} = \frac{1}{\alpha}\|\nabla f\|^{\alpha} \tag{2.66}$$

$$\alpha\text{-Erosion} \qquad \frac{\partial f}{\partial t} = -\frac{1}{\alpha}\|\nabla f\|^{\alpha} \tag{2.67}$$

Here $f : M \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ is some scalar field on M evolving over time $t \geq 0$, with $f(\cdot, 0) = f_0$ set to an initial condition. In the convection $c \in \Gamma(TM)$ denotes an invariant vector field, and in the diffusion $-(-\Delta)^{\frac{\alpha}{2}}$ denotes a (fractional) power of the Laplacian. Importantly, when using different Riemannian metrics $\mathcal{G}$, concepts such as Laplacian Δ, gradient ∇, and norm $\|\cdot\|$ change accordingly. Consequently, this alters the effect of the PDEs. The PDEs are illustrated in Figure 2.10.



Initial condition Diffused



Dilated Eroded

Figure 2.10: Illustration of diffusion (2.65), dilation (2.66), and erosion (2.67) applied to a synthetic signal on $\mathbb{M}_2$, for a spatially isotropic metric $\mathcal{G}$ (2.30). We visualize $\mathbb{M}_2$ as a three-dimensional volume, as explained in Figure 2.3. The sum-pooling projection (2.42) is visualized below the volume.

Remark 2.6: The convection PDE (2.64) is also called *advection*. However, we follow the terminology used in the PDE-based neural networks literature.

The PDE-G-CNN architecture is general in the sense that any Lie group G and homogeneous space M can be chosen. However, the existing literature mainly concerns itself with position-orientation space $M = \mathbb{M}_d$, together with the roto-translation group $G = \mathrm{SE}(d)$. Typically, input and output data are defined on a low-dimensional space while the PDE layers operate on a high-dimensional one. For this reason, PDE-based neural networks usually begin with a lifting layer and end with a projection layer (Section 2.3).

Figure 2.11 shows a diagram of a *PDE layer*, the fundamental building block of a PDE-G-CNN. Figure 2.12 shows a diagram of a PDE-based neural network architecture. Figure 2.13 shows an example of how a PDE-based neural network on $\mathbb{M}_2$ performs contour completion (Subsection 2.8.3).

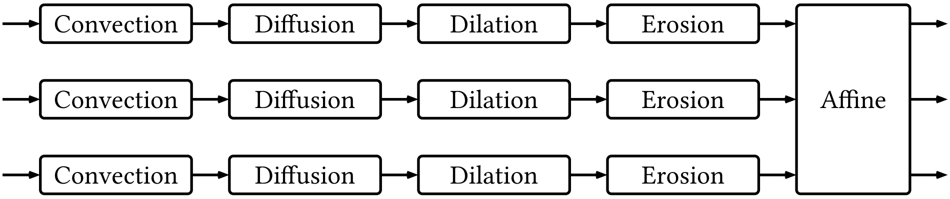


Figure 2.11: Diagram of a PDE layer.

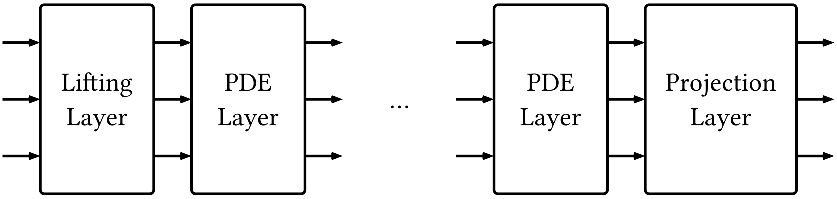


Figure 2.12: Diagram of a PDE-based neural network architecture.

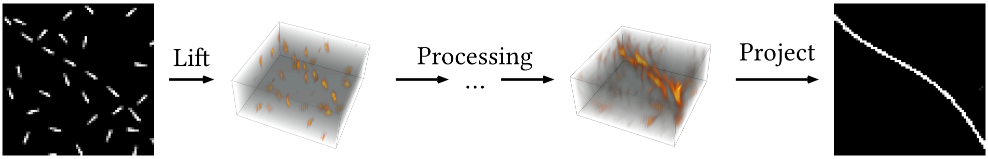


Figure 2.13: Example of how a PDE-based neural network on $\mathbb{M}_2$, when trained on the Lines dataset (Subsection 2.8.3), performs contour completion. We visualize $\mathbb{M}_2$ as a three-dimensional volume, as explained in Figure 2.3. The feature maps of a single channel are visualized.

The diffusion, dilation, and erosion PDEs were not chosen arbitrarily; they satisfy properties considered desirable from a machine learning perspective. For example, the PDEs are *quasilinear* and *equivariant* meaning that they 1) can be solved using convolutions allowing for fast parallel computation, and 2) allow for the design of inherently equivariant networks, resulting in an architecture that is robust and data-efficient (Mohamed, Cesa, Cohen and Welling (2020), Cohen, Weiler, Kicanaoglu and Welling (2019)). In fact, the desirable properties that we want the PDEs in PDE-based neural networks to have are essentially the same as those that generate scale-space representations (Section 2.4).

The diffusion, dilation, and erosion PDEs on M are (approximately) solved by performing a homogeneous convolution on the initial condition f_0 with a kernel $k : M \times M \to \mathbb{R}$ that is *only* a function of the Riemannian distance d. Namely, we have that the 2-diffusion PDE (2.65) is approximately solved by the linear convolution

$$f_t(p) \approx \int \exp\left(-\frac{1}{2}\frac{d(p,q)^2}{t}\right) \cdot f_0(q)\, \mu(\mathrm{d}q), \tag{2.68}$$

where we have used the notation $f_t = f(\cdot, t)$, and μ is the unique Riemannian measure corresponding to the Riemannian metric $\mathcal{G}$. The α-dilation PDE (2.66) is exactly solved through the supremal convolution

$$f_t(p) = \sup_{q \in M} -\frac{t}{\beta}\left(\frac{d(p,q)}{t}\right)^{\beta} + f_0(q), \tag{2.69}$$

where β is such that $1/\alpha + 1/\beta = 1$, and the α-erosion PDE (2.67) is solved using the infimal convolution

$$f_t(p) = \inf_{q \in M} \frac{t}{\beta}\left(\frac{d(p,q)}{t}\right)^{\beta} + f_0(q). \tag{2.70}$$

It is through these convolutions on the homogeneous space M that PDE-based neural networks are implemented in practice. Furthermore, one can compare these formulas directly with the scale-space representations on $\mathbb{R}^2$ as given in (2.50), (2.52) and (2.54).

An implementation of the PDE-G-CNN architecture, LieTorch, is available at the GitLab repository LieTorch. It provides both CPU and GPU solvers for the convection, 2-diffusion, α-dilation, and α-erosion PDEs on $\mathbb{R}^2$ and $\mathbb{M}_2$, and is designed as an extension to the popular machine learning library PyTorch.

2.7 PONITA

In Bekkers, Vadgama, Hesselink, van der Linden and Romero (2024) PONITA is presented, an $\mathrm{E}(3)$ equivariant machine learning architecture. It achieves state-of-the-art results when predicting molecular dynamics and properties, and in generating novel molecules with high atomic and molecular stability. PONITA is available at the GitHub repository PONITA.

The feature maps in PONITA are scalar fields $f : \mathbb{M}_3 \to \mathbb{R}$ on position-orientation space $\mathbb{M}_3$. This design decision is well supported as it can be shown (Section 2.3) that scalar fields on $\mathbb{M}_3$ provide the same expressivity as more mathematically involved networks that use $\rho : \mathrm{O}(3) \to \mathrm{GL}(V)$ representation fields $f : \mathbb{R}^3 \to V$ as done in, for example, Thomas, Smidt, Kearnes, Yang, Li, Kohlhoff and Riley (2018), Weiler, Geiger, Welling, Boomsma and Cohen (2018), Anderson, Hy and Kondor (2019), and Fuchs, Worrall, Fischer and Welling (2020).

The main operations in PONITA are $\mathrm{E}(3)$ equivariant convolution operators Φ on $\mathbb{M}_3$ of the form

$$(\Phi f)(p_1) = \int_{\mathbb{M}_3} k(p_1, p_2) f(p_2)\, \mu(\mathrm{d}p_2), \tag{2.71}$$

that process scalar fields $f : \mathbb{M}_3 \to \mathbb{R}$. Here $k : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$ is an $\mathrm{E}(3)$ invariant kernel, and μ is an invariant measure on position-orientation space $\mathbb{M}_3$.

In PONITA the kernels k are engineered to be of the form

$$k(p_1, p_2) = N_\theta(\iota_1(p_1, p_2), \iota_2(p_1, p_2), \iota_3(p_1, p_2)) \tag{2.72}$$

where $\iota : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$ are three $\mathrm{E}(3)$-invariants defined by:

$$\begin{aligned} \iota_1(p_1, p_2) &= (x_2 - x_1) \cdot n_1, \\ \iota_2(p_1, p_2) &= \|(x_2 - x_1) - \iota_1 n_1\|, \\ \iota_3(p_1, p_2) &= n_1 \cdot n_2, \end{aligned} \tag{2.73}$$

where $p_1 = (x_1, n_1), p_2 = (x_2, n_2) \in \mathbb{M}_3$, and $N_\theta : \mathbb{R}^3 \to \mathbb{R}$ is a small network with learnable parameters θ.

Figure 2.14 shows a diagram of a *PONITA layer* the fundamental building block of PONITA. Figure 2.15 shows a diagram of the PONITA architecture. In the PONITA architecture the lifting layer is also called the *embedding layer*, and the projection layer the *readout layer*.

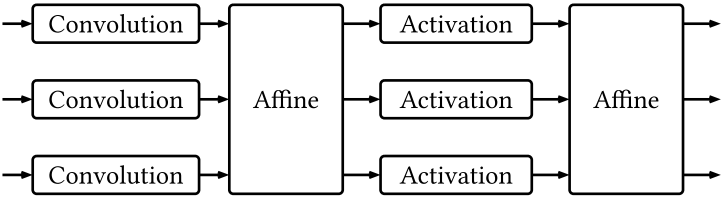


Figure 2.14: Diagram of a PONITA layer.

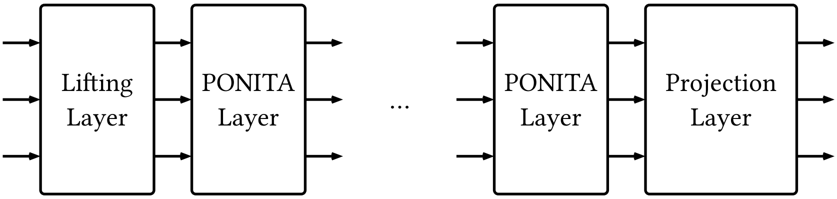


Figure 2.15: Diagram of the PONITA architecture.

2.8 DATASETS

2.8.1 DRIVE

The DRIVE dataset consists of images of the interior surface of the eye along with corresponding retinal vessel segmentations, as introduced in STAAL, ABRÀMOFF, NIEMEIJER, VIERGEVER AND VAN GINNEKEN (2004). In Figure 2.16 one can see an example of such an image and its segmentation.

The dataset consists of 40 images, which we split in a training set of 20 images and a test set of 20 images. The images are 584 × 565 pixels in 8-bit RGB color, which we rescale to 32-bit floats in the $[0, 1]$ range.

We divide the training set into overlapping patches of 64 × 64. Patches that contain no annotation are removed.

Input image of the interior surface of the eye.

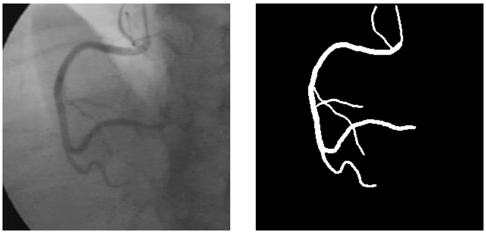
Target retinal vessel segmentation.

Figure 2.16: A sample from the DRIVE dataset.

2.8.2 DCA1

The DCA1 dataset, introduced in CERVANTES-SANCHEZ, CRUZ-ACEVES, HERNANDEZ-AGUIRRE, HERNANDEZ-GONZALEZ AND SOLORIO-MEZA (2019), is a publicly available database consisting of 130 X-ray coronary angiograms, and their corresponding ground-truth segmentations by an expert cardiologist. One such angiogram and ground-truth can be seen in Figure 2.17.

The images are 300×300 pixels in 8-bit grayscale, which we rescale to 32-bit floats in the $[0, 1]$ range. We split the DCA1 dataset into a training and test set consisting of 125 and 10 images, respectively. We divide the training set into overlapping patches of 64×64. Patches that contain no annotation are removed.



Input X-ray coronary angiogram

Target coronary artery segmentation

Figure 2.17: A sample from the DCA1 dataset.

2.8.3 LINES

The Lines dataset is a “toy” collection of synthetic images made to check if a machine learning model can learn to perform *contour completion* (FIELD, HAYES AND HESS (1993)). In the contour completion problem, the input is an image containing multiple randomly oriented line segments, and the desired output is an image of the contour that is “hidden” in the input image. Figure 2.18 shows an example of such an input and desired output.

The images are 64×64 pixels in 8-bit grayscale, which we rescale to 32-bit floats in the $[0, 1]$ range. The dataset contains 512 training images and 128 test images.

Input image

Target contour completion

Figure 2.18: A sample of the Lines dataset.

2.8.4 QM9

The QM9 dataset consists of roughly 134,000 stable small organic molecules composed of carbon, hydrogen, oxygen, nitrogen, and fluorine, along with their various chemical properties, as detailed in Ramakrishnan, Dral, Rupp and Lilienfeld (2014). In Table 2.1 a list of the chemical properties together with their description is given. The molecules contained in QM9 form a subset of the GDB-17 chemical universe of 166 billion organic molecules, as described in Ruddigkeit, Deursen, Blum and Reymond (2012).

Target	Unit	Description
μ	D	Dipole moment
α	a_0^3	Isotropic polarizability
$\varepsilon_{\text{homo}}$	eV	Highest occupied molecular orbital energy
$\varepsilon_{\text{lumo}}$	eV	Lowest unoccupied molecular orbital energy
$\Delta\varepsilon$	eV	Gap between $\varepsilon_{\text{homo}}$ and $\varepsilon_{\text{lumo}}$
$\langle R^2 \rangle$	eV	Electronic spatial extent
ZPVE	eV	Zero point vibrational energy
U_0	eV	Internal energy at 0 Kelvin
U	eV	Internal energy at 25 °Celsius
H	eV	Enthalpy at 25 °Celsius
G	eV	Free energy at 25 °Celsius
c_v	$\frac{\text{cal}}{\text{mol K}}$	Heat capacity at 25 °Celsius

Table 2.1: Overview of the chemical properties in the QM9 dataset.

DISTANCE APPROXIMATIONS ON POSITION-ORIENTATION SPACE 3

3.1 INTRODUCTION

The Riemannian distance d on two-dimensional position-orientation space $\mathbb{M}_2$ appears naturally in works that perform geometrically meaningful processing of signals living on $\mathbb{M}_2$. For example, in PDE-based neural networks on $\mathbb{M}_2$, the Riemannian distance d is used to effectively solve the diffusion, dilation and erosion PDEs (Section 2.6).

However, calculating the distance d on $\mathbb{M}_2$ is, in general, computationally expensive (BEKKERS, DUITS, MASHTAKOV AND SANGUINETTI (2015), DUITS, MEESTERS, MIREBEAU AND PORTEGIES (2018)). To alleviate this issue, we resort to estimating and substituting the exact distance d with computationally efficient *approximative distances*, denoted throughout by ρ. In this chapter we introduce and analyze various distance approximations on $\mathbb{M}_2$, both theoretically and experimentally.

We will focus on the following $\mathrm{E}(2)$ invariant metric $\mathcal{G}$ on $\mathbb{M}_2$, as this is the one that is most commonly used in the related literature:

$$\|(p,\dot{p})\|_{\mathcal{G}}^2 = (w_1|\dot{x}\cdot n|)^2 + (w_2\|\dot{x}\wedge n\|)^2 + (w_3\|\dot{n}\|)^2, \tag{3.1}$$

where $p=(x,n)\in\mathbb{M}_2$ and $\dot{p}=(\dot{x},\dot{n})\in T_p\mathbb{M}_2$. An important quantity in the analysis is the *spatial anisotropy* $\zeta = w_2/w_1$, as will become clear later. Furthermore, we will assume, without loss of generality, that $w_2 \geq w_1$, which we can understand as stating that sideways motion is at least as expensive as forward motion.

Because we are considering $\mathrm{E}(2)$ invariant metrics, and $\mathrm{E}(2)$ is a group that acts transitively on $\mathbb{M}_2$, we can, without loss of generality, restrict ourselves to analyzing and approximating $d(p_0,\cdot)$, instead of the full two-slot function $d(\cdot,\cdot)$, where $p_0\in\mathbb{M}_2$ is *any* reference position-orientation. We will choose $p_0=(0,0,0)$ as our reference, where we have used (x, y, θ) coordinates for $\mathbb{M}_2$, as defined in (2.36). Henceforth, $d(p)$ will be shorthand for $d(p_0, p)$, likewise for the approximations ρ.

In Smets, Portegies, Bekkers and Duits (2023) the *logarithmic distance approximation* ρ_c is suggested as a computationally efficient alternative to the Riemannian distance d. In short, $\rho_c(p)$ is equal to the Riemannian length of the unique projected exponential curve that connects p_0 to p. The exact definition will follow later in (3.8).

Despite showing in Smets, Portegies, Bekkers and Duits (2023) that $d \leq \rho_c$ no concrete bounds are given, apart from the asymptotic $\rho_c^2 \leq d^2 + \mathcal{O}(d^4)$. This motivates us to do a more in-depth analysis on the quality of the distance approximations.

We introduce a variation on the logarithmic approximation ρ_c called the *half-angle distance approximation* ρ_b, and analyze it. The half-angle approximation uses not the logarithmic coordinates c^i, but the half-angle coordinates b^i. The definition of these is also given later (3.19). In practice ρ_c and ρ_b do not differ much, but analyzing ρ_b is much easier.

Both the logarithmic approximation ρ_c and half-angle approximation ρ_b estimate the true Riemannian distance d quite well in certain cases. One of these cases is when the Riemannian metric has a low spatial anisotropy ζ. We can show this visually by comparing the isocontours of the exact and approximative distances. However, interpreting and comparing these surfaces can be difficult. This is why we additionally plot multiple θ-isocontours of these surfaces. In Figure 3.1 one such plot can be seen, and illustrates how it must be interpreted.

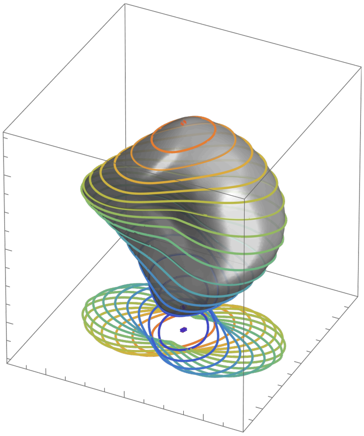


Figure 3.1: In gray the isocontour $d = 2.5$ is plotted. The metric parameters are $(w_1, w_2, w_3) = (1, 8, 1)$. We visualize $\mathbb{M}_2$ as a three-dimensional volume, as explained in Figure 2.3. The domain of the plot is $[-3, 3]^2 \times [-\pi, \pi) \subset \mathbb{M}_2$. For $\theta = k\pi/10$ with $k = -10, \ldots, 10$ isocontours are drawn and projected onto the bottom of the figure.

In Figure 3.2 a spatially isotropic $\zeta = 1$ and low-anisotropic case $\zeta = 2$ is visualized. We see that ρ_b approximates d well in these cases. In fact, ρ_b is exactly equal to the true distance d in the spatially isotropic case, which *is not true* for ρ_c.

Both the logarithm and half-angle approximation fail specifically in the high spatial anisotropy regime. For example when $\zeta = 8$. The first two columns of Figure 3.3 show that, indeed, ρ_b is no longer a good approximation of the exact distance d. For this reason we introduce a novel *sub-Riemannian* distance approximations $\rho_{b,\text{sr}}$, which is visualized in the third column of Figure 3.3.

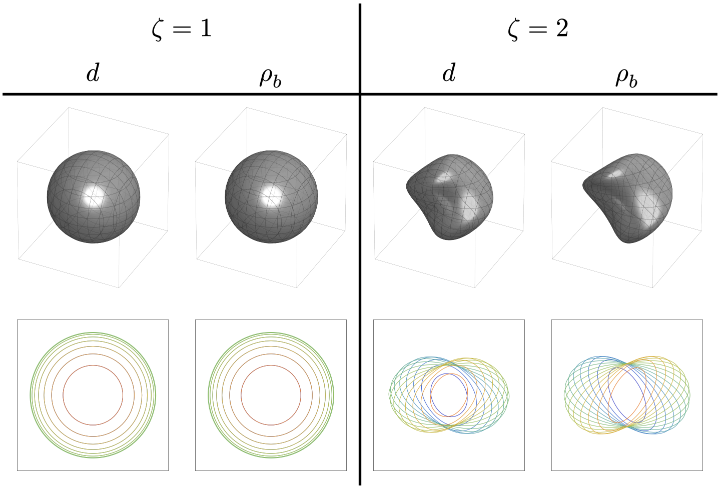


Figure 3.2: Visualization of the exact distance d and approximative distance ρ_b. The metric parameters are on the left $(w_1, w_2, w_3) = (1, 1, 1)$ and on the right $(w_1, w_2, w_3) = (1, 2, 1)$.

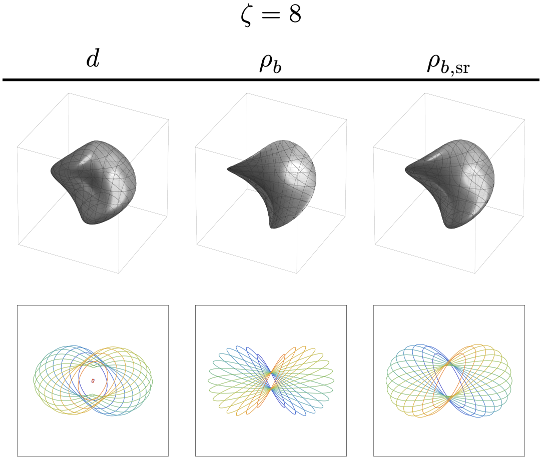


Figure 3.3: Visualization of the exact distance d, the approximative distance ρ_b, and the sub-Riemannian distance approximation $\rho_{b,\mathrm{sr}}$. The metric parameters are $(w_1, w_2, w_3) = (1, 8, 1)$. We see that the isocontours of ρ_b are too "thin" compared to the isocontours of d. The isocontours of $\rho_{b,\mathrm{sr}}$ are better in this respect.

Furthermore, we propose an approximative distance $\rho_{\mathrm{b,com}}$ that carefully combines the Riemannian and sub-Riemannian approximations into one. This combined approximation automatically switches to the estimate that is more appropriate depending on the spatial anisotropy, and hence covers both the low and high anisotropy regimes.

For every distance approximation (listed in Section 3.2) we perform an empirical analysis in Section 3.4 by seeing how the estimate changes the accuracy of the PDE-G-CNNs (Section 2.6) when applied to two datasets: Lines (Subsection 2.8.3) and DCA1 (Subsection 2.8.2).

3.1.1 CONTRIBUTIONS

Theorem 3.1 summarizes our mathematical analysis of the quality of the half-angle distance approximation ρ_b. Globally, one can show that it carries the same symmetries as the exact distance d, and that for low spatial anisotropies ζ they are almost indistinguishable. Furthermore, we show that locally the half-angle distance approximation ρ_b and the exact distance d are similar through an upper bound on the relative error. This improves upon results in SMETS, PORTEGIES, BEKKERS AND DUITS (2023), Lem.20.

Figure 3.3 demonstrates qualitatively that ρ_b becomes a poor approximation when the spatial anisotropy is high $\zeta \gg 1$. In Corollary 3.2 we underpin this theoretically and in Subsection 3.4.1 we validate this observation numerically. This motivates the use of a sub-Riemannian approximation when ζ is large.

In Section 3.2 we introduce and derive a novel sub-Riemannian distance approximation $\rho_{\mathrm{b,sr}}$, that overcomes difficulties in the high spatial anisotropy cases. Subsequently, we propose our approximation $\rho_{\mathrm{b,com}}$ that combines the Riemannian and sub-Riemannian approximations into one that automatically switches to the approximation that is more appropriate depending on the metric parameters.

Figure 3.11 and Figure 3.14 show that PDE-G-CNNs perform just as well as, and sometimes better than, G-CNNs and CNNs on the DCA1 and Lines dataset, while having the least amount of parameters. Figure 3.12 and Figure 3.15 depict an evaluation of the accuracy of PDE-G-CNNs when using the different distance approximations, again on the DCA1 and Lines dataset. We observe that our approximation $\rho_{\text{b,com}}$ provides best results.

3.1.2 OUTLINE

- In Section 3.2 all approximative distances are introduced and motivated.
- In Section 3.3 the main theorem can be found, which quantifies the quality of the approximative distances.
- In Section 3.4 we experiment with the various approximative distances within the PDE-G-CNN architecture.
- In Section 3.5 we conclude this chapter.

3.2 DISTANCE APPROXIMATIONS

Calculating the exact Riemannian distance $d(p_0, p)$, from the reference position-orientation $p_0 = (0, 0, 0)$ to some other point $p = (x, y, \theta)$, is generally computationally demanding for arbitrary Riemannian metrics $\mathcal{G}$. To alleviate this problem we approximate the exact distance d with approximative distances, denoted with ρ, which are computationally cheap.

To this end, we define the *logarithmic distance approximation* ρ_c, as explained in SMETS, PORTEGIES, BEKKERS AND DUITS (2023), Def.19, as the length L of a specific curve $\eta : [0, 1] \to \mathbb{M}_2$:

$$\rho_c(p) = L(\eta), \tag{3.2}$$

where η is the unique projected exponential curve $\eta(t) = \exp(tX)p_0$ that connects $p_0 = \eta(0)$ to $p = \exp(X)p_0 = \eta(1)$, with $X \in \mathfrak{se}(2)$.

By the invariance of the metric $\mathcal{G}$ and properties of the curve η we can simplify the length considerably:

$$\begin{aligned} L(\eta) &= \int_0^1 \|\dot{\eta}(t)\| \,\mathrm{d}t = \int_0^1 \|\exp(tX)Xp_0\| \,\mathrm{d}t \\ &= \int_0^1 \|Xp_0\| \,\mathrm{d}t = \|Xp_0\|. \end{aligned} \tag{3.3}$$

With knowledge of the SE(2) Lie group logarithm (2.17), the formula $\exp(X)p_0 = p$ can be solved for the generator X. Namely, one finds that

$$X = c^1 A_1 + c^2 A_2 + c^3 A_3, \tag{3.4}$$

where A_i is the basis of generators of $\mathfrak{se}(2)$ as defined in (2.10), and $c^i = c^i(p)$ given by

$$\begin{aligned} c^1 &= \left(x \cos\frac{\theta}{2} + y \sin\frac{\theta}{2} \right) / \operatorname{sinc}\frac{\theta}{2}, \\ c^2 &= \left(-x \sin\frac{\theta}{2} + y \cos\frac{\theta}{2} \right) / \operatorname{sinc}\frac{\theta}{2}, \\ c^3 &= \theta, \end{aligned} \tag{3.5}$$

called the *logarithmic coordinates* on $\mathbb{M}_2$. In the calculation of the logarithmic coordinates it is important to use the small-angle identification $\mathbb{R}/(2\pi\mathbb{Z}) \equiv [-\pi, \pi)$ for θ.

Furthermore, the basis of generators A_i applied to p_0 gives

$$\begin{aligned} Xp_0 &= (c^1 A_1 + c^2 A_2 + c^3 A_3) p_0 \\ &= \left(\begin{pmatrix} c^1 \\ c^2 \end{pmatrix}, \begin{pmatrix} -c^3 \\ 0 \end{pmatrix} \right) \in T_{p_0}\mathbb{M}_2, \end{aligned} \tag{3.6}$$

which, when plugged into the metric (3.1), gives:

$$\|Xp_0\| = \sqrt{(w_1 c^1)^2 + (w_2 c^2)^2 + (w_3 c^3)^2}. \tag{3.7}$$

Combining these results allows us to simplify the logarithmic distance approximation ρ_c down to

$$\rho_c = \sqrt{(w_1 c^1)^2 + (w_2 c^2)^2 + (w_3 c^3)^2}. \tag{3.8}$$

As already stated in the introduction, distance approximations such as ρ_c begin to fail in the high spatial anisotropy cases $\zeta \gg 1$. For these situations we need sub-Riemannian distance approximations. In previous literature two such sub-Riemannian approximations have been suggested. The first one is standard (TER ELST AND ROBINSON (1998), Sec.6), and the second one is a modified smooth version, as described in DUITS AND FRANKEN (2010), p.284. The second approximation also appears in BEKKERS, CHEN AND PORTEGIES (2018), eq.14, where $\nu \approx 44$ is empirically suggested:

$$\sqrt{\sqrt{\nu w_1^2 w_3^2 |c^2|} + (w_1 c^1)^2 + (w_3 c^3)^2}, \tag{3.9}$$

$$\sqrt[4]{\nu w_1^2 w_3^2 |c^2|^2 + \left((w_1 c^1)^2 + (w_3 c^3)^2\right)^2}. \tag{3.10}$$

Note that the sub-Riemannian approximations rely on the assumption that $w_2 \geq w_1$.

However, they both suffer from a major shortcoming in the interaction between w_3 and c^2. When we let $w_3 \to 0$, both approximations suggest that it becomes arbitrarily cheap to move in the c^2 direction. This is undesirable as this deviates from the exact distance d: moving spatially will always have a cost associated with it determined by at least w_1.

To make a proper sub-Riemannian distance estimate we will use a formula derived from the *Baker–Campbell–Hausdorff formula*, namely the *Zassenhaus formula*:

$$e^{t(X+Y)} = e^{tX} \, e^{tY} \, e^{-\frac{t^2}{2}[X,Y]} \, e^{\mathcal{O}(t^3)}. \tag{3.11}$$

Filling in $X = \mathcal{A}_1$ and $Y = \mathcal{A}_3$ (2.32) and neglecting the higher order terms gives:

$$e^{t(\mathcal{A}_1+\mathcal{A}_3)} \approx e^{t\mathcal{A}_1} \, e^{t\mathcal{A}_3} \, e^{\frac{t^2}{2}\mathcal{A}_2}, \tag{3.12}$$

or equivalently:

$$e^{\frac{t^2}{2}\mathcal{A}_2} \approx e^{-t\mathcal{A}_3} \, e^{-t\mathcal{A}_1} \, e^{t(\mathcal{A}_1+\mathcal{A}_3)}. \tag{3.13}$$

Here we interpret the exponential e^{tX} of a vector field $X \in \Gamma(TM)$ as being the associated flow map.

This formula says that one can successively follow the “legal” directions $\mathcal{A}_1$ and $\mathcal{A}_3$ to effectively move in the “illegal” direction of $\mathcal{A}_2$. Taking the lengths of these curves and adding them up gives an approximative upper bound on the sub-Riemannian distance:

$$\begin{aligned} d_{\text{sr}}\left(p_0, e^{\frac{t^2}{2}\mathcal{A}_2} p_0\right) &\lessapprox \left(w_1 + w_3 + \sqrt{w_1^2 + w_3^2}\right)|t| \\ &\leq 2(w_1 + w_3)|t|. \end{aligned} \tag{3.14}$$

Substituting $t \mapsto \sqrt{2|t|}$ gives:

$$d_{\text{sr}}(p_0, e^{t\mathcal{A}_2} p_0) \lessapprox 2\sqrt{2}(w_1 + w_3)\sqrt{|t|}. \tag{3.15}$$

This inequality, together with the smoothing trick to go from (3.9) to (3.10), inspires then the following *sub-Riemannian distance approximation*:

$$\rho_{c,\text{sr}} := \sqrt[4]{(\nu(w_1 + w_3))^4 |c^2|^2 + \left((w_1 c^1)^2 + (w_3 c^3)^2\right)^2} \tag{3.16}$$

for some $0 < \nu < 2\sqrt{2}$ such that the approximation is tight. We empirically suggest $\nu \approx 1.6$, based on a numerical analysis that is tangential to the one performed in Bekkers, Chen and Portegies (2018), Fig.3. Notice that this approximation does not break down when we let $w_3 \to 0$.

Figure 3.4 shows that both the old sub-Riemannian approximation (3.10) and new approximation (3.16) are appropriate in cases such as $w_3 = 1$. Figure 3.5 shows that the old approximation breaks down when we take $w_3 = 0.5$, and that the new approximation behaves more appropriate.

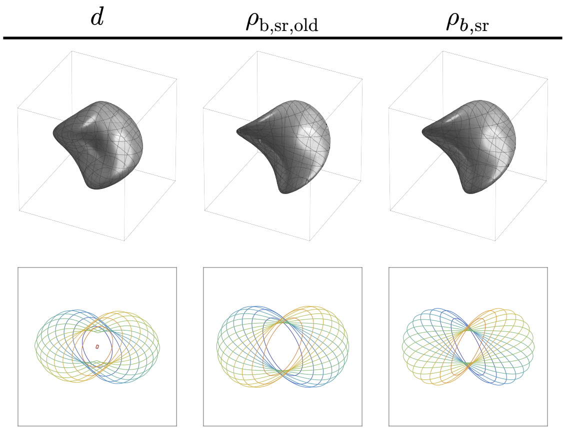


Figure 3.4: Visualization of the exact distance d, the old sub-Riemannian approximation $\rho_{\mathrm{b,sr,old}}$ (3.10), and the new approximation $\rho_{b,\mathrm{sr}}$ (3.16). The metric parameters are $(w_1, w_2, w_3) = (1, 8, 1)$. For the old approximation we chose $\nu = 44$, and for the new one $\nu = 1.6$. We see that in this case both approximations are appropriate.

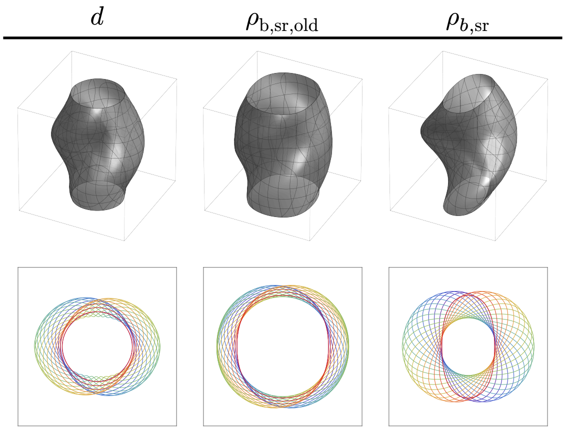


Figure 3.5: Same as Figure 3.4 but then with $(w_1, w_2, w_3) = (1, 8, 0.5)$. In this case the old sub-Riemannian approximation $\rho_{\mathrm{b,sr,old}}$ (3.10) underestimates the true distance and becomes less appropriate. The new approximation $\rho_{\mathrm{b,sr}}$ (3.16) is also not perfect but qualitatively better. Decreasing w_3 would exaggerate this effect even further.

The Riemannian and sub-Riemannian approximations can be combined into the following newly proposed practical approximation:

$$\rho_{c,\mathrm{com}} := \max\bigl(l, \min\bigl(\rho_{c,\mathrm{sr}}, \rho_c\bigr)\bigr), \tag{3.17}$$

where $l : \mathbb{M}_2 \to \mathbb{R}$ is given by:

$$l := \sqrt{\left(w_1 x\right)^2 + \left(w_1 y\right)^2 + \left(w_3 \theta\right)^2}, \tag{3.18}$$

for which we will show that it is a lower bound of the exact distance d in Lemma 3.4.

The most important property of the combined approximation is that it automatically switches between the Riemannian and sub-Riemannian approximations depending on the metric parameters. Namely, the Riemannian approximation is appropriate very close to the reference point p_0, but tends to overestimate the true distance at a moderate distance from it. The sub-Riemannian approximation is appropriate at moderate distances from p_0, but tends to overestimate very close to it, and underestimate far away. The combined approximation is such that we get rid of the weaknesses that the approximations have on their own.

On top of these approximative distances, we also define ρ_b, $\rho_{b,\mathrm{sr}}$, and $\rho_{b,\mathrm{com}}$ by replacing the logarithmic coordinates c^i with their corresponding *half-angle coordinates* b^i defined by:

$$\begin{aligned} b^1 &= x \cos\frac{\theta}{2} + y \sin\frac{\theta}{2}, \\ b^2 &= -x \sin\frac{\theta}{2} + y \cos\frac{\theta}{2}, \\ b^3 &= \theta. \end{aligned} \tag{3.19}$$

So, for example, we define the *half-angle distance approximation* ρ_b as:

$$\rho_b := \sqrt{\left(w_1 b^1\right)^2 + \left(w_2 b^2\right)^2 + \left(w_3 b^3\right)^2}. \tag{3.20}$$

Why we use the half-angle coordinates will be explained in Subsection 3.3.1.

3.3 MAIN THEOREM AND ANALYSIS

When an approximate distance, such as the half-angle approximation ρ_b, is used in place of the true Riemannian distance d, an error is introduced. The following theorem summarizes our analysis of the quality of the approximative distances.

Theorem 3.1 (Quality of approximative distance): Let $\zeta := w_2/w_1$ denote the spatial anisotropy. We assess the quality of our approximative distances in three ways:

- The exact and all approximative distances have the same symmetries, see Table 3.1.
- Globally it holds that:

  $$\frac{1}{\zeta}d \leq \rho_b \leq \zeta d, \tag{3.21}$$

  from which it directly follows that when $\zeta = 1$ we have $\rho_b = d$.
- Locally around p_0 we have:

  $$\rho_b^2 \leq (1+\varepsilon)d^2, \tag{3.22}$$

  where

  $$\varepsilon := \frac{\zeta^2 - 1}{2w_3^2}\zeta^4\rho_b^2 + C|\theta|^3. \tag{3.23}$$

Proof: The proof of the parts of the theorem will be discussed throughout the upcoming subsections.

- The symmetries are shown in Lemma 3.3.
- The global bound is shown in Corollary 3.1.
- The local bound is shown in Lemma 3.7.

$\square$

In SMETS, PORTEGIES, BEKKERS AND DUITS (2023), Lem.20 the bound $d = d(p_0, \cdot) \leq \rho_c$ is proven. Furthermore, it is shown that around p_0 one has:

$$\rho_c^2 \leq d^2 + \mathcal{O}(d^4), \tag{3.24}$$

which has the corollary that there exists a constant $C \geq 1$ such that

$$\rho_c \leq Cd \tag{3.25}$$

for any compact neighborhood around p_0. We improve on these results by;

- Showing that the approximative distances have the same symmetries as the exact Riemannian distance; Lemma 3.3.
- Finding simple global bounds on the exact distance d which can then be used to find global estimates of ρ_b in terms of d; Lemma 3.4. This improves upon (3.25) by finding an expression for the constant C.
- Estimating the leading term of the asymptotic expansion, and observing that our upper bound of the relative error between ρ_b and d explodes in the cases $\zeta \to \infty$ and $w_3 \to 0$; Lemma 3.7. This improves upon (3.24).

Note, however, that we are *not* analyzing ρ_c (3.8): we will be analyzing ρ_b (3.20). This is mainly because the half-angle coordinates are easier to work with: they do not have the sinc $\frac{\theta}{2}$ factor that he logarithmic coordinates have.

Using

$$\begin{aligned} b^1 &= c^1 \operatorname{sinc} \frac{\theta}{2}, \\ b^2 &= c^2 \operatorname{sinc} \frac{\theta}{2}, \\ b^3 &= c^3, \end{aligned} \tag{3.26}$$

recall (3.19) and (3.5), we see that

$$\operatorname{sinc} \frac{\theta}{2}\, \rho_c \leq \rho_b \leq \rho_c, \tag{3.27}$$

and thus locally ρ_c and ρ_b do not differ much. This means that bounds on ρ_b can be easily transferred to (slightly weaker) bounds on ρ_c.

3.3.1 Symmetry Preservation

Symmetries play a major role in the analysis of (sub-)Riemannian geodesics/distance in $\mathbb{M}_2$. They help to analyze symmetries in Hamiltonian flows (Moiseev and Sachkov (2010)) and corresponding symmetries in association field models (Duits, Boscain, Rossi and Sachkov (2014), Fig.11). There are together eight of them and their relation with logarithmic coordinates c^i (Lemma 3.1) shows they correspond to inversion of the Lie algebra basis $A_i \mapsto -A_i$ (2.10). They are

$$\begin{aligned}
\varepsilon^0(x,y,\theta) &= (x,y,\theta),\\
\varepsilon^1(x,y,\theta) &= (x\cos\theta + y\sin\theta, x\sin\theta - y\cos\theta, \theta),\\
\varepsilon^2(x,y,\theta) &= (-x\cos\theta - y\sin\theta, -x\sin\theta + y\cos\theta, \theta),\\
\varepsilon^3(x,y,\theta) &= (-x,-y,\theta),\\
\varepsilon^4(x,y,\theta) &= (-x,y,-\theta),\\
\varepsilon^5(x,y,\theta) &= (-x\cos\theta - y\sin\theta, x\sin\theta - y\cos\theta, -\theta),\\
\varepsilon^6(x,y,\theta) &= (x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta, -\theta),\\
\varepsilon^7(x,y,\theta) &= (x,-y,-\theta).
\end{aligned} \tag{3.28}$$

All symmetries are involutions: $\varepsilon^i \circ \varepsilon^i = \text{id}$. Henceforth, the eight symmetries will be called the “fundamental symmetries”. In Figure 3.6 we have visualized the effect of the fundamental symmetries on a single $p \in \mathbb{M}_2$.

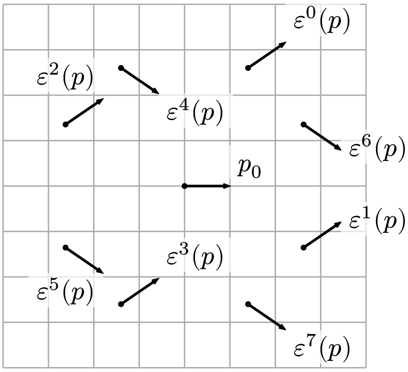


Figure 3.6: Visualization of the positions-orientations $\varepsilon^i(p)$ for a specific $p \in \mathbb{M}_2$, where ε^i are the fundamental symmetries (3.28). The reference position-orientation $p_0 = (0, 0, 0)$ is also plotted.

How all fundamental symmetries relate to each other becomes clearer if we write them down in either logarithmic or half-angle coordinates.

Lemma 3.1 (8 fundamental symmetries): The 8 fundamental symmetries ε^i, in either half-angle coordinates b^i or logarithmic coordinates c^i, correspond to sign flips as laid out in Table 3.1.

Proof: We will only show that ε^2 flips b^1. All other calculations are done analogously. Pick a point $p = (x, y, \theta)$ and let $q = \varepsilon^2(p)$. We now calculate $b^1(q)$:

$$\begin{aligned} b^1(q) &= x(q)\cos\frac{\theta(q)}{2} + y(q)\sin\frac{\theta(q)}{2} \\ &= -(x\cos\theta + y\sin\theta)\cos\frac{\theta}{2} \\ &\quad +(-x\sin\theta + y\cos\theta)\sin\frac{\theta}{2} \\ &= -x\Big(\cos\theta\cos\frac{\theta}{2} + \sin\theta\sin\frac{\theta}{2}\Big) \\ &\quad -y\Big(\sin\theta\cos\frac{\theta}{2} - \cos\theta\sin\frac{\theta}{2}\Big) \\ &= -x\cos\frac{\theta}{2} - y\sin\frac{\theta}{2} \\ &= -b^1(p), \end{aligned} \tag{3.29}$$

where we have used the trigonometric difference identities of cosine and sine in the second-to-last equality. From the relation between logarithmic and half-angle coordinates (3.26) we have that the logarithmic coordinates c^i flip in the same manner under the symmetries. ☐

	ε^0	ε^1	ε^2	ε^3	ε^4	ε^5	ε^6	ε^7
b^1, c^1	$+$	$+$	$-$	$-$	$-$	$-$	$+$	$+$
b^2, c^2	$+$	$-$	$+$	$-$	$+$	$-$	$+$	$-$
b^3, c^3	$+$	$+$	$+$	$+$	$-$	$-$	$-$	$-$

Table 3.1: The 8 fundamental symmetries ε^i, in either half-angle coordinates b^i or logarithmic coordinates c^i, correspond to sign flips.

The fixed points of the symmetries ε^2, ε^1, and ε^6 have a geometric interpretation. The logarithmic and half-angle coordinates, being so closely related to the fundamental symmetries, also carry the same interpretation. Definition 3.1 introduces this geometric idea and Lemma 3.2 makes its relation to the fixed points of the symmetries precise. In Figure 3.8 the fixed points are visualized, and in Figure 3.7 a visualization of these geometric ideas can be seen.

Definition 3.1 (Coradial, cocircular, and parallel): Two position-orientations $p_1, p_2 \in \mathbb{M}_2$ are called

- *coradial* if there exists a circle, of possibly infinite radius, such that p_1 and p_2 are both inward or outward normals to the circle.
- *cocircular* if there exists a circle, of possibly infinite radius, such that p_1 and p_2 are both clockwise or counterclockwise tangents to the circle.
- *parallel* if their orientations coincide.

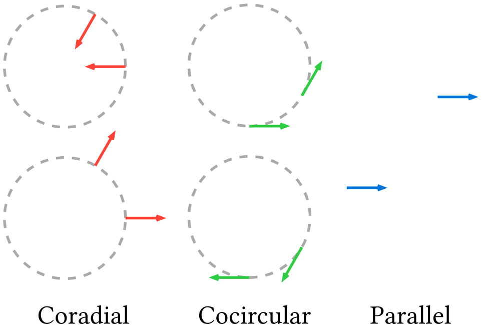


Figure 3.7: Examples of pairs of position-orientations in $\mathbb{M}_2$ that are coradial, cocircular, and parallel.

Co-circularity has a well-known characterization that is often used for line enhancement in image processing, such as tensor voting (Mordohai and Medioni (2006)).

Remark 3.1: Point $p = (r\cos\varphi, r\sin\varphi, \theta) \in \mathbb{M}_2$ is cocircular to the reference point $p_0 = (0, 0, 0)$ if and only if the double angle equality $\theta \equiv 2\varphi \bmod 2\pi$ holds.

In fact all fixed points of the fundamental symmetries can be intuitively characterized:

Lemma 3.2 (Fixed Points of Symmetries): Fix the reference point $p_0 = (0,0,0) \in \mathbb{M}_2$. The point $p = gp_0 \in \mathbb{M}_2$ with $g \in \mathrm{SE}(2)$ is, respectively,

- coradial to p_0 when

$$c^1(p) = 0 \Leftrightarrow \varepsilon^2(p) = p \Leftrightarrow g \in \exp(\langle A_2, A_3 \rangle), \tag{3.30}$$

- cocircular to p_0 when

$$c^2(p) = 0 \Leftrightarrow \varepsilon^1(p) = p \Leftrightarrow g \in \exp(\langle A_1, A_3 \rangle), \tag{3.31}$$

- parallel to p_0 when

$$c^3(p) = 0 \Leftrightarrow \varepsilon^6(p) = p \Leftrightarrow g \in \exp(\langle A_1, A_2 \rangle). \tag{3.32}$$

Proof: We will only show (3.31), the other formulas are done analogously. We start by writing $g = (r\cos\varphi, r\sin\varphi, \theta)$ and calculating that $p = gp_0 = (r\cos\varphi, r\sin\varphi, (\cos\theta, \sin\theta))$. Then by Remark 3.1 we know that p is cocircular to p_0 if and only if $2\varphi = \theta \bmod 2\pi$. We can show this is equivalent to $c^2(p) = 0$:

$$\begin{aligned} c^2(p) = 0 \Leftrightarrow b^2(p) &= 0 \\ &\Leftrightarrow -x\sin\frac{\theta}{2} + y\cos\frac{\theta}{2} = 0 \\ &\Leftrightarrow -\cos\varphi\sin\frac{\theta}{2} + \sin\varphi\cos\frac{\theta}{2} = 0 \\ &\Leftrightarrow \sin\left(\varphi - \frac{\theta}{2}\right) = 0 \Leftrightarrow 2\varphi = \theta \bmod 2\pi. \end{aligned} \tag{3.33}$$

In logarithmic coordinates ε^1 is equivalent to:

$$\varepsilon^1(c^1, c^2, c^3) = (c^1, -c^2, c^3), \tag{3.34}$$

from which we may deduce that $\varepsilon^1(p) = p$ is equivalent to $c^2(p) = 0$. Finally, the equivalence $c^2(p) = 0 \Leftrightarrow g \in \exp(\langle A_1, A_3 \rangle)$ follows immediately from the definition of the logarithmic coordinates (3.5).□

In Moiseev and Sachkov (2010) it is shown that the exact sub-Riemannian distance d_{sr} is invariant under the fundamental symmetries ε^i. However, these same symmetries hold true for the Riemannian distance d. Moreover, because the approximative distances use the logarithmic coordinates c^i and half-angle coordinates b^i they also carry the same symmetries. The following lemma makes this precise.

Lemma 3.3 (Symmetries of the exact distance and all proposed approximations): All exact and approximative (sub)-Riemannian distances (with respect to the reference point p_0) (3.8), (3.9), (3.10), (3.16), and (3.17), are invariant under all the fundamental symmetries ε^i.

Proof: By Table 3.1 one sees that ε^3, ε^4, and ε^5 generate all symmetries. Therefore if we just show that all distances are invariant under these select three symmetries we also have shown that they are invariant under all symmetries. We will first show the exact distance, in either the Riemannian or sub-Riemannian case, is invariant with respect to these three symmetries, that is $d(p) = d(\varepsilon^i(p))$ for $i \in \{3, 4, 5\}$.

Consider the push forward ε^3_*. By direct computation (in (x, y, θ) coordinates) we have $\varepsilon^3_* \mathcal{A}_i|_p = \pm \mathcal{A}_i|_{\varepsilon^3(p)}$. Because the metric tensor field $\mathcal{G}$ (3.1) is diagonal with respect to the $\mathcal{A}_i$ basis (2.34) this means that ε^3 is an isometry. Similarly, ε^4 is an isometry. Being an isometry of the metric $\mathcal{G}$ we may directly deduce that ε^3 and ε^4 preserve distance. The ε^5 symmetry flips all the signs of the logarithmic coordinates c^i, which amounts to Lie algebra inversion: $-\log g = \log(\varepsilon^5(g))$. Taking the exponential on both sides shows that $g^{-1} = \varepsilon^5(g)$. By left-invariance of the metric we have $d(gp_0, p_0) = d(p_0, g^{-1}p_0)$, which holds in both the Riemannian and sub-Riemannian case, and thus $d(gp_0) = d(\varepsilon^5(gp_0))$.

That all approximative distances (both in the Riemannian and sub-Riemannian case) are also invariant under all the symmetries is not hard to see: every b^i and c^i term is either squared or the absolute value is taken. Flipping signs of these coordinates, recall Lemma 3.1, has no effect on the approximative distance. □

In Figure 3.1 the previous lemma can be seen. The two fundamental symmetries ε^2 and ε^1 correspond, respectively, to reflecting the isocontours (depicted in colors) along their short edge and long axis. The ε^6 symmetry corresponds to mapping the positive θ isocontours to their negative θ counterparts. In Figure 3.8 one can see an isocontour of ρ_b together with the symmetry "planes" of ε^2, ε^1 and ε^6.

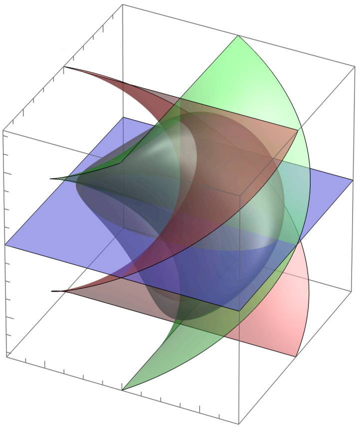


Figure 3.8: In gray the isocontour $\rho_b = 2.5$ together with the fixed points of ε^2 (red), ε^1 (green), and ε^6 (blue). For ε^2 and ε^1 only the points within the region $x^2 + y^2 \leq 2^2$ are plotted. The fixed points of ε^2, ε^1, and ε^6 correspond respectively to the points in $\mathbb{M}_2$ that are coradial, cocircular, and parallel to the reference point p_0. The metric parameters are $(w_1, w_2, w_3) = (1, 2, 1)$.

3.3.2 Simple Global Bounds

Next, we provide some basic global lower and upper bounds for the exact Riemannian distance d. Recall that the lower bound l plays an important role in the combined approximation $\rho_{c,\text{com}}$ (3.17) when far from the reference point p_0.

Lemma 3.4 (Global bounds on distance): The exact Riemannian distance $d = d(p_0, \cdot)$ is greater than or equal to the following lower bound $l : \mathbb{M}_2 \to \mathbb{R}$:

$$l := \sqrt{(w_1 x)^2 + (w_1 y)^2 + (w_3 \theta)^2} \leq d, \tag{3.35}$$

and less than or equal to the following upper bounds $u_1, u_2 : \mathbb{M}_2 \to \mathbb{R}$:

$$\begin{aligned} d &\leq u_1 := \sqrt{(w_2 x)^2 + (w_2 y)^2 + (w_3 \theta)^2}, \\ d &\leq u_2 := \sqrt{(w_1 x)^2 + (w_1 y)^2} + w_3 \pi. \end{aligned} \tag{3.36}$$

Proof: We will first show $l \leq d$. Consider the following spatially isotropic metric $\tilde{\mathcal{G}}$ with corresponding norm:

$$\|(p, \dot{p})\|_{\tilde{\mathcal{G}}}^2 = (w_1 \|\dot{x}\|)^2 + (w_3 \|\dot{n}\|)^2, \tag{3.37}$$

where $p = (x, n) \in \mathbb{M}_2$ and $\dot{p} = (\dot{x}, \dot{n}) \in T_p\mathbb{M}_2$. We assumed, without loss of generality, that $w_1 \leq w_2$, so for any vector $v \in T\mathbb{M}_2$ it holds that $\|v\|_{\tilde{\mathcal{G}}} \leq \|v\|_{\mathcal{G}}$. From this we can directly deduce that for any curve γ on $\mathbb{M}_2$ we have that $L_{\tilde{\mathcal{G}}}(\gamma) \leq L_{\mathcal{G}}(\gamma)$. Now consider a length-minimizing curve γ with respect to $\mathcal{G}$ between the reference point p_0 and some end point p. We then have the chain of (in)equalities:

$$d_{\tilde{\mathcal{G}}}(p) \leq L_{\tilde{\mathcal{G}}}(\gamma) \leq L_{\mathcal{G}}(\gamma) = d_{\mathcal{G}}(p). \tag{3.38}$$

Furthermore, because the metric $\tilde{\mathcal{G}}$ is spatially isotropic (2.39) we have

$$d_{\tilde{\mathcal{G}}}(p) = \sqrt{(w_1 x)^2 + (w_1 y)^2 + (w_3 \theta)^2} = l. \tag{3.39}$$

Putting everything together gives the desired result of $l \leq d$. To show that $d \leq u_1$ can be done analogously.

As for showing $d \leq u_2$ we will construct a curve γ of which the length $L(\gamma)$ with respect to $\mathcal{G}$ can be bounded from above with u_2. This in turn shows that $d \leq u_2$ by definition of the distance. Pick a destination position and orientation $p = (x, n)$. The constructed curve γ will be as follows. We start by aligning our starting orientation $n_0 = (1, 0) \in S^1$ to the destination position x. This desired orientation towards x is

$\hat{x} := \frac{x}{r}$ where $r = \|x\| = \sqrt{x^2 + y^2}$. This action will cost $w_3 a$ for some $a \geq 0$. Once we are aligned with $\hat{x}$ we move towards x. Because we are aligned, this action will cost $w_1 r$. Now that we are at x we align our orientation with the destination orientation n, which will cost $w_3 b$ for some $b \geq 0$. Altogether we have $L(\gamma) = w_1 r + w_3(a + b)$. In its current form the constructed curve actually does not have $a + b \leq \pi$ as desired. To fix this we realize that we do not necessarily have to align with $\hat{x}$. We could have aligned with $-\hat{x}$ and move backwards towards x, which will also cost $w_1 r$. One can show that one of the two methods (either moving forwards or backwards towards x) indeed ensures that $a + b \leq \pi$, and thus $d \leq u_2$. ☐

These bounds are simple but effective: they help us prove a multitude of insightful corollaries.

Corollary 3.1 (Global error distance): Simple manipulations, together with the fact that $x^2 + y^2 = (b^1)^2 + (b^2)^2$, give the following inequalities between l, u_1 and ρ_b:

$$l \leq \rho_b \leq u_1, \quad \frac{1}{\zeta} u_1 \leq \rho_b \leq \zeta l. \tag{3.40}$$

The second equation can be extended to inequalities between ρ_b and d:

$$\frac{1}{\zeta} d \leq \rho_b \leq \zeta d. \tag{3.41}$$

Remark 3.2: In the spatially isotropic case $w_1 = w_2 \Leftrightarrow \zeta = 1$, then the lower and upper bound coincide, thus becoming exact. Because ρ_b is within the lower and upper bound it also becomes exact.

The previous result indicates that problems can arise if $\zeta \to \infty$, which indeed turns out to be the case:

Corollary 3.2 (Observing the problem): If we restrict ourselves to $x = \theta = 0$ we have that $u_1 = \rho_b = \rho_c = w_2|y|$. From this we can deduce that one can be certain that both ρ_b and ρ_c become bad approximations away from p_0. Namely, when $\zeta > 1 \Leftrightarrow w_2 > w_1$ both approximations go above u_2 if one looks far enough away from p_0. How "fast" it goes bad is determined by all metric parameters. Namely, the intersection of the approximations ρ_b and ρ_c, and u_2 is at $|y| = (w_3\pi)/(w_2 - w_1)$, or equivalently at $\rho = (w_3\pi)/(1 - \zeta^{-1})$. This intersection is visible in Figure 3.9 in the higher anisotropy cases. From this expression of the intersection we see that in the cases $w_3 \to 0$ and $\zeta \to \infty$ the Riemannian distance approximations ρ_b and ρ_c quickly go bad. We will see exactly the same behavior in Lemma 3.7

Lemma 3.4 is visualized in Figure 3.9 and Figure 3.10. In Figure 3.9 we consider the behavior of the exact distance and bounds along the y-axis, that is at $x = \theta = 0$. We have chosen to inspect the y-axis because it consist of points that are hard to reach from the reference point p_0 when the spatial anisotropy is large, which makes it interesting. In contrast, along the x-axis l, d, ρ_b, ρ_c, u_1 and $w_1|x|$ all coincide, and is therefore uninteresting. To provide more insight we also depict the bounds along the $y = x$ axis, see Figure 3.10. Observe that in both figures, the exact distance d is indeed always above the lower bound l and below the upper bounds u_1 and u_2.

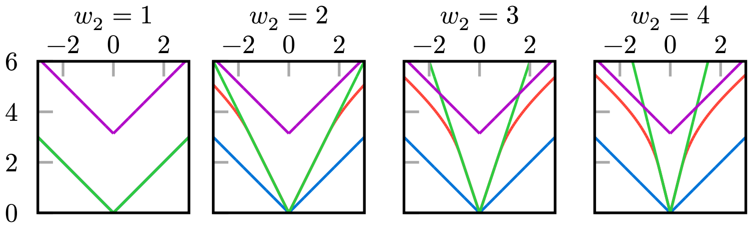


Figure 3.9: Exact distance d (red), lower bound l (blue), and upper bounds u_1 (green), u_2 (purple) along the y-axis, i.e at $x = \theta = 0$, for increasing spatial anisotropy. We keep $w_1 = w_3 = 1$ and vary w_2. The horizontal axis is y and the vertical axis the value of the distance/bound. Note how the exact distance d starts of linearly with a slope of w_2, and ends linearly with a slope of w_1.

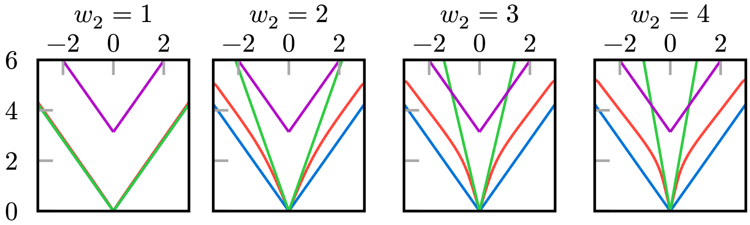


Figure 3.10: Same setting as Figure 3.9 but at $x = y, \theta = 0$. The horizontal axis moves along the line $x = y$.

3.3.3 Asymptotic Error Expansion

In this subsection we provide an asymptotic expansion of the error between the exact distance d and the half-angle distance approximation ρ_b (Lemma 3.7).

Lemma 3.5: Let $\gamma : [0, 1] \to \mathbb{M}_2$ be a minimizing geodesic from p_0 to p. We have that:

$$\rho_b(p) \leq d(p) \max_{t \in [0,1]} \left\| d\rho_b|_{\gamma(t)} \right\|. \tag{3.42}$$

Proof: The fundamental theorem of calculus tells us that:

$$\int_0^1 (\rho_b \circ \gamma)'(t) \, \mathrm{d}t = \rho_b(\gamma(1)) - \rho_b(\gamma(0)) = \rho_b(p), \tag{3.43}$$

but one can also bound this expression as follows:

$$\begin{aligned}\int_0^1 (\rho_b \circ \gamma)'(t)\,\mathrm{d}t &= \int_0^1 \langle d\rho_b|_{\gamma(t)}, \dot{\gamma}(t)\rangle\,\mathrm{d}t \\ &\leq \int_0^1 \left\|d\rho_b|_{\gamma(t)}\right\| \|\dot{\gamma}(t)\|\,\mathrm{d}t \\ &\leq \left(\max_{t\in[0,1]} \left\|d\rho_b|_{\gamma(t)}\right\|\right) \int_0^1 \|\dot{\gamma}(t)\|\,\mathrm{d}t \\ &= d(p) \max_{t\in[0,1]} \left\|d\rho_b|_{\gamma(t)}\right\|.\end{aligned} \tag{3.44}$$

Putting the two together gives the desired result. ☐

Lemma 3.6: One can bound $\|d\rho_b\|$ around p_0 by:

$$\|d\rho_b\|^2 \leq 1 + \frac{\zeta^2 - 1}{2w_3^2}\rho_b^2 + \mathcal{O}(\theta^3). \tag{3.45}$$

Proof: We start by writing out the explicit form of $\|d\rho_b\|^2$ in the left-invariant frame:

$$\|d\rho_b\|^2 = w_1^{-2}(\mathcal{A}_1\rho_b)^2 + w_2^{-2}(\mathcal{A}_2\rho_b)^2 + w_3^{-2}(\mathcal{A}_3\rho_b)^2. \tag{3.46}$$

By replacing the left-invariant derivatives with half-angle coordinates derivatives we can equivalently write this as:

$$\begin{aligned}&w_2^{-2}\left(\left|\frac{\partial\rho}{\partial b^1}\right|^2 + \left|\frac{\partial\rho}{\partial b^2}\right|^2\right) \\ &+(w_1^{-2} - w_2^{-2})\left|\cos\left(\frac{b^3}{2}\right)\frac{\partial\rho}{\partial b^1} + \sin\left(\frac{b^3}{2}\right)\frac{\partial\rho}{\partial b^2}\right|^2 \\ &+w_3^{-2}\left|\frac{1}{2}\frac{\partial\rho}{\partial\psi} + \frac{\partial\rho}{\partial b^3}\right|^2,\end{aligned} \tag{3.47}$$

where $\psi = \arctan(b^2, b^1)$, $\partial_\psi = b^2\partial_{b^1} - b^1\partial_{b^2}$, and we omitted the subscript b from ρ for conciseness. We are going to Taylor expand the sin and cosine in the second term up to the second order term. This becomes

$$\begin{aligned} \left|\cos\left(\frac{b^3}{2}\right)\frac{\partial\rho}{\partial b^1} + \sin\left(\frac{b^3}{2}\right)\frac{\partial\rho}{\partial b^2}\right|^2 &= \left|\frac{\partial\rho}{\partial b^1}\right|^2 \\ +\theta\left(\frac{\partial\rho}{\partial b^1}\frac{\partial\rho}{\partial b^2}\right) + \frac{\theta^2}{4}\left(\left|\frac{\partial\rho}{\partial b^2}\right|^2 - \left|\frac{\partial\rho}{\partial b^1}\right|^2\right) &+ O(\theta^3). \end{aligned} \tag{3.48}$$

This allows us to write $\|d\rho_b\|^2$ as

$$w_1^{-2}\left|\frac{\partial\rho}{\partial b^1}\right|^2 + w_2^{-2}\left|\frac{\partial\rho}{\partial b^2}\right|^2 + w_3^{-2}\left|\frac{\partial\rho}{\partial b^3}\right|^2 + \varepsilon. \tag{3.49}$$

Making use of the fact that the first part in this expression equals 1, we can thus write $\|d\rho_b\|^2 = 1 + \varepsilon$. The exact form of ε is as follows

$$\begin{aligned} \varepsilon = \frac{w_1^2 - w_2^2}{4w_1^2 w_2^2 w_3^2 \rho_b^2}\Big(& w_1^4 w_3^2 (b^1 b^3)^2 - w_2^4 w_3^2 (b^2 b^3)^2 \\ &+ w_1^2 w_2^2 (w_1^2 - w_2^2)(b^1 b^2)^2\Big) + O(\theta^3). \end{aligned} \tag{3.50}$$

Using that $w_i|\ b^i| \leq \rho_b$ we can bound the expression from above by

$$\varepsilon \leq \rho_b^2 \frac{|w_1^2 - w_2^2|}{4w_1^2 w_2^2 w_3^2}\left(w_1^2 + w_2^2 + |w_1^2 - w_2^2|\right) + O(\theta^3). \tag{3.51}$$

Finally the lemma follows by algebraic manipulations and the fact that $w_1 \leq w_2$.

□

By combining Lemma 3.5 and Lemma 3.6 one can find an expression for the asymptotic error between the exact distance d and the half-angle approximation ρ_b.

Lemma 3.7: Around any compact neighborhood of p_0 we have that

$$\rho_b^2 \leq (1+\varepsilon)d^2, \quad \text{where} \quad \varepsilon := \frac{\zeta^2-1}{2w_3^2}\zeta^4\rho_b^2 + C|\theta|^3. \tag{3.52}$$

for some $C \geq 0$.

Proof: Let $p \in U$ be given, and let $\gamma : [0,1] \to \mathbb{M}_2$ be the geodesic from p_0 to p. For the distance we know that

$$d(\gamma(s)) \leq d(\gamma(t)), \quad \text{for} \quad s \leq t. \tag{3.53}$$

Making use of (3.41) we know that $\frac{1}{\zeta}\rho_b \leq d \leq \zeta\rho_b$ so we can combine this with the previous equation to find:

$$\rho_b(\gamma(s)) \leq \zeta^2\rho_b(\gamma(t)), \quad \text{for} \quad s \leq t. \tag{3.54}$$

from which we get that

$$\max_{t\in[0,1]} \rho_b(\gamma(t)) \leq \zeta^2\rho_b(p). \tag{3.55}$$

Combining this fact with the above two lemmas allows us to conclude (3.52). ☐

3.4 EXPERIMENTS

In this section we perform three experiments. First, we quantify the error of the half-angle approximation ρ_b by numerically computing the exact Riemannian distance d and comparing it with ρ_b. Second, we assess the performance of PDE-G-CNNs versus CNNs and G-CNNs on the DCA1 dataset (Subsection 2.8.2), and examine how different distance approximations influence the accuracy of the PDE-G-CNN architecture. Third, we repeat the DCA1 experiments on the Lines dataset (Subsection 2.8.3). The code for these experiments is available in the GitLab repository LieTorch.

3.4.1 ERROR OF HALF-ANGLE APPROXIMATION

We can quantitatively analyze the error between any distance approximation ρ and the exact Riemannian distance d as follows. We do this by first choosing a region $\Omega \subseteq \mathbb{M}_2$ in which we will analyze the approximation. Just as in Figure 3.2 and Figure 3.3 we decided to inspect $\Omega := [-3,3] \times [-3,3] \times [-\pi,\pi) \subseteq \mathbb{M}_2$. As for our exact measure of error ε we have decided on the *mean relative error* defined as:

$$\varepsilon := \frac{1}{\mu(\Omega)} \int_{\Omega} \frac{|\rho_b(p) - d(p)|}{d(p)} \, \mu(\mathrm{d}p) \tag{3.56}$$

where μ is the induced Riemannian measure determined by the Riemannian metric $\mathcal{G}$. We then discretized our domain Ω into a grid of $101 \times 101 \times 101$ uniformly spaced points $p_i \in \Omega$, indexed by some index set $i \in I$, and numerically solved for the exact distance d on this grid. This numerical scheme is of course not exact and we will refer to these values as $\tilde{d}_i \approx d(p_i)$. We also calculate the value of the distance approximation ρ on the grid points $\rho_i := \rho(p_i)$. Once we have these values we can approximate the true mean relative error ε by calculating the numerical error $\tilde{\varepsilon}$ defined by:

$$\varepsilon \approx \tilde{\varepsilon} := \frac{1}{|I|} \sum_{i \in I} \frac{|\rho_i - \tilde{d}_i|}{\tilde{d}_i} \tag{3.57}$$

In Table 3.2 the numerical mean relative error $\tilde{\varepsilon}$ between the half-angle approximation ρ_b and the numerical Riemannian distance $\tilde{d}$ can be seen for different spatial anisotropies ζ. We keep $w_1 = w_3 = 1$ constant and vary w_2. We see that, as shown visually in Figure 3.2 and Figure 3.3, that ρ_b gets worse and worse when we increase the spatial anisotropy ζ.

There is a discrepancy in the table worth mentioning. We know from Remark 3.2 that when $\zeta = 1$ then $\rho_b = d$ and thus $\varepsilon = 0$. But surprisingly we do not have $\tilde{\varepsilon} = 0$ in the $\zeta = 1$ case in Table 3.2. This can be simply explained by the fact that the numerical solution $\tilde{d}$ is not exactly equal to the true distance d. We expect that $\tilde{\varepsilon}$ will go to 0 in the $\zeta = 1$ case if we discretize our region Ω more and more finely.

We can compare these numerical results to our theoretical results. Namely, we can deduce from (3.41) that: $\frac{|\rho_b - d|}{d} \leq \zeta - 1$, which means

$$\varepsilon \leq \zeta - 1. \tag{3.58}$$

And so we expect this to also approximately hold for the numerical mean relative error $\tilde{\varepsilon}$. Indeed, in Table 3.2 we can see that $\tilde{\varepsilon} \lessapprox \zeta - 1$.

Interestingly, we see that $\tilde{\varepsilon}$ is relatively small compared to our theoretical bound (3.58) even in the high anisotropy cases. However, this is only a consequence of the relative smallness of Ω. If we make Ω bigger and bigger we can be certain that ε converges to $\zeta - 1$. This follows from an argument similar to the reasoning in Corollary 3.2.

ζ	1	1.5	2	3	4	6	8
$\tilde{\varepsilon}$	0.027	0.051	0.14	0.41	0.71	1.4	2.1

Table 3.2: Numerical mean relative error $\tilde{\varepsilon}$ between ρ_b and d for multiple spatial anisotropies ζ.

3.4.2 DCA1

To establish a baseline we train a 3, 6, and 12 layer CNN, G-CNN and PDE-G-CNN on the DCA1 dataset (Subsection 2.8.2).

Since the goal is coronary artery segmentation, we will be training using the Dice loss (2.59). The exact architectures are identical/analogous to the ones used in Smets, Portegies, Bekkers and Duits (2023), Fig.15. For the baseline the logarithmic distance approximation ρ_c was used within the PDE-G-CNNs, as this approximation was the first to be proposed and used. Every network was trained 10 times for 80 epochs. During training, the average Dice coefficient on the test set is computed after each epoch, and the best value is recorded. The result is 10 Dice coefficients for every architecture. The number of parameters of the networks can be found in Table 3.3. The outcome of this baseline can be seen in Figure 3.11.

We see that PDE-G-CNNs consistently perform equally well as, and sometimes outperform, G-CNNs and CNNs, all the while having the fewest parameters of all architectures.

Parameters	3 layers	6 layers	12 layers
CNN	2814	25662	73614
G-CNN	2058	24632	72728
PDE-G-CNN	1264	2560	2698

Table 3.3: The total amount of parameters in the networks that are used in Figure 3.11.

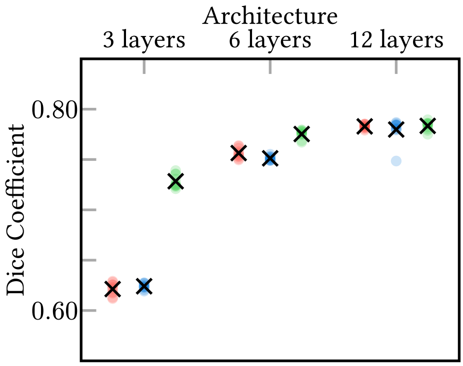


Figure 3.11: A scatterplot showing how a 3, 6, and 12 layer CNN (red), G-CNN (blue), and PDE-G-CNN (green) compare on the DCA1 dataset. The crosses indicate the mean. We see the PDE-G-CNNs provide equal or better results with respectively 2, 10 and 35 times less parameters, see Table 3.3.

To compare the effect of using different approximative distances we decided to train the 6 layer PDE-G-CNN (with 2560 parameters) 10 times for 80 epochs using each distance approximation. The results can be found in Figure 3.12 and Figure 3.13.

We see that on DCA1 all distance approximations have comparable accuracy. We notice a small drop in effectiveness when using $\rho_{b,\mathrm{sr}}$, and a small increase when using $\rho_{b,\mathrm{com}}$.

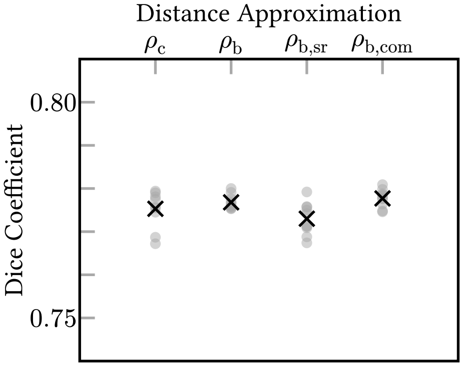


Figure 3.12: A scatterplot showing how the use of different distance approximations affect the accuracy of the 6 layer PDE-G-CNN on the DCA1 dataset. The crosses indicate the mean.

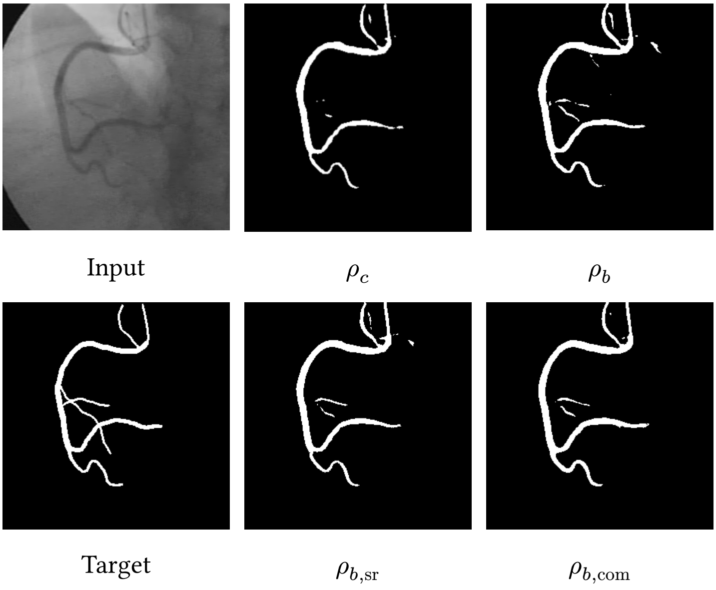


Figure 3.13: One sample from DCA1 and the corresponding outputs of the 6 layer PDE-G-CNN for each distance approximation. The networks have an accuracy approximately equal to the mean in Figure 3.12. In agreement with the scatterplot, it is hard to spot a noticeable difference between distance approximations.

3.4.3 LINES

We anticipate that PDE-G-CNNs outperform classical CNNs on the Lines dataset (Subsection 2.8.3) due to PDE-G-CNNs being able to dilate and erode. To establish a baseline we train a 6 layer CNN, G-CNN and PDE-G-CNN.

For this baseline we again used ρ_c within the PDE-G-CNN, but changed the number of channels to 30, and the kernel sizes to $[9, 9, 9]$, making the total amount of parameters 6018. By increasing the kernel size we anticipate that the difference in effectiveness of using the different distance approximations, if there is any, becomes more pronounced. Every network was trained 15 times for 60 epochs. The result of this baseline can be seen in Figure 3.14.

We see that the PDE-G-CNN outperforms the G-CNN, which in turn outperforms the CNN, while having the fewest parameters.

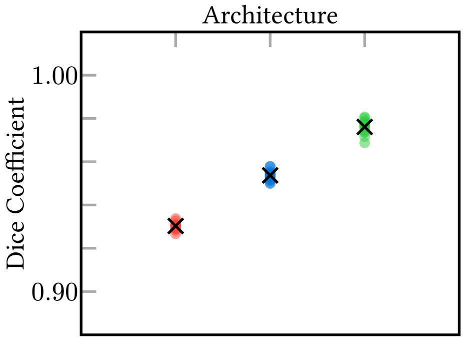


Figure 3.14: A scatterplot showing how a 6 layer CNN (red, 25662 parameters), G-CNN (blue, 24632 parameters), and a PDE-G-CNN (green, 6018 parameters) compare on the Lines dataset. The crosses indicate the mean.

We again test the effect of using different approximative distances by training the 6 layer PDE-G-CNN 15 times for 60 epochs for every approximation. The results can be found in Figure 3.15.

We see that on the Lines dataset all distance approximations again have a comparable accuracy. We again notice an increase in effectiveness when using $\rho_{b,\text{com}}$, just as on the DCA1 dataset.

Interestingly, using $\rho_{b,\mathrm{sr}}$ does not seem to hurt the accuracy on the Lines dataset, which is in contrast with DCA1. This is in line with what one would expect in view of the existing sub-Riemannian line-perception models in neurogeometry.

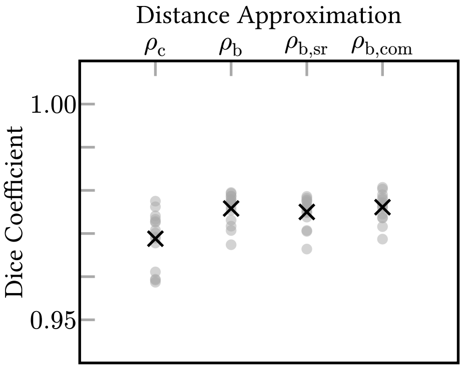


Figure 3.15: A scatterplot showing how the use of different distance approximations affect the accuracy of the 6 layer PDE-G-CNN on the Lines dataset. The crosses indicate the mean.

3.5 Conclusion

In Theorem 3.1 we show local and global bounds on the error between the half-angle distance approximation ρ_b and the exact Riemannian distance d on $\mathbb{M}_2$. Our new estimates for distance on $\mathbb{M}_2$ have bounds that explicitly depend on the metric tensor field coefficients. This allows us to theoretically underpin that if the spatial anisotropy goes up, the distance approximations become less and less accurate.

Indeed, as we show qualitatively in Figure 3.3 and quantitatively in Subsection 3.4.1, if the spatial anisotropy ζ is high one must resort to sub-Riemannian approximations. Furthermore, we propose a single distance approximation $\rho_{b,\mathrm{com}}$ that works both for low and high spatial anisotropy.

Apart from how well we approximate the true Riemannian distance, there is the issue of how well each of the distance approximations perform in applications within the PDE-G-CNNs. In practice, the approximations ρ_b, ρ_c, $\rho_{b,\mathrm{com}}$ perform similarly. This is not surprising as our theoretical result Lemma 3.3 reveals that all distance approximations carry the correct eight fundamental symmetries. Nevertheless, Figure 3.12 and Figure 3.15 do reveal advantages of using the new distance approximations (in particular $\rho_{b,\mathrm{com}}$) over the old approximation ρ_c.

In all cases, the PDE-G-CNNs still outperform G-CNNs and CNNs on the DCA1 and Lines datasets: they have a higher (or equal) accuracy, while having a huge reduction in network complexity, even when using 3 layers. Regardless of the choice of distance approximation (ρ_c, ρ_b, $\rho_{b,\mathrm{sr}}$, $\rho_{b,\mathrm{com}}$) the advantage of PDE-G-CNNs over G-CNNs and CNNs is significant, as can be clearly observed in Figure 3.11 and Figure 3.14.

Altogether, PDE-G-CNNs have better geometric reduction, accuracy, and geometric interpretation, than classical CNNs and G-CNNs on various segmentation problems.

INVARIANT METRICS ON POSITION-ORIENTATION SPACE

4

4.1 INTRODUCTION

Riemannian metrics $\mathcal{G}$ on position-orientation space $\mathbb{M}_3$ that are invariant under the roto-translation group $\mathrm{SE}(3)$ appear in various works related to medical image analysis, including enhancement, denoising, and segmentation (DUITS AND FRANKEN (2011), PORTEGIES, FICK, SANGUINETTI, MEESTERS, GIRARD AND DUITS (2015), PORTEGIES, SANGUINETTI, MEESTERS AND DUITS (2015), PORTEGIES (2018), DUITS, ST-ONGE, PORTEGIES AND SMETS (2019), SMETS, PORTEGIES, ST-ONGE AND DUITS (2021), SMETS, PORTEGIES, BEKKERS AND DUITS (2023)). Such invariant metrics $\mathcal{G}$ are used to define (nonlinear) PDEs on $\mathbb{M}_3$ which process the data in a roto-translation equivariant manner.

In practice, we are interested in the induced Riemannian distance $d_{\mathcal{G}} : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}_{\geq 0}$. This is because, if one has access to the Riemannian distance, it can be used to efficiently solve the PDEs numerically, as for example seen in Section 2.6.

However, in general, the exact Riemannian distance on $\mathbb{M}_3$ is expensive to determine. In PORTEGIES, SANGUINETTI, MEESTERS AND DUITS (2015) it is therefore suggested to instead use, what we will call, the *mav distance*

$$\mu_{\mathcal{G}}(p_1, p_2) = L_{\mathcal{G}}(\eta), \tag{4.1}$$

where $\eta : [0,1] \to \mathbb{M}_3$ is the curve $\eta(t) = \exp(M(p_1,p_2)t) \triangleright p_1$, and $M(p_1,p_2) \in \mathfrak{se}(3)$ is the unique generator between p_1 and $\exp(M(p_1,p_2)) \triangleright p_1 = p_2$ with *minimal angular velocity* (mav).

The mav distance is favored over the Riemannian distance as it is more tractable: the calculation of the length of η is straightforward. Indeed, one can show that

$$L_{\mathcal{G}}(\eta) = \|M(p_1, p_2) \triangleright p_1\|_{\mathcal{G}}. \tag{4.2}$$

We see an application of the mav distance $\mu_{\mathcal{G}}$ in equivariant machine learning, specifically to be used as an invariant in the PONITA architecture (Section 2.7). Namely, all invariant metrics $\mathcal{G}$ on a homogeneous space can be parametrized by a small set of real numbers – the metric parameters w_i – and $\mathbb{M}_3$ is no different. This means that the mav distance $\mu_{\mathcal{G}}$ is in fact a *trainable* invariant with the metric parameters w_i acting as learnable weights.

In this chapter, we investigate if the mav distance $\mu_{\mathcal{G}}(p_1, p_2)$ can positively impact the accuracy of the PONITA architecture. However, before we can do this we need to: 1) classify and parametrize all $\mathrm{SE}(3)$ invariant metrics $\mathcal{G}$ on $\mathbb{M}_3$, and 2) explicitly construct the mav generator $M(p_1, p_2)$.

4.1.1 CONTRIBUTIONS

We classify and parametrize all $\mathrm{SE}(3)$ invariant metrics $\mathcal{G}$ on $\mathbb{M}_3$. Our rigorous approach reveals a broader family of metrics than the related literature currently suggests.

We explicitly construct the mav generator $M(p_1, p_2) \in \mathfrak{se}(3)$ between two position-orientations $p_1, p_2 \in \mathbb{M}_3$. Our approach is more geometric than algebraic, building upon the fact that any roto-translation can be realized as a *screw displacement*.

We incorporate the mav distance $\mu_{\mathcal{G}}(p_1, p_2)$ as a trainable invariant into the PONITA architecture and examine its impacts on accuracy.

4.1.2 OUTLINE

- In Section 4.2 we classify and parametrize all $\mathrm{SE}(3)$ invariant metrics $\mathcal{G}$ on $\mathbb{M}_3$.
- In Section 4.3 we explicitly construct the mav generator $M(p_1, p_2)$.
- In Section 4.4 we experimentally evaluate the mav distance $\mu_{\mathcal{G}}$ using the PONITA architecture.
- In Section 4.5 we conclude this chapter.

4.2 INVARIANT METRICS

In this section we classify all $\mathrm{SE}(3)$ invariant metrics $\mathcal{G}$ on $\mathbb{M}_3$. We will only state the parallelogram law abiding norm $\|\cdot\|_{\mathcal{G}}$. One can reconstruct the corresponding inner product using the polarization identity.

Theorem 4.1 (SE(3) Invariant Metrics on $\mathbb{M}_3$): Let $p = (x, n) \in \mathbb{M}_3$ and $\dot{p} = (\dot{x}, \dot{n}) \in T_p\mathbb{M}_3$. Every $\mathrm{SE}(3)$ invariant Riemannian metric tensor field $\mathcal{G}$ on $\mathbb{M}_3$ yields a norm of the form

$$\begin{aligned}\|(p,\dot{p})\|_{\mathcal{G}}^2 = (w_1|\dot{x}\cdot n|)^2 + (w_2\|\dot{x}\times n\|)^2 + (w_3\|\dot{n}\|)^2 \\ +2w_4\dot{x}\cdot\dot{n} + 2w_5\dot{x}\cdot(\dot{n}\times n),\end{aligned} \tag{4.3}$$

with constants $w_i \in \mathbb{R}$ that satisfy the positivity constraint $w_2^2 w_3^2 > w_4^2 + w_5^2$.

Proof:

Invariance. To prove that the stated metric is indeed $\mathrm{SE}(3)$ invariant we must show that $\|g \triangleright (p,\dot{p})\|_{\mathcal{G}} = \|(p,\dot{p})\|_{\mathcal{G}}$ for all $g \in \mathrm{SE}(3)$ and $(p,\dot{p}) \in T\mathbb{M}_3$. This is quickly verified using (2.23) and (2.24) and basic properties of the inner and cross product.

Classification. Consider an arbitrary $\mathrm{SE}(3)$ invariant metric $\mathcal{H}$. To prove that $\mathcal{H}$ is of the form (4.3) it suffices to show that $\mathcal{H}_p$ coincides with (4.3) at any position-orientation $p = (x, n)$. That the w_i have to be constants then follows from the invariance together with the fact that $\mathrm{SE}(3)$ acts transitively on $\mathbb{M}_3$.

Let e_1, e_2, and e_3 be any orthonormal basis of $\mathbb{R}^3$ with $e_1 = n$. Define the basis $f_1 = (e_1, 0)$, $f_2 = (e_2, 0)$, $f_3 = (e_3, 0)$, $f_4 = (0, e_2)$, $f_5 = (0, e_3)$ for $T_p\mathbb{M}_3$. One can check that this is indeed a basis of $T_p\mathbb{M}_3$ (2.21). Define the components $h_{ij} = \mathcal{H}_p(f_i, f_j)$.

The metric $\mathcal{H}_p$ should be invariant under $\operatorname{stab} p \subset \mathrm{SE}(3)$ which are rotations in the plane spanned by e_2 and e_3 (positioned at x). Specifically, consider the $g \in \operatorname{stab} p$ that rotates $e_2 \mapsto e_3$ and $e_3 \mapsto$

$-e_2$. Enforcing invariance under this specific rotation in the f basis we find the equations $\mathcal{H}_p(f_i, f_j) = \mathcal{H}_p(gf_i, gf_j)$ for all $i, j = 1, \ldots, 5$. Two examples are

$$\begin{aligned} h_{23} &= \mathcal{H}_p(f_2, f_3) = \mathcal{H}_p(gf_2, gf_3) = \mathcal{H}_p(f_3, -f_2) = -h_{32}, \\ h_{34} &= \mathcal{H}_p(f_3, f_4) = \mathcal{H}_p(gf_3, gf_4) = \mathcal{H}_p(-f_2, f_5) = -h_{25}. \end{aligned} \tag{4.4}$$

These equations, together with the symmetry constraints $h_{ij} = h_{ji}$, leaves us with the following component matrix

$$\begin{aligned} h_{ij} &= \begin{pmatrix} h_{11} & 0 & 0 & 0 & 0 \\ 0 & h_{22} & 0 & h_{24} & h_{25} \\ 0 & 0 & h_{22} & -h_{25} & h_{24} \\ 0 & h_{24} & -h_{25} & h_{44} & 0 \\ 0 & h_{25} & h_{24} & 0 & h_{44} \end{pmatrix}_{ij} \\ &= \begin{pmatrix} w_1^2 & 0 & 0 & 0 & 0 \\ 0 & w_2^2 & 0 & w_4 & w_5 \\ 0 & 0 & w_2^2 & -w_5 & w_4 \\ 0 & w_4 & -w_5 & w_3^2 & 0 \\ 0 & w_5 & w_4 & 0 & w_3^2 \end{pmatrix}_{ij}, \end{aligned} \tag{4.5}$$

where we relabel $h_{11}, h_{22}, h_{44}, h_{24}, h_{25} \mapsto w_1^2, w_2^2, w_3^2, w_4, w_5$.

Consider a tangent vector $\dot{p} = (\dot{x}, \dot{n}) = c^i f_i$, that is $\dot{x} = c^1 e_1 + c^2 e_2 + c^3 e_3$ and $\dot{n} = c^4 e_2 + c^5 e_3$. Recall that $n = e_1$. Finally, we can check that $\|(p, \dot{p})\|_{\mathcal{H}}^2$ is indeed of the form (4.3):

$$\begin{aligned}\|(p,\dot{p})\|^2_{\mathcal{H}} &= h_{ij}c^i c^j \\ &= w_1^2 c^1 c^1 + w_2^2(c^2c^2 + c^3c^3) + w_3^2(c^4c^4 + c^5c^5) \\ &+2w_4(c^2c^4 + c^3c^5) + 2w_5(c^2c^5 - c^3c^4) \\ &= w_1^2((c^1e_1 + c^2e_2 + c^3e_3)\cdot e_1)^2 \\ &+w_2^2\|(c^1e_1 + c^2e_2 + c^3e_3)\times e_1\|^2 \\ &+w_3^2\|c^4e_2 + c^5e_3\|^2 \\ &+2w_4(c^1e_1 + c^2e_2 + c^3e_3)\cdot(c^4e_2 + c^5e_3) \\ &+2w_5(c^1e_1 + c^2e_2 + c^3e_3)\cdot((c^4e_2 + c^5e_3)\times e_1) \\ &= \|(p,\dot{p})\|^2_{\mathcal{G}}.\end{aligned} \tag{4.6}$$

Positivity. To prove the positivity constraints we will apply Sylvester's criterion to the matrix (4.5) that encodes the metric. The determinants of the upper-left minors are, respectively, w_1^2, $w_1^2w_2^2$, $w_1^2w_2^4$, $w_1^2w_2^2(w_2^2w_3^2 - w_4^2 - w_5^2)$, and $w_1^2(w_2^2w_3^2 - w_4^2 - w_5^2)^2$. Requiring all of these determinants to be positive and simplifying the resulting system of inequalities yields the single inequality $w_2^2w_3^2 > w_4^2 + w_5^2$. □

Remark 4.1: This result is in contradiction with Smets, Portegies, Bekkers and Duits (2023), Prop.14 where it is claimed that every invariant metric $\mathcal{G}$ yields a norm of the form

$$\|(p,\dot{p})\|^2_{\mathcal{G}} = (w_1|\dot{x}\cdot n|)^2 + (w_2\|\dot{x}\times n\|)^2 + (w_3\|\dot{n}\|)^2. \tag{4.7}$$

Remark 4.2: The positivity constraint $w_2^2w_3^2 > w_4^2 + w_5^2$ seems to be related to

$$\|\dot{x}\times n\|^2\|\dot{n}\|^2 = (\dot{x}\cdot\dot{n})^2 + (\dot{x}\cdot(\dot{n}\times n))^2 \tag{4.8}$$

but the exact relation is unclear to the authors.

Remark 4.3: In a similar fashion one can show that every metric $\mathcal{G}$ that is invariant under the rigid transformation group $\mathrm{E}(3)$ yields a norm of the form

$$\begin{aligned}\|(p,\dot{p})\|_{\mathcal{G}}^2 = {} & (w_1|\dot{x}\cdot n|)^2 + (w_2\|\dot{x}\times n\|)^2 + (w_3\|\dot{n}\|)^2 \\ & + 2w_4\dot{x}\cdot\dot{n},\end{aligned} \tag{4.9}$$

which can be obtained from (4.3) by setting $w_5 = 0$. In fact, this result generalizes to $\mathrm{E}(d)$ invariant metrics on $\mathbb{M}_d$ for *all* $d \geq 2$ by replacing $\|\dot{x}\times n\|$ with $\|\dot{x}\wedge n\|$.

4.3 MAV GENERATOR

In this section, we describe how one can obtain the minimal angular velocity (mav) generator $M(p_1,p_2) \in \mathfrak{se}(3)$ between two position-orientations $p_1, p_2 \in \mathbb{M}_3$. In Figure 4.1 the curve $\eta(t) := \exp(Mt) \triangleright p_1$ of the mav generator M is shown, alongside the curves of two other generators.

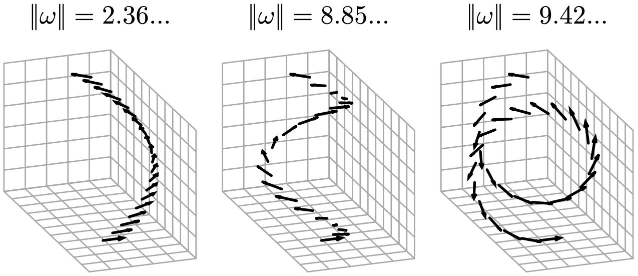


Figure 4.1: Visualization of the curve $\eta(t) := \exp(Xt) \triangleright p_1$ connecting the position-orientations $\eta(0) = p_1$ and $\eta(1) = p_2$, for three different generators $X \in \mathfrak{se}(3)$. The angular velocity $\|\omega\|$ of the generator $X = (v,\omega)$ is indicated above the plots. The leftmost plot corresponds to the mav generator $X = M(p_1,p_2)$, which, by definition, has minimal angular velocity.

We will proceed as follows. First, we start by constructing the planar roto-translation $S = \exp(M) \in \mathrm{SE}(3)$, that being the exponential of the mav generator. Second, we rewrite the planar roto-translation S as a screw displacement. Third, we find a generator of the obtained screw displacement, this being the mav generator. We take this approach because identifying a generator of a screw displacement is more straightforward.

Although the planar roto-translation was already described in Portegies, Sanguinetti, Meesters and Duits (2015), we provide a new geometric perspective and construction, combined with a way to efficiently calculate the corresponding mav generator.

Definition 4.1 (Planar Roto-Translation & Mav Generator): Let $p_1 = (x_1, n_1), p_2 = (x_2, n_2) \in \mathbb{M}_3$ be two position-orientations.

The *planar roto-translation* $S(p_1, p_2) \in \mathrm{SE}(3)$ from p_1 to $S \rhd p_1 = p_2$ is the unique roto-translation whose rotation is purely in the plane spanned by the orientations n_1 and n_2.

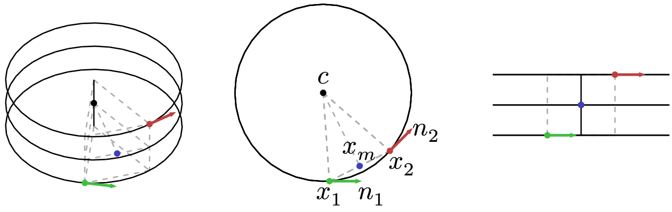
The *mav generator* $M(p_1, p_2) \in \mathfrak{se}(3)$ is the unique generator of the planar roto-translation (so $\exp M = S$) with minimal angular velocity.

Explicitly constructing the planar roto-translation S can be done as follows. Let $N = \mathrm{span}\ \{n_1, n_2\}$ be the rotation plane, $\theta = \arccos(n_1 \cdot n_2) \in [0, \pi]$ the rotation angle, $L = (n_2 n_1^\top - n_1 n_2^\top) / \sin(\theta)$ the unit radian rotation generator in N, $\omega = \theta L$ the rotation generator, and $R = \exp \omega$ the corresponding rotation matrix. Using Rodrigues' rotation formula we can simplify the exponential to $R = \exp \omega = I + \sin(\theta) L + (1 - \cos(\theta)) L^2$. The planar roto-translation is then $S : x \mapsto R(x - x_1) + x_2$.

We can now explicitly construct the mav generator M. To find it we will write the planar roto-translation as a *screw displacement*.

Definition 4.2 (Screw Displacement): A *screw displacement* is a roto-translation in the form of $x \mapsto c + R(x - c) + t$ with $Rt = t$. That is, the translational part t is orthogonal to the *plane of rotation*, or, equivalently, the translational part is parallel to the *axis of rotation*. We call c the *center of rotation* of the screw displacement.

Every roto-translation can be written as a screw displacement, this is known as Chasles' theorem (Heard (2005)). More specifically, we can write the planar roto-translation as a screw displacement as follows. Let $\vec{x} = x_2 - x_1$ be the vector from x_1 to x_2, $x_m = (x_1 + x_2)/2$ the midpoint, and decompose $\vec{x} = \vec{x}_\| + \vec{x}_\perp$ into a parallel and perpendicular part with respect to the rotation plane N. Define $c = x_m + \frac{1}{2}\cot\left(\frac{\theta}{2}\right)L\vec{x}_\|$. The *planar screw displacement* is then $x \mapsto c + R(x - c) + \vec{x}_\perp$. The situation is visualized in Figure 4.2, and explains geometrically how we obtained the formula for the center of rotation c.



Overview. Top view. Front view.

Figure 4.2: Diagram of the center of rotation c of the planar screw displacement between two position-orientations $p_1 = (x_1, n_1)$, $p_2 = (x_2, n_2) \in \mathbb{M}_3$. The circles are aligned with the rotation plane N. For the sake of clarity, only in the top view the labels have been placed.

Finding a generator for a screw displacement $x \mapsto c + R(x - c) + t$ is straightforward. Namely, a generator is $x \mapsto \omega(x - c) + v$, where ω is a generator of R in $\mathrm{SO}(3)$, and $v = t$ is the generator of t in $\mathbb{R}^3$. Particularly, the mav generator M for the planar screw displacement S is $x \mapsto \omega(x - c) + \vec{x}_\perp$, where $\omega = \theta L$.

So, all in all, we have the following explicit form for the mav generator.

Proposition 4.1 (Explicit Mav Generator): The *mav generator* $M(p_1, p_2) \in \mathfrak{se}(3)$ between two position-orientations $p_1 = (x_1, n_1), p_2 = (x_2, n_2) \in \mathbb{M}_3$ is

$$M = (-\omega c + v, \omega) \tag{4.10}$$

where $\omega = \theta L$, $c = x_m + \frac{1}{2}\cot\left(\frac{\theta}{2}\right)L\vec{x}_{\parallel}$, $v = \vec{x}_{\perp}$, $\theta = \arccos(n_1 \cdot n_2)$, $L = \left(n_2 n_1^\top - n_1 n_2^\top\right) / \sin(\theta)$, $x_m = (x_1 + x_2)/2$, and $\vec{x} = x_2 - x_1$.

4.4 EXPERIMENTS

The standard PONITA architecture builds upon 3 scalar SE(3) invariants:

$$\begin{aligned} \iota_1(p_1, p_2) &= (x_2 - x_1) \cdot n_1 \\ \iota_2(p_1, p_2) &= \|(x_2 - x_1) - \iota_1(p_1, p_2) n_1\| \\ \iota_3(p_1, p_2) &= \arccos(n_1 \cdot n_2) \end{aligned} \tag{4.11}$$

where $p_1 = (x_1, n_1)$, $p_2 = (x_2, n_2) \in \mathbb{M}_3$. We compare this architecture against one where these 3 invariants are *completely* replaced by the mav distance $\mu_{\mathcal{G}}$:

$$\mu_{\mathcal{G}}(p_1, p_2) = \|M(p_1, p_2) \triangleright p_1\|_{\mathcal{G}}, \tag{4.12}$$

where $\mathcal{G}$ is the SE(3) invariant metric on $\mathbb{M}_3$ determined by the learnable metric parameters $w_1, \ldots, w_5$ as described in Theorem 4.1, and $M(p_1, p_2)$ the mav generator as described in Proposition 4.1. We investigate if this replacement has a positive impact on the accuracy of PONITA.

Through a reparameterization of the metric parameters w_4 and w_5 one can enforce the (semi)positivity constraint $w_2^2 w_3^2 \geq w_4^2 + w_5^2$. For example, one can define $w_4 = a_4\delta$ and $w_5 = a_5\delta$, where $\delta = 2w_2w_3/(1 + a_4^2 + a_5^2)$. However, in practice, enforcing these constraints yielded no noticeable benefits. We therefore left the metric parameters unconstrained, allowing the mav “distance” to take on negative values. This modification preserves the $\mathrm{SE}(3)$ invariance of the mav distance.

We performed some experiments on the QM9 dataset (Subsection 2.8.4), where the task is to predict properties of various molecules. We choose to discretize with 16 orientations, use 6 layers, 128-dimensional features, and train for 800 epochs. All other model settings and hyperparameters are kept the same as in Bekkers, Vadgama, Hesselink, van der Linden and Romero (2024), App.E.2. We report the mean absolute error on the test set with the model that did best on the validation set during training. The code for this experiment is available in the GitLab repository PONITA Invariants. The results can be found in Table 5.1.

We see that the use of the mav distance within PONITA has a positive impact on accuracy on 7 of the 12 targets we experimented on, with an average improvement of −4.0%.

Target	Unit	Bekkers (4.11)	Mav (4.12)	Diff. %
μ	D	0.0195	0.0181	−07.2
α	a_0^3	0.0556	0.0540	−02.9
$\varepsilon_{\text{homo}}$	eV	0.0225	0.0229	+01.8
$\varepsilon_{\text{lumo}}$	eV	0.0205	0.0207	+01.0
$\Delta\varepsilon$	eV	0.0414	0.0431	+04.0
$\langle R^2 \rangle$	a_0^2	0.4160	0.4942	+18.8
ZPVE	meV	1.5647	1.5613	−00.2
U_0	eV	0.9920	0.7047	−28.9
U	eV	1.3593	1.0947	−19.5
H	eV	1.0204	1.0856	+06.4
G	eV	1.1856	0.9691	−18.3
c_v	$\frac{\text{cal}}{\text{mol K}}$	0.0291	0.0283	−02.8

Table 4.1: PONITA trained to predict chemical properties of various molecules (QM9 dataset). Mean absolute error on the test set is reported (lower is better).

4.5 CONCLUSION

In Section 4.2 we classified and parametrized all SE(3) invariant Riemannian metric tensor fields $\mathcal{G}$ on $\mathbb{M}_3$. In Section 4.3 we explicitly constructed the mav generator $M(p_1, p_2) \in \mathfrak{se}(3)$ between two position-orientations $p_1, p_2 \in \mathbb{M}_3$. These two results allowed us to use the mav distance $\mu_{\mathcal{G}}(p_1, p_2)$ as a trainable invariant in the PONITA architecture.

In Section 4.4 we performed an experimental comparison between the mav distance (4.12) and the invariants proposed in Bekkers et al. (4.11). We did this by training the PONITA architecture to predict chemical properties of various molecules (QM9 dataset, Subsection 2.8.4). We observe marginal improvements in accuracy when using the mav distance.

INVARIANTS BETWEEN PAIRS OF POSITION-ORIENTATIONS 5

5.1 INTRODUCTION

To create $\mathrm{E}(3)$ equivariant neural networks on $\mathbb{M}_3$, one is motivated to create and study scalar $\mathrm{E}(3)$ invariant kernels $k : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$, that being functions with the property that

$$k(g \triangleright p_1, g \triangleright p_2) = k(p_1, p_2), \tag{5.1}$$

for all $p_1, p_2 \in \mathbb{M}_3$ and $g \in \mathrm{E}(3)$. Namely, these invariant kernels k are used to perform equivariant processing of scalar fields $f : \mathbb{M}_3 \to \mathbb{R}$, a core ingredient in any equivariant architecture, such as PONITA (Section 2.7).

Consider any collection of scalar invariants $\iota_1, \ldots, \iota_n : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$. We can create a new invariant ι' easily by considering any function $h : \mathbb{R}^n \to \mathbb{R}$ and defining $\iota' = h(\iota_1, \ldots, \iota_n)$. This observation has an immediate application in our architectures: we can decide to parameterize the kernels k by, for example, a multi-layer perceptron $\mathrm{MLP}_\theta : \mathbb{R}^n \to \mathbb{R}$ with *trainable* parameters θ, and inserting a predesigned collection of n invariants: $k = \mathrm{MLP}_\theta(\iota_1, \ldots, \iota_n)$. This motivates looking into what an "optimal" collection of scalar invariants would be, so that we can construct networks that are maximally expressive and efficient.

Suppose we have a collection of invariants where one of them is a function of the others. If this happens we say the collection of invariants is *dependent* (Definition 5.3). A dependent collection is not "optimal" in the sense that we could remove the dependent invariant and (theoretically) lose no expressiveness.

On the other hand, suppose we have a collection of invariants for which we know that any other invariant one can think of is a function of them. We say such a collection of invariants is *universal*

(Definition 5.2). A universal collection of invariants is “optimal” in the sense that there is no reason to add another invariant because we (theoretically) gain no expressiveness.

This motivates our search for a collection of scalar invariants that is both independent and universal, which will serve as a type of basis for invariant kernels.

The PONITA architecture (Section 2.7) utilizes the following collection of three $\mathrm{E}(3)$ invariants $\iota_i : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$:

$$\begin{aligned} \iota_1(p_1, p_2) &= (x_2 - x_1) \cdot n_1, \\ \iota_2(p_1, p_2) &= \|(x_2 - x_1) - \iota_1(p_1, p_2) n_1\|, \\ \iota_3(p_1, p_2) &= \arccos(n_1 \cdot n_2), \end{aligned} \tag{5.2}$$

where $p_1 = (x_1, n_1)$, $p_2 = (x_2, n_2) \in \mathbb{M}_3 = \mathbb{R}^3 \times S^2$. However, we noticed that these invariants are *not* universal, see Remark 5.2, meaning that not all kernels can be written in terms of this collection of invariants, which could impact the accuracy of PONITA negatively in certain applications.

In this chapter, we theoretically investigate in detail what an independent and universal collection of scalar $\mathrm{E}(3)$ invariants on $\mathbb{M}_3 \times \mathbb{M}_3$ would be, and whether such a collection can be used to improve the accuracy of PONITA in practice.

5.1.1 CONTRIBUTIONS

In Section 5.4 we motivate theoretically that one needs four smooth scalar $\mathrm{E}(3)$ invariants on $\mathbb{M}_3 \times \mathbb{M}_3$ to create an independent and universal collection.

In Definition 5.5 we give our own collection of four smooth $\mathrm{E}(3)$ invariants $\iota_i : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$, of which we show in Theorem 5.1 that it is both independent and universal.

In Section 5.6 we show that using a universal collection of invariants has a significant positive impact on the accuracy of PONITA when predicting molecular properties.

5.1.2 OUTLINE

- In Section 5.2 we define universality and the existence of a representer, and show that the latter implies the former.
- In Section 5.3 we define (in)dependence and show that invariants that form a submersion are independent.
- In Section 5.4 we theoretically analyze the orbits of $\mathbb{M}_3 \times \mathbb{M}_3$ under the action of $\mathrm{E}(3)$, leading us to conclude that four smooth scalar invariants are necessary to form an independent and universal collection.
- In Section 5.5 we present a collection of four smooth scalar invariants, along with proofs of its independence and universality.
- In Section 5.6 we evaluate our universal collection of invariants using the PONITA architecture.
- In Section 5.7 we conclude this chapter.

5.2 UNIVERSALITY

Here, we define what an invariant in our context, what universality is, and clarify what it means for an invariant to have a representer. We will demonstrate in Lemma 5.1 that the existence of a representer implies universality, a result we will later use to prove that our collection of invariants (5.9) is universal.

The following definitions are stated for arbitrary sets X and groups G. Although this level of generality exceeds our needs – since our focus will be on $X = \mathbb{M}_3 \times \mathbb{M}_3$ and $G = \mathrm{E}(3)$ – we present the full definition here for the sake of completeness.

Definition 5.1 (Invariant): Let X be a set and G a group acting on it. Define $\sim$ as the equivalence relation indicating if two elements are in the same orbit, that is $x \sim x' \Leftrightarrow \exists g \in G : gx = x'$. Let Y be any set. An *invariant* $\iota : X \to Y$ is a mapping such that $x \sim x' \Rightarrow \iota(x) = \iota(x')$.

Definition 5.2 (Universality and Representer):

- An invariant $\iota : X \to Y$ is *universal* if any other invariant $j : X \to Z$ can be written as a function of it, that is there exists $F : \operatorname{im} \iota \to Z$ such that $j = F \circ \iota$.
- A *representer* φ for an invariant $\iota : X \to Y$ is a map $\varphi : \operatorname{im} \iota \to X$ such that $\varphi(\iota(x)) \sim x$ for all $x \in X$.

In other words, a representer φ takes in the value $\iota(x)$ of the invariant ι of an $x \in X$ and returns an $x' = \varphi(\iota(x))$ that is in the orbit of x.

Lemma 5.1: Let ι be an invariant. If ι has a representer then ι is universal.

Proof: Let φ be a representer for ι and let j be any other invariant. Now consider $F = j \circ \varphi$. One can check that, indeed, $j = F \circ \iota$, thus showing that ι is universal. □

Remark 5.1: Even with a universal collection of invariants, we cannot perfectly represent arbitrary invariants in practice, as expressivity is constrained by the chosen architecture. For instance, if our invariants are continuous and we use multi-layer perceptrons with continuous activation functions to parametrize the kernels, we can only represent continuous invariants, meaning we exclude any with discontinuities.

Remark 5.2: We can show that the invariants (5.2) used in BEKKERS, VADGAMA, HESSELINK, VAN DER LINDEN AND ROMERO (2024) are not universal as follows. Let e_1, e_2, e_3 be the standard basis of $\mathbb{R}^3$. Consider the following two pairs of position-orientations:

$$\begin{aligned} (p_1, p_2) &= ((0, e_3), (e_1, e_2)) \\ (q_1, q_2) &= ((0, e_3), (e_1, e_1)). \end{aligned} \tag{5.3}$$

As per (5.2), we calculate their invariants to be

$$\begin{aligned}\iota_1(p_1,p_2) &= \iota_1(q_1,q_2) = 0\\ \iota_2(p_1,p_2) &= \iota_2(q_1,q_2) = 1\\ \iota_3(p_1,p_2) &= \iota_3(q_1,q_2) = \frac{\pi}{2}.\end{aligned} \tag{5.4}$$

We see that all invariants agree between the two pairs. Consider the invariant $\iota = (x_2 - x_1) \cdot n_2$. We calculate that $\iota(p_1, p_2) = 0$ and $\iota(q_1, q_2) = 1$, so ι can not be a function of ι_1, ι_2, and ι_3. Thereby, we conclude that (5.2) is not a universal collection of invariants.

5.3 DEPENDENCE

In this section, we will define what it means for a collection of invariants to be (in)dependent. We will show in Lemma 5.2 that a collection of invariants that form a submersion is independent, a result we will later use to prove that our collection of invariants (5.9) is independent.

Definition 5.3 (Dependent): A collection of scalar invariants $\iota_1, \ldots, \iota_n : X \to \mathbb{R}$ is *dependent* over X if one of them can be written as a function of the others on X, and *independent* otherwise.

If a collection of invariants is dependent on a set X then it is also dependent on any subset $X' \subseteq X$. Contrapositively, if a collection of invariants is independent on a set X' it is also independent on any superset $X \supseteq X'$.

This observation raises an issue in our definition of independence: it is easy to create a “large” set on which the invariants are technically independent; one only needs to include a “small” subset on which this is the case. To remedy this we introduce the following stronger definition.

Definition 5.4 (Somewhere Dependent): Let X be a topological space and $\iota : X \to \mathbb{R}^n$ a collection of scalar invariants. We say ι is *somewhere dependent* if there exists an open subset $U \subseteq X$ such that ι is dependent on U, and *everywhere independent* otherwise.

It might seem hard to prove that a collection of scalar invariants is everywhere independent. However, if we specialize our setting by considering spaces that are differentiable manifolds and invariants that are differentiable maps, we have the following simpler sufficient condition.

Lemma 5.2: Let M be a differentiable manifold and $\iota : M \to \mathbb{R}^n$ a collection of differentiable scalar invariants that form a *submersion*, that is the differential $\mathrm{d}\iota|_p : T_pM \to \mathbb{R}^n$ is a surjective linear map at every point $p \in M$. Then ι is everywhere independent.

Proof: A submersion is an *open map* (Lee (2012), Prop.4.28), meaning that it maps open sets to open sets. So, given any arbitrary open subset $U \subseteq M$ the image $\iota(U) \subseteq \mathbb{R}^n$ is open. There is no functional relation between all elements of an open subset in $\mathbb{R}^n$ (such an open set is never the graph of a function), thus ι is independent on U. As U was arbitrary, this shows that ι is everywhere independent. ☐

Remark 5.3: In practice, constructing a network using a dependent collection of invariants poses no issues, as the redundancy does not hinder training and may even enhance stability. However, proving the independence of a universal collection remains valuable, as it implies that removing *any* invariant renders the collection *non*-universal: the invariant that is removed can not be written as a function of the remaining invariants.

5.4 How Many Invariants Do We Need?

In this part, we will explore how many smooth scalar $\mathrm{E}(3)$ invariants on $\mathbb{M}_3 \times \mathbb{M}_3$ one needs to form a collection that is both independent and universal.

We start by investigating the orbit space; the invariants are constant on the orbits after all. To save on notation, let $G = \mathrm{E}(3)$, $\mathfrak{g} = \mathfrak{e}(3)$, and $M = \mathbb{M}_3 \times \mathbb{M}_3$ for the moment. Pick any $p \in M$. We are interested in the nature of the orbit $Gp = \{gp \mid g \in G\} \subseteq M$.

We can understand the orbit Gp as the image of the map $\varphi : G \to M$ defined by $\varphi(g) = gp$. But more precisely, we can reinterpret φ as $\varphi : G/H \to M$, where G/H is the coset space made from the closed stabilizer Lie subgroup $H = \operatorname{stab} p \subseteq G$ of p, which can be endowed with the structure of a manifold (Lee (2012), Thm.21.17). So, even more precisely, $\varphi : G/H \to M$ is an *injective immersion*. This means that the orbit Gp is an *injectively immersed submanifold* and its dimension is equal to the dimension of G/H. We have that $\dim G/H = \dim G - \dim H$, so to find the dimension of G/H we need to find the dimension of H. To find the dimension of H we will study its Lie algebra $\mathfrak{h} \subseteq \mathfrak{g}$.

The Lie group H of stabilizers consists of all group elements $h \in G$ for which $hp = p$. It follows that its Lie algebra $\mathfrak{h}$ consists of all the generators $X \in \mathfrak{g}$ for which $Xp = 0$. Let us write $X = (v, \omega) \in \mathfrak{g}$, $p = (p_1, p_2)$, $p_1 = (x_1, n_1)$, and $p_2 = (x_2, n_2)$. With this notation we expand Xp and set it zero

$$\begin{aligned} &(v,\omega)((x_1,n_1),(x_2,n_2)) \\ &= ((v+\omega x_1, \omega n_1),(v+\omega x_2, \omega n_2)) \\ &= 0. \end{aligned} \tag{5.5}$$

We identify the rotation generator $\omega \in \mathbb{R}^{3\times 3}$ with its angular velocity vector $\vec{\omega} \in \mathbb{R}^3$ and rewrite the above as:

$$v = -\vec{\omega} \times x_1 = -\vec{\omega} \times x_2, \vec{\omega} \times n_1 = 0, \text{ and } \vec{\omega} \times n_2 = 0. \tag{5.6}$$

The first equation also implies that $\vec{\omega} \times (x_2 - x_1) = 0$. We see that if and only if $\vec{\omega}$ is (anti)parallel to n_1, n_2, and $x_2 - x_1$, it is a solution to this system of equations (where the zero vector is parallel to everything). We thus classify two cases for $\mathfrak{h}$:

$$\mathfrak{h} = \begin{cases} \text{span } \{(-n_1 \times x_1, [n_1]_\times)\} \text{ if } \; n_1 \parallel n_2 \parallel x_2 - x_1 \\ \{0\} \text{ otherwise} \end{cases} \tag{5.7}$$

where $[n_1]_\times \in \mathbb{R}^{3\times 3}$ is the unique antisymmetric matrix such that $[n_1]_\times y = n_1 \times y$ for all $y \in \mathbb{R}^3$, and $(v, \omega) = (-n_1 \times x_1, [n_1]_\times) \in \mathfrak{e}(3)$.

Translating the cases of $\mathfrak{h}$ back to the dimension of the orbit we get that

$$\begin{aligned} \dim Gp &= \dim G/H \\ &= \dim G - \dim \mathfrak{h} \\ &= \begin{cases} 5 \text{ if } \; n_1 \parallel n_2 \parallel x_2 - x_1 \\ 6 \text{ otherwise} \end{cases} \end{aligned} \tag{5.8}$$

We collect the above derivations and results in the following proposition.

Proposition 5.1: The orbits of $\mathbb{M}_3 \times \mathbb{M}_3$ under the action of E(3) are injectively immersed submanifolds. Points $((x_1, n_1), (x_2, n_2)) \in \mathbb{M}_3 \times \mathbb{M}_3$ for which n_1, n_2, and $x_2 - x_1$ are all parallel, form 5-dimensional orbits. All other orbits are 6 -dimensional.

Notice that the set of all points for which $n_1 \parallel n_2 \parallel x_2 - x_1$ is *closed* in M. Equivalently, the set of all points with a 6-dimensional orbit is *open*.

So, (at least) locally in a neighborhood $U \subseteq M$ around a point $p \in M$ with $\dim Gp = 6$, we in fact have a 6-dimensional *foliation*, that being a disjoint collection of injectively immersed submanifolds of the same dimension, also called *leaves*, that partition the 10-dimensional neighborhood U of M.

Given this foliation, it is known that we can find coordinates $x^1, \ldots, x^{10}$ on U such that every leaf is described by $x^7 =$ constant, ..., $x^{10} =$

constant. Notice that these last four coordinates are *exactly* the kinds of invariants we are looking for: they are smooth, everywhere independent, and universal (on U, that is), precisely because they serve as coordinates.

All this theory tells us something important: we should be looking for exactly $10 - 6 = 4$ smooth, universal & everywhere independent $\mathrm{E}(3)$ invariants on, possibly an open subset of, $\mathbb{M}_3 \times \mathbb{M}_3$. In the case that the domain of definition is not all of $\mathbb{M}_3 \times \mathbb{M}_3$, it would be desirable for it to be dense in the whole space, thus making the collection of invariants still "generically" applicable.

5.5 OUR INVARIANTS

Definition 5.5 (Our Invariants): Write $p_1 = (x_1, n_1)$, $p_2 = (x_2, n_2) \in \mathbb{M}_3$. We define the following smooth functions $\iota_i : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$:

$$\begin{aligned} \iota_1(p_1, p_2) &= (x_2 - x_1) \cdot n_1, \\ \iota_2(p_1, p_2) &= (x_2 - x_1) \cdot n_2, \\ \iota_3(p_1, p_2) &= (x_2 - x_1) \cdot (x_2 - x_1), \\ \iota_4(p_1, p_2) &= n_1 \cdot n_2. \end{aligned} \tag{5.9}$$

We will prove that
- the maps ι_i are invariants,
- they form a universal collection,
- and that they are independent everywhere.

These results are collected in the following theorem.

Theorem 5.1: The functions (5.9) form a universal and everywhere independent collection of $\mathrm{E}(3)$ invariants on $\mathbb{M}_3 \times \mathbb{M}_3$.

Proof: We will begin by proving that the functions ι_i are invariants, then demonstrate their universality, and finally show that they are independent everywhere.

Invariance: To prove that these are invariants we must show that each $\iota_i(p_1, p_2) = \iota_i(gp_1, gp_2)$ for all pairs of position-orientations $(p_1, p_2) \in \mathbb{M}_3 \times \mathbb{M}_3$ and all rigid transformations $g = (t, Q) \in \mathrm{E}(3)$. This is a straightforward calculation using that $Qx \cdot Qy = x \cdot y$ for all $x, y \in \mathbb{R}^3$ and orthogonal matrices $Q \in \mathrm{O}(3)$.

Universality: Let $p_1 = (x_1, n_1)$, $p_2 = (x_2, n_2) \in \mathbb{M}_3$ be a pair of position-orientations. From the values of the invariants $\iota_1(p_1, p_2), \ldots, \iota_4(p_1, p_2)$ we will construct another pair of position-orientations $\bar{p}_1, \bar{p}_2$ such that $(\bar{p}_1, \bar{p}_2) \sim (p_1, p_2)$. The described construction will act as our representer, thus showing that the collection of invariants is universal, as per Lemma 5.1.

Consider the vectors $v_1 = n_1, v_2 = n_2, v_3 = x_2 - x_1$. Their Gram matrix is

$$v_i \cdot v_j = \begin{pmatrix} 1 & \iota_4 & \iota_1 \\ \iota_4 & 1 & \iota_2 \\ \iota_1 & \iota_2 & \iota_3 \end{pmatrix}_{ij}, \text{ with } \iota_k = \iota_k(p_1, p_2), \tag{5.10}$$

meaning that we have access to this Gram matrix purely from the values of the invariants. Let $\bar{v}_1, \bar{v}_2, \bar{v}_3$ be *any* list of vectors with the same Gram matrix. Lists of vectors with the same Gram matrix are related by an orthogonal matrix Q, i.e. $Q\bar{v}_i = v_i$. Now pick *any* position $\bar{x}_1$, and define $\bar{n}_1 = \bar{v}_1$, $\bar{n}_2 = \bar{v}_2$, and $\bar{x}_2 = \bar{x}_1 + \bar{v}_3$. We check that the rigid transformation $g = (-Q\bar{x}_1 + x_1, Q) \in \mathrm{E}(3)$ maps $(\bar{p}_1, \bar{p}_2)$ to (p_1, p_2), showing that $(\bar{p}_1, \bar{p}_2) \sim (p_1, p_2)$. For example

$$\begin{aligned} g \triangleright \bar{p}_2 &= (-Q\bar{x}_1 + x_1 + Q\bar{x}_2, Q\bar{n}_2) \\ &= (Q(\bar{x}_2 - \bar{x}_1) + x_1, Q\bar{n}_2) \\ &= (Q\bar{v}_3 + x_1, Q\bar{v}_2) \\ &= (v_3 + x_1, v_2) \\ &= (x_2 - x_1 + x_1, n_2) \\ &= p_2, \end{aligned} \tag{5.11}$$

and similarly one finds $g \triangleright \bar{p}_1 = p_1$.

Independence: To prove that this collection of invariants is everywhere independent we will show that the differentials $\mathrm{d}\iota_i|_p : T_p(\mathbb{M}_3 \times \mathbb{M}_3) \to \mathbb{R}$ are linearly independent on the following dense and open subset U of $\mathbb{M}_3 \times \mathbb{M}_3$:

$$U := \{(p_1, p_2) \mid x_2 - x_1, n_1, n_2 \text{ form a basis of } \mathbb{R}^3\}, \quad (5.12)$$

where $p_1 = (x_1, n_1)$, $p_2 = (x_2, n_2) \in \mathbb{M}_3$. This shows that $\mathrm{d}\iota|_p$ is surjective everywhere on U, meaning that ι is a submersion from U to $\mathbb{R}^4$, and therefore everywhere independent, as per Lemma 5.2. In short, it suffices to show that $\mathrm{d}\iota_i|_p$ are independent on U, which we will do next.

Let $\dot{p}_1 = (\dot{x}_1, \dot{n}_1) \in T_{p_1}\mathbb{M}_3$, $\dot{p}_2 = (\dot{x}_2, \dot{n}_2) \in T_{p_2}\mathbb{M}_3$. The differentials $\mathrm{d}\iota_i|_p$ are

$$\begin{aligned} \mathrm{d}\iota_1|_p\ (\dot{p}_1, \dot{p}_2) &= (\dot{x}_2 - \dot{x}_1) \cdot n_1 + (x_2 - x_1) \cdot \dot{n}_1, \\ \mathrm{d}\iota_2|_p\ (\dot{p}_1, \dot{p}_2) &= (\dot{x}_2 - \dot{x}_1) \cdot n_2 + (x_2 - x_1) \cdot \dot{n}_2, \\ \mathrm{d}\iota_3|_p\ (\dot{p}_1, \dot{p}_2) &= 2(\dot{x}_2 - \dot{x}_1) \cdot (x_2 - x_1), \\ \mathrm{d}\iota_4|_p\ (\dot{p}_1, \dot{p}_2) &= \dot{n}_1 \cdot n_2 + n_1 \cdot \dot{n}_2. \end{aligned} \quad (5.13)$$

Suppose that we could find coefficients $c^i \in \mathbb{R}$ such that $\sum_{i=1}^4 c^i\, \mathrm{d}\iota_i|_p = 0$ (that being for *all* tangents $\dot{p}_1, \dot{p}_2$). Specifically, take $\dot{x}_1 = \dot{n}_1 = \dot{n}_2 = 0$. We find $\left(c^1 n_1 + c^2 n_2 + 2c^3(x_2 - x_1)\right) \cdot \dot{x}_2 = 0$ for *all* $\dot{x}_2$. This implies that $c^1 n_1 + c^2 n_2 + 2c^3(x_2 - x_1) = 0$. Because we have assumed that these three vectors form a basis, we get that $c^1 = c^2 = c^3 = 0$. Now consider $\dot{x}_1 = \dot{x}_2 = \dot{n}_2 = 0$. We find $\left(c^1(x_2 - x_1) + c^4 n_2\right) \cdot \dot{n}_1 = 0$ for *all* $\dot{n}_1$ with $\dot{n}_1 \cdot n_1 = 0$. This implies that $c^1(x_2 - x_1) + c^4 n_2$ is in the span of n_1. Again, because we have assumed that these three vectors form a basis, we have to conclude that $c^1 = c^4 = 0$. All in all, we find that $\sum_{i=1}^4 c^i\, \mathrm{d}\iota_i|_p = 0$ implies that all $c^i = 0$, thus showing that the differentials are linearly independent. ☐

Remark 5.4: In tasks like drug discovery, it is important to design a network to be sensitive to *chirality*, as a molecule and its mirror image can have entirely different effects. In mathematical terms, this means the focus is on $\mathrm{SE}(3) := \mathbb{R}^3 \rtimes \mathrm{SO}(3)$ rather than the full group $\mathrm{E}(3)$. By using an $\mathrm{SE}(3)$ invariant that is *not* invariant under reflections, such as $\iota = (x_2 - x_1) \cdot (n_1 \times n_2)$, it becomes possible to design a network that respects chirality.

5.6 EXPERIMENTS

By Remark 5.2, the invariants (5.2) used in PONITA are not universal, which may limit its expressivity and accuracy in specific applications. In contrast, the invariants we propose (5.9) are universal, suggesting that, in theory, a performance difference between the two collections of invariants should be observable.

For this reason we performed some experiments on the QM9 dataset (Subsection 2.8.4) specifically predicting chemical properties of various molecules. We choose to discretize with 16 orientations, use 6 layers, 128-dimensional features, and train for 800 epochs. All other model settings and hyperparameters are kept the same as in BEKKERS, VADGAMA, HESSELINK, VAN DER LINDEN AND ROMERO (2024), App.E.2. We report the mean absolute error (MAE) on the test set for the model that did best on the validation set during training. The code for this experiment is available in the GitLab repository PONITA INVARIANTS. The results can be found in Table 5.1.

We see that the universal collection of invariants we propose (5.9) outperforms the non-universal invariants suggested in (5.2) on 10 of the 12 targets we evaluated, with an average improvement of −14.4%.

On some targets ($\langle R^2 \rangle$, ZPVE, and U_0) the change in accuracy is less pronounced, in two cases even deteriorating. One explanation is that a universal collection may be unnecessary for these targets, meaning the Bekkers et al. invariants are already sufficiently expressive. The remaining variability in accuracy can be attributed to initialization and training.

On other targets (U, H, and G) the improvement in accuracy is stark. We can identify this as a direct consequence of using a universal collection of invariants: these targets cannot be accurately predicted using the incomplete invariants of Bekkers et al.

Target	Unit	Bekkers (5.2)	Ours (5.9)	Diff. %
μ	D	0.0195	0.0166	−15.0
α	a_0^3	0.0557	0.0489	−12.1
$\varepsilon_{\text{homo}}$	eV	0.0226	0.0202	−10.4
$\varepsilon_{\text{lumo}}$	eV	0.0206	0.0187	−9.0
$\Delta\varepsilon$	eV	0.0415	0.0378	−8.9
$\langle R^2 \rangle$	a_0^2	0.4160	0.4251	+2.2
ZPVE	meV	1.5647	1.5241	−2.6
U_0	eV	0.9920	1.0285	+3.7
U	eV	1.3593	0.7362	−45.8
H	eV	1.0205	0.6934	−32.1
G	eV	1.1856	0.7721	−34.9
c_v	$\frac{\text{cal}}{\text{mol K}}$	0.0292	0.0270	−7.4

Table 5.1: PONITA trained to predict chemical properties of various molecules (QM9 dataset). Mean absolute error on the test set is reported (lower is better). Our universal invariants perform better.

5.7 Conclusion

In Definition 5.5 we introduced a collection of four smooth $\mathrm{E}(3)$ invariants $\iota_i : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$. We proved in Theorem 5.1 that this collection is both independent and universal, meaning that all invariants are pertinent, and any other invariant is a function of them.

In Section 5.6 we performed an experimental comparison between our collection of invariants (5.9) and those proposed in Bekkers et al., as defined in (5.2). We did this by training the PONITA architecture to predict chemical properties of various molecules (QM9 dataset, Subsection 2.8.4). We observe improvements in accuracy as a result of using a universal set of invariants.

GEOMETRIC ADAPTATIONS OF PDE-BASED NEURAL NETWORKS 6

6.1 INTRODUCTION

In this chapter we consider two geometric adaptations of PDE-based neural networks (Section 2.6) on the two-dimensional position-orientation space $\mathbb{M}_2$.

The first geometric adaptation we consider is the fixing of the lifting layer. Currently, in LIETORCH the lifting is done using learned kernels. We motivate theoretically that, within the PDE-G-CNN framework, a learned lifting layer can be replaced by a fixed lifting layer using *cake wavelets* (Section 2.3). We experimentally substantiate this finding by reporting that the accuracy of PDE-G-CNNs does not decrease when the lifting layer is fixed. Moreover, using cake wavelets for lifting has the important benefit of increasing the geometric interpretation of PDE-G-CNNs.

The second geometric adaptation is the implementation of a more "free" SE(2) invariant Riemannian metric $\mathcal{G}$ on $\mathbb{M}_2$ than the one that is used in the PDE-G-CNN literature. In SMETS, PORTEGIES, BEKKERS AND DUITS (2023) only *diagonal* metrics with respect to the natural SE(2) invariant frame $\mathcal{A}_i$ (2.32) are used, that is

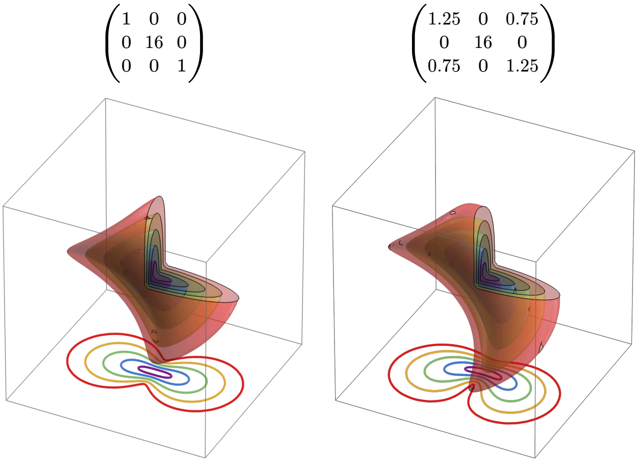
$$\mathcal{G}(\mathcal{A}_i, \mathcal{A}_j) = \begin{pmatrix} w_1^2 & 0 & 0 \\ 0 & w_2^2 & 0 \\ 0 & 0 & w_3^2 \end{pmatrix}_{ij} \tag{6.1}$$

Using *nondiagonal* metrics, that is, metrics of the form

$$\mathcal{G}(\mathcal{A}_i, \mathcal{A}_j) = G_{ij}, \tag{6.2}$$

for any symmetric positive definite matrix $G \in \mathbb{R}^{3\times3}$, has been beneficial in geometric image processing (contour perception AUGUST AND ZUCKER (2003), denoising SMETS, PORTEGIES, ST-ONGE AND DUITS

(2021), segmentation Zhang, Dashtbozorg, Bekkers, Pluim, Duits and ter Haar Romeny (2016), and tracking Liu, Wang, Zhou, Shu, Cohen and Chen (2023)). We investigate if such a nondiagonal metric can be beneficial within PDE-G-CNNs. Figure 6.1 shows the difference between a diagonal and nondiagonal metric by looking at contours of the Riemannian distance $d(p_0, \cdot)$ from the reference position-orientation $p_0 = (0, 0, 0)$.



$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1.25 & 0 & 0.75 \\ 0 & 16 & 0 \\ 0.75 & 0 & 1.25 \end{pmatrix}$$

Figure 6.1: Contours $d(p_0, \cdot) = 0.5, 1.0, 1.5, 2.0, 2.5$ for a diagonal metric (6.1) and nondiagonal metric (6.2). The domain of the plot is $(x, y, \theta) \in [-3, 3]^2 \times [-\pi, \pi) \subset \mathbb{M}_2$, with the vertical axis being θ. At the bottom, we have identically valued isocontours of the min-pooling projection over θ.

6.1.1 Contributions

In Section 6.3 we show theoretically that a fixed lifting layer (6.11) is as expressive as a trained lifting layer (6.8).

Table 6.1 shows that fixing the lifting layer has no significant impact on the accuracy of PDE-G-CNNs. However, fixing the lifting layer does enrich the geometric interpretation of PDE-G-CNNs.

Table 6.2 shows that using nondiagonal metrics has no significant impact on the accuracy of PDE-G-CNNs. However, we do see an improvement in training stability by using nondiagonal metrics.

6.1.2 OUTLINE

- In Section 6.2 we cover convection over invariant vector fields on $\mathbb{M}_2$.
- In Section 6.3 we investigate if fixing the lifting layer can improve the accuracy of PDE-G-CNNs.
- In Section 6.4 we investigate if nondiagonal metrics can improve the accuracy of PDE-G-CNNs.
- In Section 6.5 we conclude this chapter.

6.2 INVARIANT CONVECTION AND ORIENTATION SCORE TRANSFORM

The aim is to achieve the first geometric adaptation of PDE-G-CNNs: fixing the lifting layer. For this objective we need to study the invariant convection PDE on $\mathbb{M}_2$, how it is solved, and how it interacts with the orientation score transform $\mathcal{W}_\psi$ (2.41).

Definition 6.1 (Invariant Convection): Let $c \in \Gamma(T\mathbb{M}_2)$ be a $\mathrm{SE}(2)$ invariant vector field on $\mathbb{M}_2$. The invariant convection PDE on $\mathbb{M}_2$ is defined as:

$$\frac{\partial f}{\partial t} = -cf, \quad f(\cdot, 0) = f_0, \tag{6.3}$$

where $f : \mathbb{M}_2 \times \mathbb{R} \to \mathbb{R}$ is a smooth scalar field evolving over time t, and f_0 some initial condition.

We can succinctly state the solution of a convection PDE using the flow Φ^t of the vector field c as the following proposition shows.

Proposition 6.1 (Solution of Convection): Let $\Phi^t : \mathbb{M}_2 \to \mathbb{M}_2$ be the flow of a vector field c. The solution to convection (6.3), in terms of the initial condition f_0, is:

$$f(p, t) = f_0(\Phi^{-t}(p)) \tag{6.4}$$

Proof: We will show that any solution of the convection PDE satisfies (6.4). Assume there exists some solution $f(p,t)$. Consider the following:

$$\frac{\partial}{\partial t}[f(\Phi^t(p),t)] = (cf)(\Phi^t(p),t) + \frac{\partial f}{\partial t}(\Phi^t(p),t) = 0, \quad (6.5)$$

where the first equality is due to the chain rule together with the fact that $\frac{\mathrm{d}}{\mathrm{d}t}\Phi^t(p) = c_{\Phi^t(p)}$, and the last by the assumption that f is a solution of (6.3). Given that this derivative is 0, we conclude that $f(\Phi^t(p),t) = f(\Phi^0(p),0) = f_0(p)$, which we can rewrite as $f(p,t) = f_0(\Phi^{-t}(p))$. □

The following proposition shows that performing an orientation score transform with a roto-translated kernel is equivalent to doing a specific convection after transforming with the original kernel. For ease of notation, we define $\mathcal{C}_c^t$ as the operator that takes in an initial condition f_0 and runs the convection PDE (6.3) for t time; that is, it returns $f(\cdot,t)$.

Proposition 6.2: Let $h \in \mathrm{SE}(2)$ and $c \in \Gamma(T\mathbb{M}_2)$ be an invariant vector field such that $hp_0 = \Phi^{-1}(p_0)$, where Φ^t is the flow of c. We have:

$$\mathcal{C}_c^1 \circ \mathcal{W}_\psi = \mathcal{W}_{h\psi}. \quad (6.6)$$

Proof:

$$\begin{aligned}(\mathcal{C}_c^1\mathcal{W}_\psi f)(gp_0) &= (\mathcal{W}_\psi f)(\Phi^{-1}(gp_0)) = (\mathcal{W}_\psi f)(g\Phi^{-1}(p_0)) \\ &= (\mathcal{W}_\psi f)(ghp_0) = \langle gh\psi, f\rangle \\ &= \langle g(h\psi), f\rangle = (\mathcal{W}_{h\psi} f)(gp_0)\end{aligned} \quad (6.7)$$

The first equality is due to Proposition 6.1. The second equality follows from the fact that the flow over an invariant vector field is equivariant, that is $\Phi^t(gp) = g\Phi^t(p)$. □

6.3 Fixing the Lifting Layer

A trained lifting layer L_{train} is defined as

$$(L_{\text{train}} f)_i = \mathcal{W}_{\psi_i} f, \quad i = 1, 2, ..., C, \tag{6.8}$$

where $\mathcal{W}_{\psi_i}$ are orientation score transforms (Section 2.3), and $\psi_i : \mathbb{R}_2 \to \mathbb{R}$ learned kernels. That is, the trained lifting layer turns an input image $f : \mathbb{R}^2 \to \mathbb{R}$ into C orientation scores $\mathcal{W}_{\psi_i} f : \mathbb{M}_2 \to \mathbb{R}$. Figure 6.2 shows a diagram of this layer.

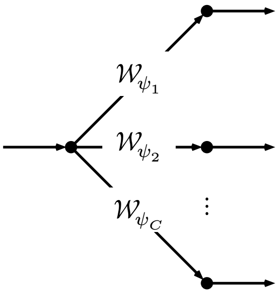


Figure 6.2: A trained lifting layer L_{train} (6.8). We perform C lifts $\mathcal{W}_{\psi_i}$ with learned kernels ψ_i.

In practice, the kernels ψ_i are implemented as 2D arrays of a certain size. This means we can create a basis for this finite-dimensional space of $k \times k$ kernels using roto-translated versions of some "mother" wavelet $\hat{\psi}$ and write:

$$\psi_i = \sum_{j=1}^{k^2} A_{ij} g_j \hat{\psi}, \tag{6.9}$$

with $A \in \mathbb{R}^{C \times k^2}$ and $g_j \in \text{SE}(2)$.

With this information we can show that the lifts $\mathcal{W}_{\psi_i}$ amount to:

$$\mathcal{W}_{\psi_i} = \mathcal{W}_{\sum_{j=1}^{k^2} A_{ij} g_j \hat{\psi}} = \sum_{j=1}^{k^2} A_{ij} \mathcal{W}_{g_j \hat{\psi}} = \sum_{j=1}^{k^2} A_{ij} \mathcal{C}^1_{c_j} \mathcal{W}_{\hat{\psi}}, \tag{6.10}$$

where the vector fields c_j are such that $g_j p_0 = \Phi^{-1}_{c_j}(p_0)$. The second equality is due to linearity of the lift with respect to the wavelet, and the third due to Proposition 6.2.

This motivates the creation of a *fixed lifting layer*:

$$\left(L_{\text{fix}} f\right)_i = \sum_{j=1}^{K} A_{i,j} \mathcal{C}^1_{c_j} \mathcal{W}_{\hat{\psi}} f, \quad i = 1, 2, \ldots, C, \tag{6.11}$$

where we perform K convections with learned invariant vector fields $c_1, \ldots, c_K$, after a single lift with the mother wavelet $\hat{\psi}$. The number of convections K becomes a new network hyperparameter. A diagram of the fixed lifting layer can be seen in Figure 6.3.

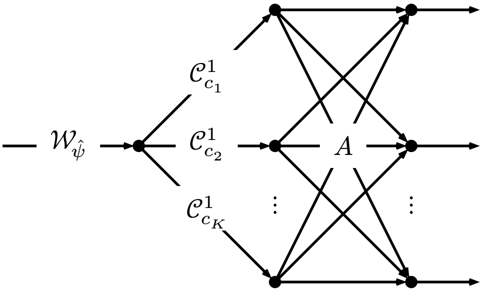


Figure 6.3: A fixed lifting layer L_{fix} (6.11). We first perform a single fixed lift $\mathcal{W}_{\hat{\psi}}$ with "mother" wavelet $\hat{\psi}$, then K learned convections $\mathcal{C}^1_{c_i}$, and finally apply a learned linear combination $A \in \mathbb{R}^{C \times K}$.

We perform the lift $\mathcal{W}_{\hat{\psi}}$ in the fixed lifting layer L_{fix} using cake wavelets (Section 2.3). This is a well-motivated decision: 1) the PDE-based neural networks already operate on a discretized set of orientations, 2) lifting using cake wavelets is (essentially) an invertible operation, and 3) lifting using cake wavelets has the *fast reconstruction* (Remark 2.4) property, meaning that sum-pooling (2.42) over the orientations (essentially) undoes the lift.

It can also be argued that the fixed lifting layer L_{fix} is more general than the trained lifting layer L_{train} in the following sense. We can reinterpret a trained lifting layer as a fixed lifting layer but with the convections constrained to pixel-step translations. This is in contrast to the fixed lifting layer in which the convections are allowed to take arbitrarily large sub-pixel roto-translations.

As we have shown above, the fixed and trained lifting layers are theoretically equivalent. However, in general, functionally equivalent architectures do not necessarily yield comparable accuracy in practice (e.g. in view of sampling constraints). We therefore experimentally investigate these theoretical claims on the DCA1 dataset (Subsection 2.8.2). The code for this experiment is available in the GitLab repository LieTorch.

We consider 4 networks, all with 6 layers: a standard CNN, a G-CNN with a trained lifting layer, a PDE-G-CNN with a trained lifting layer, and a PDE-G-CNN with a fixed lifting layer. Every network was trained 10 times for 80 epochs. During training, the average Dice coefficient on the test set is monitored, and the maximum value is stored. The result is 10 Dice coefficients for every architecture.

The results can be found in Table 6.1. We see that the PDE-G-CNN with a fixed lifting layer has identical accuracy to the one with a trained lifting layer, while having an effectively equal number of parameters.

Type	Lifting	C	Params.	Dice Coeff.
CNN	-	-	25662	$0.756_{\pm 0.0051}$
G-CNN	Trained	8	24632	$0.751_{\pm 0.0020}$
PDE-G-CNN	Trained	16	2560	$0.775_{\pm 0.0042}$
PDE-G-CNN	Fixed	16	2536	$0.775_{\pm 0.0052}$

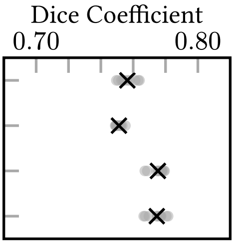


Table 6.1: The architecture of different 6-layer networks, and their accuracy when trained on the DCA1 dataset. The fixed lifting PDE-G-CNN was done with $K = 40$ convections (6.11).

6.4 Nondiagonal Metric

Next we address the second geometric adaptation of PDE-G-CNNs. We want to investigate whether the generalization to nondiagonal metrics can change the accuracy of the PDE-G-CNNs. To this end, we perform some experiments on the Lines dataset (Subsection 2.8.3). The code for this experiment is available in the GitLab repository LieTorch.

The nondiagonal metrics are of the form $\mathcal{G}(\mathcal{A}_i, \mathcal{A}_j) = G_{ij}$, with $G \in \mathbb{R}^{3\times 3}$ a symmetric positive definite matrix. To avoid enforcing the SPD constraint explicitly, we parametrize G as $G = H^\top H$ with $H \in \mathbb{R}^{3\times 3}$, which is symmetric positive *semi*-definite by construction.

We consider two different 6-layer PDE-G-CNNs: one with diagonal metrics and one with nondiagonal metrics. We increased the kernel size to $[9, 9, 9]$. Every network was trained 15 times for 60 epochs.

The results can be found in Table 6.2. We see that using nondiagonal metrics does not significantly affect the average accuracy of the PDE-G-CNNs on the Lines dataset. However, it does lead to a significant reduction in variance across training runs, indicating improved training stability.

Metric	Params.	Dice Coeff.
Diag.	6018	$0.9758_{\pm 0.0035}$
Nondiag.	7458	$0.9764_{\pm 0.0009}$

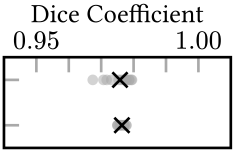


Table 6.2: The accuracy of two different 6-layer PDE-G-CNNs when trained on the Lines dataset.

6.5 Conclusion

In this chapter, we have investigated two geometric adaptations of PDE-G-CNNs.

1. We showed both theoretically and in practice (Table 6.1) that one does not need to lift with learned kernels in PDE-G-CNNs. A fixed lifting layer using cake wavelets gives identical accuracy on the DCA1 dataset, while keeping the network complexity effectively the same. Our experiments confirm that fixing the lifting layer preserves the accuracy while considerably increasing the geometric interpretation of PDE-G-CNNs.
2. In Table 6.2 we showed on the Lines dataset that using a nondiagonal metric with respect to the SE(2) invariant frame $\mathcal{A}_i$ (6.2) for the erosion and dilation, does not significantly increase the accuracy on the Lines dataset. However, we do see an improvement in training stability by using nondiagonal metrics.

SEMIFIELD SCALE-SPACES IN PDE-BASED NEURAL NETWORKS

7

7.1 INTRODUCTION

In this chapter we will be deriving in an axiomatic way which PDEs should be used in PDE-based neural networks. This framework includes the PDEs that are currently already used, but also reveals previously unused PDEs, meaning that the accuracy of PDE-G-CNNs could possibly be improved by adding them. Our approach is inspired by the axioms of scale-space theory, which we generalize by introducing *semifield-valued* signals, and motivated from a machine learning perspective.

The architecture of PDE-based neural networks is very general in the sense that the feature maps $f : M \to \mathbb{R}$ can be defined on an arbitrary homogeneous space M on which a Lie group G acts. We will *not* consider the general setting in this chapter, and restrict ourselves to $M = \mathbb{R}^2$ and $G = \mathrm{E}(2)$, i.e. standard two-dimensional Euclidean space with its symmetries, for simplicity. We call this specific instance a *PDE-CNN*.

We experimentally examine what impact the newly found PDEs have on the accuracy of PDE-CNNs. We also confirm for small networks that PDE-CNNs offer fewer parameters, increased accuracy, and better data efficiency when compared to CNNs.

7.1.1 CONTRIBUTIONS

We list six axioms (Definition 7.27) that a PDE used in a PDE-based neural network should satisfy. The axioms are closely related to classical scale-space theory, but are more general in the sense that we permit semifield-valued signals. The goal of this generalization is to allow for the discovery (or invention) of new PDEs that can be used in the design of PDE-based neural networks.

We will only consider semifields that are commutative and one-dimensional, and the domain of the PDEs will be the two-dimensional Euclidean space $\mathbb{R}^2$. To maintain a practical perspective, we will consistently connect the overarching theory using five example semifields: the linear, root, logarithmic, tropical min, and tropical max semifields.

From the axioms, we demonstrate in Theorem 7.1 that every semifield corresponds to a unique family of scale-spaces, this being the main theoretical contribution. This shows that PDE-based neural networks, in their current form, can be extended greatly by adding new PDEs that generate currently unused scale-spaces.

We experimentally assess how effective the incorporation of new semifields and their corresponding PDEs is in Subsection 7.7.1.

In Subsection 7.7.2 we verify that PDE-CNNs exhibit superior data efficiency, reduced parameter count, and competitive accuracy compared to traditional CNNs.

7.1.2 OUTLINE

- In Section 7.2 we provide background on scale-spaces, semifields, and a non-exhaustive list of related literature.
- In Section 7.3 we define semifields and all related structures and operations.
- In Section 7.4 we state the semifield scale-space axioms.
- In Section 7.5 we show that once a semifield is chosen a unique (one-parameter) family of scale-spaces arise (Theorem 7.1).
- In Section 7.6 we briefly note on the architectural design of PDE-CNNs.
- In Section 7.7 we lay out two experiments and discuss their results.
- In Section 7.8 we conclude this chapter.

7.2 BACKGROUND

7.2.1 SEMIFIELDS & QUASILINEARITY

Every scale-space representation has a natural corresponding algebra called a *semifield*. In this section we will discuss which semifields correspond to the diffusion, dilation, and erosion scale-spaces, as described in Section 2.4.

Consider the Gaussian scale-space representation f_t (2.50) of an image $f_0 : \mathbb{R}^2 \to \mathbb{R}$:

$$f_t(x) = \int_{\mathbb{R}^2} \frac{1}{2\pi t} \exp\left(-\frac{\|x-y\|^2}{2t}\right) f_0(y)\,\mathrm{d}y. \tag{7.1}$$

This scale-space representation is *linear* in the sense that if one takes two images $f_0, g_0 : \mathbb{R}^2 \to \mathbb{R}$ and two scalars $a, b \in \mathbb{R}$, then the scale-space h_t of the image $h_0 = af_0 + bg_0$ is equal to $h_t = af_t + bg_t$. But, in an analogous manner, the dilation scale-space (2.52) is *quasilinear* in the sense that the scale-space of the image $h_0 = \max\{a + f_0, b + g_0\}$, where we interpret the maximum pointwise, is equal to $h_t = \max\{a + f_t, b + g_t\}$. In the same way, the erosion scale-space (2.54) is quasilinear in the min sense. To define what we mean with quasilinear more precisely we need to introduce semifields.

A *semifield* $(R, \mathbb{0}, \mathbb{1}, \oplus, \otimes)$ is an algebraic structure like a field but where we relax the requirement that the addition $\oplus$ has inverses. The prototypical example of a semifield are the nonnegative real numbers $L_{\geq 0} = (\mathbb{R}_{\geq 0}, 0, 1, +, \times)$ with standard addition and multiplication. We have already seen two other examples of semifields in the dilation and erosion scale-spaces. Namely, the so-called *tropical max* semifield defined as $T_+ = (\mathbb{R} \cup \{-\infty\}, -\infty, 0, \max, +)$ and the *tropical min* semifield $T_- = (\mathbb{R} \cup \{\infty\}, \infty, 0, \min, +)$. In the tropical semifields the minimum (or maximum) of two numbers becomes semifield addition, and normal addition becomes semifield multiplication.

With the definition of a semifield we can state the quasilinearity of a scale-space formally as *semifield R-linearity*. So, like before, consider a semifield R and two semifield-valued images $f_0, g_0 : \mathbb{R}^2 \to R$ and two elements $a, b \in R$. Then by a scale-space being R-linear we mean that the scale-space of the R-linear combination of images $h_0 = (a \otimes f_0) \oplus (b \otimes g_0)$ is equal to the R-linear combination of scale-spaces $h_t = (a \otimes f_t) \oplus (b \otimes g_t)$. In other words, the operation that takes an image and returns its scale-space representation is a semifield linear operator. For example, the Gaussian scale-space is $L_{\geq 0}$-linear, the quadratic dilation scale-space is T_+-linear, and the quadratic erosion scale-space is T_--linear.

In Pauwels, van Gool, Fiddelaers and Moons (1995) and Duits, Florack, de Graaf and ter Haar Romenij (2004) it is argued in an axiomatic way that the only linear scale-space representations correspond to solutions of the α-diffusion (pseudo-)PDE $\frac{\partial f}{\partial t} = -(-\Delta)^{\alpha/2} f$. In a completely analogous manner, one can show (Welk (2003), Schmidt and Weickert (2016)) that the only morphological scale-spaces, that being scale-spaces that are T_+ or T_- linear, correspond to (viscosity) solutions of the α-dilation PDE $\frac{\partial f}{\partial t} = +\frac{1}{\alpha} \lVert \nabla f \rVert^{\alpha} f$ and α-erosion PDE $\frac{\partial f}{\partial t} = -\frac{1}{\alpha} \lVert \nabla f \rVert^{\alpha} f$.

These facts reveal something important: to discover new PDEs that can be used in PDE-based neural network we *need* to generalize scale-space theory to semifields other than just $L_{\geq 0}$, T_+, and T_-.

7.2.2 Related Work

In this subsection we provide a non-exhaustive list of related scale-space literature.

Linear Scale-Spaces. In Iijima (1959) the first (Weickert, Ishikawa and Imiya (1999)) axiomatic treatment of linear scale-space theory is presented. Axioms such as linearity, roto-translation equivariance, one-parameter semigroup property, and most notably, the scale equivariance, can all be found in Iijima's article, axioms we will also be using. Iijima shows that the Gaussian scale-space arises from his axioms, and that the Laplacian generates it. Witkin (1983) and Koenderink (1984) brought the Gaussian scale-space representation to wider attention. In Pauwels, van Gool, Fiddelaers and Moons (1995) and Duits, Florack, de Graaf and ter Haar Romenij

(2004) an extended class of linear scale-spaces is explored. They derive that (fractional) powers of the Laplacian are valid linear scale-space generators. Analysis of the scale-space axioms in the Fourier domain is extensively used, an approach we apply in the broader semifield setting.

Morphological Scale-Spaces. In Brockett and Maragos (1992) it was shown for the first time (Heijmans and van den Boomgaard (2002)) that morphological operators like dilations and erosions in image processing can be described in terms of PDEs. In Dorst and van den Boomgaard (1994) the slope transform is shown to be the morphological counterpart of the Fourier transform, and related to the Legendre-Fenchel transform. The semifield Fourier transform we introduce reduces to the Legendre-Fenchel transform in the tropical semifield cases. In Burgeth and Weickert (2005) and Schmidt and Weickert (2016) a connection between linear and morphological scale-spaces is described using the Cramér transform. The Cramér transform gives us a way to translate between the kernels of the linear and morphological scale-spaces. We will show that the kernels of all semifield scale-spaces have the same form in the Fourier domain (this being our main theorem), illuminating further the connection between the linear and morphological world.

Other Scale-Space Theory. In Alvarez, Guichard, Lions and Morel (1993) an axiomatic approach to PDE-based scale-spaces is described. The strength of this approach is that it also includes mean curvatures flows Sapiro (2001) as highly powerful nonlinear PDEs. Solutions of such nonlinear PDEs may be solved with median filtering (Guichard and Morel (1997)), however, they lack a semifield structure (taking mean/median are not associative binary operations), thus falling outside the scope of the theory presented here. In Florack (2001) nonlinear scale-spaces are obtained by performing a monotonic transformation (known as a *Cole-Hopf transform* Evans (2010), Ch.4.4) on the gray-values of a standard linear scale-space and deducing what nonlinear PDE corresponds to the obtained evolution. This transformation neatly bridges linear, logarithmic, and in the extreme cases, morphological scale-spaces, and we will also use this link. In

Heijmans and van den Boomgaard (2002) an algebraic framework for scale-spaces is given. Importantly, their perspective is (initially) totally divorced from PDEs, convolutions, and kernels, and focuses solely on the evolution operator. We will define our semifield scale-spaces in the same manner.

Scale-Spaces in Machine Learning. In Jacobsen, van Gemert, Lou and Smeulders (2016), Pintea, Tomen, Goes, Loog and van Gemert (2021), Tomen, Pintea and van Gemert (2021), and Saldanha, Pintea, van Gemert and Tomen (2021) the Gaussian scale-space and its spatial (fractional) derivatives are employed to design architectures that can learn filters at the appropriate scale by optimizing the scale parameter(s) during training. Architectures presented in Luan, Chen, Zhang, Han and Liu (2018) and Romero, Bruintjes, Tomczak, Bekkers, Hoogendoorn and van Gemert (2022) also learn scale parameters. PDE-G-CNNs learn Riemannian metric tensor fields, which, due to the scale-equivariance of scale-space representations, is equivalent to learning scale-parameters. In this sense PDE-based neural networks are closely related to these “scale learning” architectures. In Worrall and Welling (2019), Sosnovik, Szmaja and Smeulders (2020), Lindeberg (2022), and Sangalli, Blusseau, Velasco-Forero and Angulo (2021) (discrete) Gaussian and morphological scale-space representations are used to create architectures that are scale equivariant.

7.3 Semifield Theory

In this section we define semifields (Definition 7.1) and all mathematical structures and operations made from them. This includes important concepts such as semimodules (Definition 7.7), linearity (Definition 7.8), measures (Definition 7.16), integration (Definition 7.18), convolution (Definition 7.20), and Fourier transforms (Definition 7.25).

7.3.1 SEMIFIELD, SEMIMODULES & LINEARITY

Definition 7.1 (Semifield): A (commutative) *semifield* R is a tuple $R = (R, \mathbb{0}, \mathbb{1}, \oplus, \otimes)$ where $\oplus, \otimes : R \times R \to R$ are two commutative and associative binary operations on R called semifield addition and multiplication, such that for all $a, b, c \in R$:

$$a \oplus \mathbb{0} = a,$$
$$a \otimes \mathbb{1} = a,$$
$$a \neq \mathbb{0} : \exists a^{-1} : a \otimes a^{-1} = \mathbb{1},$$
$$a \otimes \mathbb{0} = \mathbb{0},$$
$$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c).$$

In other words, a semifield is a field where we do not require to have "negative elements", that being additive inverses.

Throughout we will denote an arbitrary semifield-related operation with a circled version of the most closely related linear counterpart. Some example symbols are $\oplus$, $\otimes$, $\oint$, and $\circledast$, which respectively correspond to semifield addition, multiplication, integration, and convolution.

We will mainly consider the following semifields:

Definition 7.2 (Semifields of Interest):

a) The linear semifield $L = (\mathbb{R}, 0, 1, +, \times)$ with the usual addition $+$ and multiplication $\times$. We can restrict the set to $\mathbb{R}_{\geq 0}$ and we write $L_{\geq 0}$ in that case.

b) The root semifields $R_p = \left(\mathbb{R}_{\geq 0}, 0, 1, \oplus_p, \times\right)$ with $p \neq 0$ where semifield addition is $a \oplus_p b := \sqrt[p]{a^p + b^p}$, and where semifield multiplication is normal multiplication.

c) The logarithmic semifields $L_\mu = \left(\mathbb{R} \cup \{\pm\infty\}, \pm\infty, 0, \oplus_\mu, +\right)$ with $\mu \neq 0$ where semifield addition is $a \oplus_\mu b := \frac{1}{\mu} \ln\left(e^{\mu a} + e^{\mu b}\right)$, and where semifield multiplication is normal addition. If $\mu > 0$ we add $-\infty$ to the ring to act as the additive identity, and if $\mu < 0$ we add $+\infty$.

d) The tropical max semifield $T_+ = (\mathbb{R} \cup \{-\infty\}, -\infty, 0, \max, +)$, where max is semifield addition, and usual addition is semifield multiplication.
e) The tropical min semifield $T_- = (\mathbb{R} \cup \{\infty\}, \infty, 0, \min, +)$, where min is semifield addition, and usual addition is semifield multiplication.

The family of logarithmic semifields is interesting as in the limits one has:

$$\begin{aligned} \lim_{\mu \to +\infty} a \oplus_\mu b &= \max(a, b), \\ \lim_{\mu \to -\infty} a \oplus_\mu b &= \min(a, b). \end{aligned} \tag{7.2}$$

Thereby, the family of logarithmic semifields L_μ relate to the tropical semifields $T_\pm$ in the extreme cases of μ.

Definition 7.3 (Semifield Isomorphism): Let $R = (R, \mathbb{0}, \mathbb{1}, \oplus, \otimes)$ and $\tilde{R} = \left(\tilde{R}, \tilde{\mathbb{0}}, \tilde{\mathbb{1}}, \tilde{\oplus}, \tilde{\otimes}\right)$ be two semifields. A *semifield isomorphism* $\varphi : R \to \tilde{R}$ is a bijective map that satisfies for all $a, b \in R$:

$$\begin{aligned} &\varphi(\mathbb{0}) = \tilde{\mathbb{0}}, \\ &\varphi(\mathbb{1}) = \tilde{\mathbb{1}}, \\ &\varphi(a \oplus b) = \varphi(a) \mathbin{\tilde{\oplus}} \varphi(b), \\ &\varphi(a \otimes b) = \varphi(a) \mathbin{\tilde{\otimes}} \varphi(b). \end{aligned}$$

If there exists a semifield isomorphism between two semifields they are called isomorphic.

Proposition 7.1 (Some Semifield Isomorphisms):

- The root semifields R_p are isomorphic to the nonnegative linear semifield $L_{\geq 0}$, with the isomorphism $\varphi_p : R_p \to L_{\geq 0}$ being $\varphi_p(x) = x^p$.

- The logarithmic semifields L_μ are isomorphic to the nonnegative linear semifield $L_{\geq 0}$, with the isomorphism $\varphi_\mu : L_\mu \to L_{\geq 0}$ being $\varphi_\mu(x) = e^{\mu x}$.
- The tropical max semifield T_+ is isomorphic to the tropical min semifield T_-, with the isomorphism $\varphi : T_+ \to T_-$ being $\varphi(x) = -x$.
- Informally, in the limit $\mu \to \pm\infty$ the logarithmic semifields L_μ "converge" to the tropical semifields $T_\pm$.

The above proposition shows that although we defined five semifields of interest, as listed in Definition 7.2, we are, in fact, only working with two non-isomorphic ones.

For the purpose of analysis we endow the semifields with a metric. Normally, a linear structure X is endowed with a *norm* $\|\cdot\| : X \to \mathbb{R}_{\geq 0}$ and afterwards a metric d is defined through $d(a, b) = \|a - b\|$. This is not possible in our semifield setting as we do not necessarily have additive inverses (consider for example the tropical semifields).

Definition 7.4 (Semifield Metric): Let R be a semifield. A *semifield metric* $\rho : R \times R \to \mathbb{R}_{\geq 0}$ is a metric such that for all $a, b, c \in R$ we have:

$$\rho(c \oplus a, c \oplus b) \leq \rho(a, b),$$
$$\rho(c \otimes a, c \otimes b) = \rho(c, \mathbb{0})\rho(a, b).$$

These properties are direct generalizations of the common notions of translation invariance and absolute homogeneity. More importantly, they ensure that semifield addition $\oplus$ and multiplication $\otimes$ are continuous (w.r.t the metric).

Definition 7.5 (Employed Semifield Metrics):

a) In the linear semifield L case we use the metric $\rho_L(a, b) = |a - b|$.

b) In the root semifields R_p case we use the metric $\rho_{R_p}(a,b) = |a^p - b^p|$.
c) In the logarithmic semifields L_μ case we use the metric $\rho_{L_\mu}(a,b) = |e^{\mu a} - e^{\mu b}|$.
d) In the tropical max semifield T_+ case we use the metric $\rho_{T_+}(a,b) = |e^a - e^b|$.
e) In the tropical min semifield T_- case we use the metric $\rho_{T_-}(a,b) = |e^{-a} - e^{-b}|$.

The root and logarithmic semifield metrics are natural as they borrow the metric on the linear semifield L through the isomorphisms $\varphi_p(x) = x^p$ and $\varphi_\mu(x) = e^{\mu x}$, see Proposition 7.1. Similarly, the tropical min and max semifield metrics relate by their isomorphism $\varphi(x) = -x$.

Definition 7.6 (One-Dimensional Semifield): Let R be a metric semifield. If R as a topological space (with the topology induced by the metric) is locally homeomorphic to one-dimensional Euclidean space we say it is one-dimensional.

Just as mathematical rings and fields can be used to create modules and vector spaces, we define an analogous structure called a semimodule using semifields.

Definition 7.7 (Semimodule): Let $R = (R, \mathbb{0}, \mathbb{1}, \oplus, \otimes)$ be a semifield. An *R-semimodule* $V = (V, \oplus_V, \otimes_V, \mathbb{0}_V)$ over R is a set with a commutative and associative binary operation $\oplus_V : V \times V \to V$ called addition, and another binary operation $\otimes_V : R \times V \to V$ called (left) scalar multiplication, such that for all $a, b \in R$ and $u, v \in V$:

$$v \oplus_V \mathbb{0}_V = v,$$
$$\mathbb{1} \otimes_V v = v,$$
$$\mathbb{0} \otimes_V v = \mathbb{0}_V,$$

$$(a \oplus b) \otimes_V v = (a \otimes_V v) \oplus_V (b \otimes_V v),$$
$$a \otimes_V (u \oplus_V v) = (a \otimes_V u) \oplus_V (a \otimes_V v).$$

We do not write the subscript V on the operations of a semimodule V from here on out, as is usual.

Now that we have semimodules we can speak of semifield-linearity in its full generality. The notion of semifield-linearity is totally analogous to the normal notion of linearity, therefore the name.

Definition 7.8 (Semifield Linear): Let V_1, V_2 be two semimodules over the same semifield R. A map $\varphi : V_1 \to V_2$ is called R-linear if for all $a, b \in R$ and $u, v \in V_1$ we have:

$$\varphi(a \otimes u \oplus b \otimes v) = a \otimes \varphi(u) \oplus b \otimes \varphi(v). \tag{7.3}$$

7.3.2 FUNCTIONS, MEASURABILITY & INTEGRATION

The prototypical semimodule over a semifield is the space of all semifield-valued functions on a set.

Definition 7.9 (Function Semimodule): Let R be a semifield. Consider the set $F(\mathbb{R}^2, R)$ of all R-valued functions $f : \mathbb{R}^2 \to R$. The set $F(\mathbb{R}^2, R)$ forms an R-semimodule under pointwise semifield addition and multiplication. The semimodule $F(\mathbb{R}^2, R)$ is called the function semimodule over $\mathbb{R}^2$. More generally, any subsemimodule of $F(\mathbb{R}^2, R)$ is also called a function semimodule over $\mathbb{R}^2$.

On the function semimodule $F(\mathbb{R}^2, R)$ we define the following natural R-linear domain transformation operators:

Definition 7.10 (Operators on Function Semimodule):

- **Translation Operator:** For all translation vectors $v \in \mathbb{R}^2$ we define the translation operator $\mathcal{T}_v$
$$(\mathcal{T}_v f)(x) := f(x - v). \tag{7.4}$$
- **Rotoreflection Operator:** For all orthonormal matrices $Q \in \mathbb{R}^{2\times 2}$ we define the rotoreflection operator $\mathcal{R}_Q$
$$\big(\mathcal{R}_Q f\big)(x) := f(Q^{-1}x). \tag{7.5}$$
- **Scaling Operator:** For all scalings $s \in \mathbb{R}_{>0}$ we define the scaling operator $\mathcal{S}_s$
$$(\mathcal{S}_s f)(x) := f\left(\frac{x}{s}\right). \tag{7.6}$$
- **Pointwise Operator:** For all φ we define the pointwise operator $\mathcal{P}_\varphi$
$$\Big(\mathcal{P}_\varphi(f)\Big)(x) := \varphi(f(x)). \tag{7.7}$$

To avoid pathological cases, we use standard measure theoretical concepts which we introduce next.

Definition 7.11 (Measurable Space & Set): Let (X, d) be a complete metric space. We equip the space X with the natural Borel sigma algebra B induced by the metric d. This turns X into a *measurable space.* A *measurable set* is any element of the Borel sigma algebra B.

With the above definition we can turn both $\mathbb{R}^2$ and any metric semifield R into a measurable space.

Definition 7.12 (Measurable Function): Let R be a metric semifield and $f : \mathbb{R}^2 \to R$ a function. The function f is called *measurable* if the pre-image of any measurable set is a measurable set.

The set of measurable functions is broad enough to be well-behaved under pointwise limits, as the following lemma describes.

Lemma 7.1: Let R be a metric semifield, and let $f(x) = \lim_{n\to\infty} f_n(x)$ be the pointwise limit of measurable functions $f_n : \mathbb{R}^2 \to R$. The limit here is with respect to the semifield metric ρ (Definition 7.4), that is, $\lim_{n\to\infty} \rho(f_n(x), f(x)) = 0$. Then f is also measurable.

A proof of a generalization of this lemma can be found in SAZ (2014). But there exists an even stronger statement that describes measurable functions as pointwise limits of indicator functions simple functions, which are made from indicator functions.

Definition 7.13 (Indicator Function): Let R be a semifield and $A \subseteq \mathbb{R}^2$ any set. We define the *indicator function* $\mathbb{1}_A$ of A as:

$$\mathbb{1}_A(x) = \begin{cases} \mathbb{1} & \text{if} \quad x \in A \\ \mathbb{0} & \text{otherwise} \end{cases} \tag{7.8}$$

Definition 7.14 (Simple Function): Let R be a semifield. A *simple function* $s : \mathbb{R}^2 \to R$ is a finite R-linear combination of indicator functions of measurable sets A_i.

$$s = \bigoplus_{i=1}^{n} a_i \otimes \mathbb{1}_{A_i}, \tag{7.9}$$

where each $a_i \in R$.

The link between measurable and simple functions is a follows.

Lemma 7.2: Let R be a metric semifield and consider semifield-valued functions on $\mathbb{R}^2$. The pointwise limit of a sequence of simple functions is measurable. Every measurable function is the pointwise limit of a sequence of simple functions.

Proof: The pointwise limit of a sequence of simple functions being measurable follows immediately from Lemma 7.1, as simple functions are measurable. Showing that every measurable function is the pointwise limit of simple functions goes via a straightforward construction. □

For every semifield there is a natural associated class of functions. We would like to specify this class in an axiomatic sense. This is where the sum-approachable definition comes into play. It is a restriction of the well-known statement that "every measurable function is the limit of simple functions".

Definition 7.15 (Sum-Approachable): Let R be a metric semifield. A function $f : \mathbb{R}^2 \to R$ is *sum-approachable* if there exists $a_i \in R$ and $A_i \subseteq \mathbb{R}^2$ open such that we have

$$f(x) = \lim_{n \to \infty} \bigoplus_{i=1}^{n} a_i \otimes \mathbb{1}_{A_i}(x). \tag{7.10}$$

The semimodule of all sum-approachable functions $f : \mathbb{R}^2 \to R$ is denoted by $S(\mathbb{R}^2, R)$.

There are two differences between sum-approachable and measurable: we only consider open sets, not measurable sets, and we have a limit of a semifield sum of indicator functions, not just a limit.

A function being sum-approachable is more restrictive than one might think at first sight. The following lemma illustrates this by showing that in the tropical cases the sum-approachable functions enjoy the property of being semicontinuous, something that does *not* happen in the linear case.

Lemma 7.3: A sum-approachable function $f : \mathbb{R}^2 \to T_+$ is lower semicontinuous. A sum-approachable function $f : \mathbb{R}^2 \to T_-$ is upper semicontinuous.

Proof: Consider the tropical max semifield case for the moment. Every indicator function $\mathbb{1}_A(x)$ with $A \subseteq \mathbb{R}^2$ open is lower semicontinuous in this case. The limit-semifield-sum in the definition of sum-approachable turns into a pointwise supremum in this case.

The pointwise supremum of lower semicontinuous functions is again lower semicontinuous. Let $f(x) = \sup_n f_n(x)$. Let $\varepsilon > 0$ and $x_0 \in \mathbb{R}^2$. Choose N such that $f_N(x_0) > f(x_0) - \frac{\varepsilon}{2}$. Choose $\delta > 0$ such that $f_N(x) > f_N(x_0) - \frac{\varepsilon}{2}$ when $|x - x_0| < \delta$. Then $f(x) > f_N(x) > f_N(x_0) - \frac{\varepsilon}{2} > f(x_0) - \varepsilon$ (Matematleta (2018)).

Thus, every sum-approachable function $f : \mathbb{R}^2 \to T_+$ is lower semicontinuous. *Mutatis mutandis*, the exact same argument holds in the tropical min semifield case. ☐

Definition 7.16 (Semifield Measure): Let Σ be the Borel sigma algebra on $\mathbb{R}^2$ and $R = (R, \mathbb{0}, \mathbb{1}, \oplus, \otimes)$ a semifield. A *semifield measure* $\mu : \Sigma \to R$ is a map that satisfies the following properties.

- **Nullity of the Empty Set:**

$$\mu(\varnothing) = \mathbb{0}. \tag{7.11}$$

- **Disjoint Additivity:** For all disjoint sets $A, B \in \Sigma$:

$$\mu(A \cup B) = \mu(A) \oplus \mu(B), \tag{7.12}$$

  which we extend to countable collections of pairwise disjoint sets.

- **Unity of Unit Square:**

$$\mu\left([0,1]^2\right) = \mathbb{1}. \tag{7.13}$$

- **Translation Invariance:** For all $A \in \Sigma$ and $v \in \mathbb{R}^2$:

$$\mu(A + v) = \mu(A). \tag{7.14}$$

- **Rotoreflection Invariance:** For all $A \in \Sigma$ and all orthonormal matrices $Q \in \mathbb{R}^{2\times2}$:

$$\mu(QA) = \mu(A). \tag{7.15}$$

- **Scaling Equivariance:** There exist a group homomorphism $\chi : (\mathbb{R}_{>0}, \times) \to (R \setminus \{\mathbb{0}\}, \otimes)$ such that for all scalings $s \in \mathbb{R}_{>0}$ and all $A \in \Sigma$:

$$\mu(sA) = \chi(s) \otimes \mu(A). \tag{7.16}$$

Definition 7.17 (Employed Semifield Measure): Let A be a measurable subset of $\mathbb{R}^2$.

a) In the linear semifield L case we use standard Lebesgue measure λ. $\mu_L(A) = \lambda(A)$. The scaling factor is $\chi(s) = s^2$.
b) In the root semifields R_p cases we use $\mu_{R_p}(A) = \sqrt[p]{\lambda(A)}$. The scaling factor is $\chi(s) = \sqrt[p]{s^2}$.
c) In the logarithmic semifields L_μ cases we use $\mu_{L_\mu}(A) = \frac{1}{\mu} \ln \lambda(A)$. The scaling factor is $\chi(s) = \frac{1}{\mu} \ln s^2$.
d) In the tropical max semifield T_+ case we use $\mu_{T_+}(A) = 0$. The scaling factor is $\chi(s) = 0$.
e) In the tropical min semifield T_- case we use $\mu_{T_-}(A) = 0$. The scaling factor is $\chi(s) = 0$.

The tropical measures might seem odd at first glance. However, remember that the semifield one $\mathbb{1}$ in the tropical semifield cases is 0 (Definition 7.2).

Definition 7.18 (Semifield Integration): Let R be a metric semifield, $S = S(\mathbb{R}^2, R)$ the space of sum-approachable functions, and μ a semifield measure. Let $\oint : (\mathrm{dom}(\oint) \subset S) \to R$ be a functional with the following properties.

- **Semifield Linearity:** For all $a, b \in R$ and $f, g \in \operatorname{dom}(\oint)$
$$\oint a \otimes f \oplus b \otimes g = a \otimes \left(\oint f\right) \oplus b \otimes \left(\oint g\right). \tag{7.17}$$
- **Indicator Function:** For all measurable sets $A \subseteq \mathbb{R}^2$ we have
$$\oint \mathbb{1}_A = \mu(A). \tag{7.18}$$
- **Translation Invariance:** For all $v \in \mathbb{R}^2$ and $f \in \operatorname{dom}(\oint)$
$$\oint \mathcal{T}_v f = \oint f. \tag{7.19}$$
- **Rotoreflection Invariance:** For all orthonormal matrices $Q \in \mathbb{R}^{2\times 2}$ and $f \in \operatorname{dom}(\oint)$:
$$\oint \mathcal{R}_Q f = \oint f. \tag{7.20}$$
- **Scaling Equivariance:** For all scalings $s \in \mathbb{R}_{>0}$ and $f \in \operatorname{dom}(\oint)$:
$$\oint \mathcal{S}_s f = \chi(s) \otimes \oint f, \tag{7.21}$$
where $\chi(s)$ is the scaling of the semifield measure $\oint$ (Definition 7.16).
- **Fubini:** For all $f : \mathbb{R}^2 \times \mathbb{R}^2 \to R$ with both $f(\cdot, y), f(x, \cdot) \in \operatorname{dom}(\oint)$, if one of the following integrals exists then they are equal:
$$\oint_y \oint_x f(x, y) = \oint_x \oint_y f(x, y). \tag{7.22}$$

We say that such a functional is a semifield integration. A function f that is in the domain of the semifield integration is called integrable.

To emphasize over what slot we are integrating we may also write $\oint_{x \in \mathbb{R}^2} f(x) = \oint f$. To emphasize over what semifield R the integration is taking place we may also write $\oint = \oint^R$.

The first two properties of the semifield integration essentially nail down what the integration has to be. That is, for every simple function $s(x) = \bigoplus_{i=1}^{n} a_i \otimes \mathbb{1}_{A_i}(x)$ we have

$$\oint s = \bigoplus_{i=1}^{n} a_i \otimes \mu(A_i) \tag{7.23}$$

by semifield linearity. This then extends naturally to sum-approachable functions $f(x) = \lim_{n \to \infty} \bigoplus_{i=1}^{n} a_i \otimes \mathbb{1}_{A_i}(x)$ by defining (with some caveats)

$$\oint f = \lim_{n \to \infty} \bigoplus_{i=1}^{n} a_i \otimes \mu(A_i). \tag{7.24}$$

The caveats here being that we need requirements on the exact nature of the sequence of simple functions for the above to be well-defined. To not get bogged down into the details we will just state what integration we will use for our relevant semifields, together with their domain of definition. In the case of the tropical semifields one can show that semifield integration is indeed the correct one (Bellaard, Sakata, Smets and Duits (2025), App.B).

Definition 7.19 (Employed Semifield Integration):

a) In the linear semifield L case we use standard Lebesgue integration

$$\oint^L f = \int_{x \in \mathbb{R}^2} f(x) \, \mathrm{d}x. \tag{7.25}$$

The domain $\mathrm{dom}\left(\oint^L\right)$ is the space of Lebesgue integrable functions.

b) In the root semifields R_p cases we use

$$\oint^{R_p} f = \sqrt[p]{\int_{x \in \mathbb{R}^2} f(x)^p \, \mathrm{d}x}. \tag{7.26}$$

The domain $\operatorname{dom}\left(\oint^{R_p}\right)$ consist of all functions f such that f^p is Lebesgue integrable.

c) In the logarithmic semifields L_μ cases we use

$$\oint^{L_\mu} f = \frac{1}{\mu} \ln \int_{x \in \mathbb{R}^2} e^{\mu f(x)} \, \mathrm{d}x. \tag{7.27}$$

The domain $\operatorname{dom}\left(\oint^{L_\mu}\right)$ consist of all functions f such that $e^{\mu f}$ is Lebesgue integrable.

d) In the tropical max semifield T_+ case we use the supremum sup.

$$\oint^{T_+} f = \sup_{x \in \mathbb{R}^2} f(x). \tag{7.28}$$

The domain $\operatorname{dom}\left(\oint^{T_+}\right)$ consist of all functions f that are bounded from above.

e) In the tropical min semifield T_- case we use the infimum inf.

$$\oint^{T_-} f = \inf_{x \in \mathbb{R}^2} f(x). \tag{7.29}$$

The domain $\operatorname{dom}\left(\oint^{T_-}\right)$ consists of all functions f that are bounded from below.

The logarithmic and root semifield integration is natural as these semifields are isomorphic to the linear semifield, see Proposition 7.1. Additionally, the tropical max and min semifield integration are related through their isomorphism $\varphi(x) = -x$. Indeed, one has $\sup_{s \in S} s = -\inf_{s \in S}\{-s\}$.

Definition 7.20 (Semifield Convolution): Let R be a metric semifield. We define the *semifield convolution* $\circledast$ of two integrable functions $f, g \in \operatorname{dom}(\oint)$ as the new function $f \circledast g \in \operatorname{dom}(\oint)$:

$$(f \circledast g)(x) := \oint_{y \in \mathbb{R}^2} f(x-y) \otimes g(y). \tag{7.30}$$

Showing that $f \circledast g$ is indeed in $\operatorname{dom}(\oint)$ is an immediate consequence of the Fubini property of semifield integration (7.22). Moreover, the Fubini property gives us that the semifield convolution is associative:

$$f \circledast (g \circledast h) = (f \circledast g) \circledast h, \tag{7.31}$$

and the translation invariance of semifield integration together with the commutativity of the semifield multiplication gives us that the semifield convolution is commutative.

We want to perform some analysis in our function spaces, so we need a (pseudo)metric $\delta : S(\mathbb{R}^2, R) \times S(\mathbb{R}^2, R) \to \mathbb{R}_{\geq 0}$ (possibly with a restricted domain). Similarly as before, when we introduced a metric on the semifields, we cannot make due with a norm on the function space as we have no additive inverses to turn the norm into a metric.

Definition 7.21 (Function Pseudometric): Let R be a semifield with metric ρ and $S = S(\mathbb{R}^2, R)$ the space of sum-approachable functions. A *function (pseudo)metric* $\delta : S \times S \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a (pseudo)metric such that for all $f, g, h \in S$ and $a \in R$ we have:

$\delta(h \oplus f, h \oplus g) \leq \delta(f, g)$,
$\delta(a \otimes f, a \otimes g) = \rho(a, \mathbb{0})\delta(f, g)$.

We allow for the (pseudo)metric to return ∞.

Again, just as in Definition 7.4, these properties are generalizations of the common notions of translation invariance and absolute homogeneity, and they ensure that both function addition $\oplus$:

$S \times S \to S$ and function scalar multiplication $\otimes : R \times S \to S$ are continuous (in both slots).

Definition 7.22 (Employed Function Pseudometric):

a) In the linear semifield L case we use

$$\delta_L(f, g) = \sqrt{\int_{\mathbb{R}^2} |f(x) - g(x)|^2 \, \mathrm{d}x}. \tag{7.32}$$

b) In the root semifield R_p case we use

$$\delta_{R_p}(f, g) = \sqrt{\int_{\mathbb{R}^2} |f(x)^p - g(x)^p|^2 \, \mathrm{d}x}. \tag{7.33}$$

c) In the logarithmic semifields L_μ case we use

$$\delta_{L_\mu}(f, g) = \sqrt{\int_{\mathbb{R}^2} |e^{\mu f(x)} - e^{\mu g(x)}|^2 \, \mathrm{d}x}. \tag{7.34}$$

d) In the tropical max semifield T_+ case we use

$$\delta_{T_+}(f, g) = \sup_{x \in \mathbb{R}^2} |e^{f(x)} - e^{g(x)}|. \tag{7.35}$$

e) In the tropical min semifield T_- case we use

$$\delta_{T_-}(f, g) = \sup_{x \in \mathbb{R}^2} |e^{-f(x)} - e^{-g(x)}|. \tag{7.36}$$

Using the function (pseudo)metric we can make an appropriate function space:

Definition 7.23 (Metric Function Space): Let R be a metric semifield, $S = S(\mathbb{R}^2, R)$ the space of sum-approachable functions, and $\delta : S \times S \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ a function (pseudo)metric. The function (pseudo)metric space $H = H(\mathbb{R}^2, R, \delta)$ is defined as

$$H := \{f \in S \mid \delta(\mathbb{0}, f) < \infty\}. \tag{7.37}$$

To turn it into an actual metric space we need to identify elements using the following natural equivalence relation $\sim$.

$$f \sim g \Leftrightarrow \delta(f, g) = 0. \tag{7.38}$$

This function space will be denoted with $\mathcal{H} = H/\sim$.

Definition 7.24 (Employed Function Spaces):

a) In the linear semifield L case we have

$$\mathcal{H}_L = L^2(\mathbb{R}^2). \tag{7.39}$$

b) In the root semifield R_p case we have

$$\mathcal{H}_{R_p} = \left\{f : \mathbb{R}^2 \to R_p \mid f^p \in L^2(\mathbb{R}^2)\right\}. \tag{7.40}$$

c) In the logarithmic semifield L_μ case we have

$$\mathcal{H}_{L_\mu} = \left\{f : \mathbb{R}^2 \to L_\mu \mid e^{\mu f} \in L^2(\mathbb{R}^2)\right\}. \tag{7.41}$$

d) In the tropical max semifield T_+ case we have

$$\mathcal{H}_{T_+} = \{f : \mathbb{R}^2 \to T_+ \ \text{ l.s.c and b.f.a }\}, \tag{7.42}$$

where l.s.c means lower semicontinuous and b.f.a means bounded from above.

e) In the tropical min semifield T_- case we have

$$\mathcal{H}_{T_-} = \{f : \mathbb{R}^2 \to T_- \ \text{ u.s.c. and b.f.b}\}, \tag{7.43}$$

where u.s.c means upper semicontinuous and b.f.b means bounded from below.

7.3.3 FOURIER TRANSFORM

We assume the existence of an injective Fourier transform that need only work on a very restricted class of semifield integrable functions.

Definition 7.25 (Semifield Fourier Transform): Let R be a metric semifield. A *semifield Fourier Transform* $\mathcal{F}_R : (\operatorname{dom}(\mathcal{F}_R) \subseteq \operatorname{dom}(\oint)) \to \operatorname{dom}(\oint)$ is an operator satisfying (where we drop the subscript R for conciseness):

- **Semifield Linearity:** For all $a, b \in R$ and $f, g \in \operatorname{dom}(\mathcal{F})$
$$\mathcal{F}(a \otimes f \oplus b \otimes g) = a \otimes (\mathcal{F} f) \oplus b \otimes (\mathcal{F} g). \tag{7.44}$$
- **Convolution Property:** For all $f, g \in \operatorname{dom}(\mathcal{F})$ with $f \circledast g \in \operatorname{dom}(\mathcal{F})$
$$\mathcal{F}(f \circledast g) = (\mathcal{F} f) \otimes (\mathcal{F} g). \tag{7.45}$$
- **Rotoreflection Equivariance:** For all orthonormal matrices $Q \in \mathbb{R}^{2\times 2}$
$$\mathcal{F} \circ \mathcal{R}_Q = \mathcal{R}_{Q^{-\top}} \circ \mathcal{F}. \tag{7.46}$$
- **Scaling Equivariance:** For all scalings $s \in \mathbb{R}_{>0}$:
$$\mathcal{F} \circ \mathcal{S}_s = \chi(s) \otimes \mathcal{S}_{\frac{1}{s}} \circ \mathcal{F}, \tag{7.47}$$
  where $\chi(s)$ is the scaling of the semifield measure (Definition 7.16).
- **Invertibility:** The domain $\operatorname{dom}(\mathcal{F})$ is chosen such that the transform is injective and thus invertible on its image.

In the next definition we will specify the choice of semifield Fourier transform together with its appropriate choice of domain for all the semifields we consider (Definition 7.2). The choices we make here are sometimes more restrictive than strictly needed, but, as we will see in Subsection 7.5.2, we only need to be able to take the semifield Fourier transform of a very “small” set of functions.

Definition 7.26 (Employed Semifield Fourier Transform):

a) In the linear semifield L case we use

$$(\mathcal{F}_L f)(\omega) = \int_{\mathbb{R}^2} f(x) e^{-i\omega\cdot x}\,\mathrm{d}x. \tag{7.48}$$

The domain $\operatorname{dom}(\mathcal{F}_L)$ is chosen to be the space of even, continuous, and absolutely integrable functions, with absolutely integrable Fourier transforms. The inverse on its image is

$$\left(\mathcal{F}_L^{-1}\hat{f}\right)(x) = \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \hat{f}(\omega) e^{i\omega\cdot x}\,\mathrm{d}\omega. \tag{7.49}$$

b) In the root semifield R_p case we use

$$\left(\mathcal{F}_{R_p} f\right)(\omega) = \sqrt[p]{\int_{\mathbb{R}^2} f(x)^p e^{-i\omega\cdot x}\,\mathrm{d}x}. \tag{7.50}$$

The domain $\operatorname{dom}\left(\mathcal{F}_{R_p}\right)$ is chosen such that f^p is in the domain of the linear Fourier transform $\operatorname{dom}(\mathcal{F}_L)$, together with the restriction that the input of the p'th root is nonnegative. The inverse on its image is

$$\left(\mathcal{F}_{R_p}^{-1}\hat{f}\right)(x) = \sqrt[p]{\frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \hat{f}(\omega)^p e^{i\omega\cdot x}\,\mathrm{d}\omega}. \tag{7.51}$$

c) In the logarithmic semifield L_μ case we use

$$\left(\mathcal{F}_{L_\mu} f\right)(\omega) = \frac{1}{\mu}\ln\int_{x\in\mathbb{R}^2} e^{\mu f(x)} e^{-i\omega\cdot x}\,\mathrm{d}x. \tag{7.52}$$

The domain $\operatorname{dom}\left(\mathcal{F}_{L_\mu}\right)$ is chosen such that $e^{\mu f}$ is in the domain of the linear Fourier transform $\operatorname{dom}(\mathcal{F}_L)$, together with the restriction that the input of the natural logarithm is positive. The inverse on its image is

$$\left(\mathcal{F}_{L_\mu}^{-1}\hat{f}\right)(x) = \frac{1}{\mu}\ln\left(\frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} e^{\mu\hat{f}(\omega)} e^{i\omega\cdot x}\,\mathrm{d}\omega\right) \tag{7.53}$$

d) In the tropical max semifield T_+ case we use

$$\left(\mathcal{F}_{T_+} f\right)(\omega) = \sup_{x \in \mathbb{R}^2} f(x) - \omega \cdot x. \tag{7.54}$$

The domain $\operatorname{dom}\left(\mathcal{F}_{T_+}\right)$ is chosen to be the space of even, continuous, concave, superlinear functions. The inverse on its image is

$$\left(\mathcal{F}_{T_+}^{-1} \hat{f}\right)(x) = \inf_{\omega \in \mathbb{R}^2} \hat{f}(\omega) + \omega \cdot x. \tag{7.55}$$

e) In the tropical min semifield T_- case we use

$$\left(\mathcal{F}_{T_-} f\right)(\omega) = \inf_{x \in \mathbb{R}^2} f(x) - \omega \cdot x. \tag{7.56}$$

The domain $\operatorname{dom}\left(\mathcal{F}_{T_-}\right)$ is chosen to be the space of even, continuous, convex, superlinear functions. The inverse on its image is

$$\left(\mathcal{F}_{T_-}^{-1} \hat{f}\right)(x) = \sup_{\omega \in \mathbb{R}^2} \hat{f}(\omega) + \omega \cdot x. \tag{7.57}$$

Lemma 7.4: The employed semifield Fourier transforms satisfy Definition 7.25.

Proof: In the linear semifield L case we know that the familiar Fourier transform satisfies the definition.

As for the root and logarithmic semifields, being isomorphic to the linear semifield, we can quickly deduce that they also satisfy the definitions through the equalities

$$\begin{aligned} \mathcal{F}_{L_\mu} &= \mathcal{P}_{\varphi_\mu^{-1}} \circ \mathcal{F}_L \circ \mathcal{P}_{\varphi_\mu}, \\ \mathcal{F}_{R_p} &= \mathcal{P}_{\varphi_p^{-1}} \circ \mathcal{F}_L \circ \mathcal{P}_{\varphi_p}, \end{aligned} \tag{7.58}$$

where $\mathcal{P}$ is the pointwise operator (7.7), $\varphi_\mu(x) = e^{\mu x}$ the semifield isomorphism $\varphi_\mu : L_\mu \to L_{\geq 0}$, and $\varphi_p(x) = x^p$ the semifield isomorphism $\varphi_p : R_p \to L_{\geq 0}$. For example, to show that $\mathcal{F}_{L_\mu}$ satisfies the convolution property:

$$\begin{aligned}
&\mathcal{F}_{L_\mu}(f \circledast g) \\
&= \mathcal{P}_{\varphi_\mu^{-1}} \mathcal{F}_L \mathcal{P}_{\varphi_\mu}(f \circledast g) \\
&= \mathcal{P}_{\varphi_\mu^{-1}} \mathcal{F}_L \left(\left(\mathcal{P}_{\varphi_\mu} f \right) * \left(\mathcal{P}_{\varphi_\mu} g \right) \right) \\
&= \mathcal{P}_{\varphi_\mu^{-1}} \left(\left(\mathcal{F}_L \mathcal{P}_{\varphi_\mu} f \right) \times \left(\mathcal{F}_L \mathcal{P}_{\varphi_\mu} g \right) \right) \\
&= \left(\mathcal{P}_{\varphi_\mu^{-1}} \mathcal{F}_L \mathcal{P}_{\varphi_\mu} f \right) \otimes \left(\mathcal{P}_{\varphi_\mu^{-1}} \mathcal{F}_L \mathcal{P}_{\varphi_\mu} g \right) \\
&= \left(\mathcal{F}_{L_\mu} f \right) \otimes \left(\mathcal{F}_{L_\mu} g \right),
\end{aligned} \tag{7.59}$$

where $\circledast$ and $\otimes$ are the semifield convolution and multiplication of L_μ and where $\times$ denotes the standard pointwise product of functions. In the above derivation we have used that

$$\begin{aligned}
\mathcal{P}_{\varphi_\mu}(f \circledast g) &= \left(\mathcal{P}_{\varphi_\mu} f \right) * \left(\mathcal{P}_{\varphi_\mu} g \right), \\
\mathcal{P}_{\varphi_\mu^{-1}}(f \times g) &= \left(\mathcal{P}_{\varphi_\mu^{-1}} f \right) \otimes \left(\mathcal{P}_{\varphi_\mu^{-1}} g \right),
\end{aligned} \tag{7.60}$$

and that $\mathcal{F}_L$ has the convolution property.

Consider now the tropical max semifield T_+. That $\mathcal{F}_{T_+}$ satisfies the linearity, equivariance, and the zero-frequency properties is immediate. As for the convolution property we have

$$\begin{aligned}
&\big(\mathcal{F}_{T_+}(f \circledast g)\big)(\omega) \\
&= \sup_x \, (f \circledast g)(x) - \omega \cdot x \\
&= \sup_x \left(\sup_y f(x-y) + g(y) \right) - \omega \cdot x \\
&= \sup_x \left(\sup_{x_1 + x_2 = x} f(x_1) + g(x_2) \right) - \omega \cdot x \\
&= \sup_{x_1, x_2} f(x_1) + g(x_2) - \omega \cdot (x_1 + x_2) \\
&= \left(\sup_{x_1} f(x_1) - \omega \cdot x_1 \right) + \left(\sup_{x_2} g(x_2) - \omega \cdot x_2 \right) \\
&= \big(\mathcal{F}_{T_+} f\big)(\omega) \otimes \big(\mathcal{F}_{T_+} g\big)(\omega),
\end{aligned} \tag{7.61}$$

where $\otimes$ and $\circledast$ are the tropical max T_+ multiplication and convolution. For the invertibility we refer to the *Fenchel biconjugation theorem* (Borwein and Lewis (2006), Thm.4.2.1). That the tropical min semifield Fourier transform $\mathcal{F}_{T_-}$ satisfies all properties follows from the fact that T_- is semifield isomorphic to T_+ with the isomorphism being $\Phi(x) = -x$. □

Remark 7.1: The Laplace-like transform

$$(\mathcal{L}f)(\omega) = \int_{\mathbb{R}^2} f(x) e^{-\omega \cdot x} \, \mathrm{d}x \tag{7.62}$$

also satisfies Definition 7.25. However, and this is also mentioned in Schmidt and Weickert (2016), this transform is limited in its applicability because it is only finitely-valued for functions with super-exponential decay (the transform is two-sided thus resulting in this extreme condition). Given this limitation of this transform, we instead use the normal Fourier transform.

Remark 7.2: The above semifield Fourier transforms typically relate to transforms of the type

$$(\mathcal{F} f)(\omega) = \oint f(x) \otimes \chi_\omega(x), \tag{7.63}$$

where χ_ω is an irreducible semifield-linear representation of $\mathbb{R}^2$, but we choose to express them in common Fourier/Fenchel transforms to keep a clear track of function space restrictions.

Even though we have used complex numbers in the Fourier transforms, the resulting transformed functions are always of the proper form $\mathbb{R}^2 \to R$ due the domain consisting of even functions. This means that we could have freely replaced the $e^{-i\omega\cdot x}$ with $\cos(\omega \cdot x)$. In other words, we could have instead used the *Fourier cosine transform*.

7.4 SEMIFIELD SCALE-SPACE

In this section we will state and motivate the semifield scale-space axioms, consider some examples semifield scale-spaces, and define what we mean with isomorphic scale-spaces.

7.4.1 AXIOMS

In PAUWELS, VAN GOOL, FIDDELAERS AND MOONS (1995) it is stated that “The only really nontrivial (and possibly too restrictive) assumption imposed on the scale-space operators, is that of linearity.” By generalizing to semifield linearity we sidestep this restrictive assumption, without making the theory too abstract to be practically useful.

Let us shortly motivate the semifield scale-space axioms from a machine-learning perspective (building upon similar findings in mathematical deep learning WORRALL AND WELLING (2019), SOSNOVIK, SZMAJA AND SMEULDERS (2020), SANGALLI, BLUSSEAU, VELASCO-FORERO AND ANGULO (2021), LINDEBERG (2022)). The semifield linearity (Axiom 7.1) together with the translation equivariance

(Axiom 7.5) will induce semifield convolutions that allow for fast parallel computations. The one-parameter semigroup property (Axiom 7.2) together with the strong continuity (Axiom 7.3) provides consistency and stability over evolution time. The one-parameter semigroup property (Axiom 7.2) together with the scaling equivariance (Axiom 7.4) allows us to constrain ourselves to a fixed end-time in a PDE sublayer, say $t = 1$, without loss of generality, further reducing the total parameter count. The scaling, translation, and roto-reflection equivariance (Axiom 7.4, Axiom 7.5 and Axiom 7.6) of the scale-space allows for the design of inherently equivariant networks, resulting in an architecture that is robust and data-efficient (Mohamed, Cesa, Cohen and Welling (2020), Cohen, Weiler, Kicanaoglu and Welling (2019)).

Definition 7.27 (Semifield Scale-space): Let R be a one-dimensional metric semifield (Definition 7.6), $\mathcal{H} = \mathcal{H}(\mathbb{R}^2, R, \delta)$ a corresponding metric function space (Definition 7.23), and $\Phi_t : \mathcal{H} \to \mathcal{H}$ be a family of operators, indexed by $t \geq 0$. We call Φ_t a *semifield scale-space* if it satisfies the following axioms:

Axiom 7.1 (Semifield Linearity and Integral Operator): We require that Φ is R-linear, that is for all $f, g \in \mathcal{H}$ and $a, b \in R$:

$$\Phi_t(a \otimes f \oplus b \otimes g) = a \otimes (\Phi_t f) \oplus b \otimes (\Phi_t g). \tag{7.64}$$

More specifically, for positive time $t > 0$ we will assume that Φ_t can be written as an integral operator:

$$(\Phi_t f)(x) = \oint_{y \in \mathbb{R}^2} \kappa_t(x, y) \otimes f(y), \tag{7.65}$$

for some continuous kernel $\kappa_t : \mathbb{R}^2 \times \mathbb{R}^2 \to R$ with $\kappa_t(x, \cdot)$ within the domain of the semifield Fourier transform $\mathcal{F}_R$ (Definition 7.25).

Axiom 7.2 (One-Parameter Semigroup): We require that Φ_t forms a one-parameter semigroup , that is for all $t, s \geq 0$:

$$\Phi_t \circ \Phi_s = \Phi_{t+s} \quad \text{and} \quad \Phi_0 = \text{id}, \tag{7.66}$$

where id is the identity map on $\mathcal{H}$.

Axiom 7.3 (Strong Continuity): We require that $\Phi_t f$ is continuous with respect to time t at any $t_0 > 0$ for all $f \in \mathcal{H}$:

$$\lim_{t \to t_0} (\Phi_t f) = \Phi_{t_0} f, \tag{7.67}$$

where the limit is taken in the metric function space $\mathcal{H}$ (Definition 7.23).

Axiom 7.4 (Scaling Equivariance): There exists a *scaling power* $\alpha > 0$ such that for all scalings $s > 0$ and all times $t \geq 0$:

$$\Phi_t \circ \mathcal{S}_s = \mathcal{S}_s \circ \Phi_{t/s^\alpha}, \tag{7.68}$$

where $\mathcal{S}_s$ is the scaling operator (7.6).

Axiom 7.5 (Translation Equivariance): We require that Φ commutes with all translations $v \in \mathbb{R}^2$:

$$\Phi_t \circ \mathcal{T}_v = \mathcal{T}_v \circ \Phi_t, \tag{7.69}$$

where $\mathcal{T}_v$ is the translation operator (7.4).

Axiom 7.6 (Rotoreflection Equivariance): We require that Φ commutes with all orthonormal matrices $Q \in \mathbb{R}^{2\times 2}$:

$$\Phi_t \circ \mathcal{R}_Q = \mathcal{R}_Q \circ \Phi_t, \tag{7.70}$$

where $\mathcal{R}_Q$ is the rotoreflection operator (7.5).

Note that in our axioms we do not impose any restriction on the creation of new structures when transitioning from finer to coarser scales. This is in contrast to the requirement of *causality* or *non-enhancement of local extrema* in Koenderink (1984) and Lindeberg (1997).

In the linear semifield L case the linearity axiom in some sense already implies the integral operator axiom. The precise statement is known as the Schwartz kernel theorem, a main result in the theory of generalized functions/distributions. In the tropical semifield case a similar statement can be made, as demonstrated in Kolokoltsov and Maslov (1997), Thm.2.1. But for other semifields such a statement cannot be made just yet. For simplicity, and to be on the safe side, we therefore assume the integral operator axiom.

The one-parameter semigroup property is a natural axiom in the sense that it implies that the (infinitesimal) evolution "looks the same" at all times t. More precisely, the (strongly continuous) one-parameter semigroup property relates to the existence of a single generator that encapsulates the whole operator family. To understand, consider the linear semifield case, some initial $f_0 \in \mathcal{H}$, and its evolution $f_t := \Phi_t(f_0)$. From the one-parameter semigroup axiom we have:

$$\begin{aligned} f_{t+h} - f_t &= \Phi_{h+t} f_0 - \Phi_t f_0 \\ &= \Phi_h \Phi_t f_0 - \Phi_t f_0 \\ &= (\Phi_h - \Phi_0) f_t. \end{aligned} \tag{7.71}$$

dividing by h and taking the limit $h \downarrow 0$ in conjunction with strong continuity, we get the time-invariant evolution equation:

$$\frac{\mathrm{d}f_t}{\mathrm{d}t} = \Psi f_t, \text{ where } \Psi := \lim_{h \downarrow 0} \frac{\Phi_h - \Phi_0}{h}. \tag{7.72}$$

The operator $\Psi : D(\Psi) \to \mathcal{H}$ is called the generator of the operator family Φ_t and its natural domain $D(\Psi)$ consists of all functions $f \in \mathcal{H}$ for which the above limit makes sense. Typically, this domain will be dense in $\mathcal{H}$.

Thus, we can interpret Φ_t as the solution operator of an evolution equation. Given that the generator exists, it is possible through various means, for example the spectral theorem, to give meaning to the expression:

$$\Phi_t = e^{t\Psi}, \tag{7.73}$$

which can be used to quickly confirm (at least formally) that:

$$\frac{\mathrm{d}\Phi_t}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}(e^{t\Psi}) = \Psi e^{t\Psi} = \Psi \Phi_t, \tag{7.74}$$

which corresponds what we already saw in (7.72).

Remark 7.3: Given the existence of a generator Ψ, the scaling equivariance axiom can be equivalently written as $\Psi \circ \mathcal{S}_s = \frac{1}{s^\alpha} \mathcal{S}_s \circ \Psi$, revealing that the scaling equivariance can also be understood as a sort of α-homogeneity of the generator.

An easy and illustrative example of a generator together with its operator family is the derivative operator and the family of translation operators in one-dimensional space:

$$(\Phi_t f)(x) = f(x + t), \quad \Psi = \frac{\mathrm{d}}{\mathrm{d}x}. \tag{7.75}$$

A well-known related theorem in functional analysis is Stone's Theorem. This theorem shows that there is a one-to-one correspondence between strongly continuous unitary one-parameter semigroups and (possible unbounded) densely defined self-adjoint operators on a Hilbert space.

In our case getting everything precise is made difficult by the fact that we want to generalize to semifields other than the linear semifield. For example, we cannot even directly make sense of (7.72) for general semifields as there is not necessarily a $-$ operation: we only have $\oplus$. Given these obstacles, we will not attempt to rigorously prove that every semifield scale-spaces corresponds to a PDE, but will state the related PDEs in our primary cases of interest (Definition 7.28).

The scaling equivariance says that the scale-space representation of a scaled image should be a scaled version of the scale-space representation of the original image. In a sense we want a scale-space that does not “care” about absolute scale: it should qualitatively looks the same no matter the starting scale of the input. The translation and rotoreflection equivariance requirements are also not surprising: the Euclidean plane has its natural translation and rotoreflection symmetries, and demanding the scale-space to respect these is commonplace.

In PDE-based neural networks the trainable parameters take the form of Riemannian metrics $\mathcal{G}$ on a homogeneous space M. In the case of PDE-CNNs, that being $M = \mathbb{R}^2$, this reduces to an inner product $\mathcal{G} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ which we can always write as $\mathcal{G}(x, y) = x^\top G y$, where G is the corresponding *Gram matrix*. In the axioms, we implicitly make, without loss of generality, the assumption that we use the “standard” inner product on $\mathbb{R}^2$, namely $\mathcal{G}(x, y) = x^\top y$. Later in Section 7.6 we explain how we bring back general inner products in the PDE-CNN architecture.

7.4.2 EXAMPLES

Definition 7.28 (Scale-spaces of Interest):

a) The Gaussian scale-space over the linear semifield L:

$$\begin{aligned}(\Phi_t f)(x) &= \int_{y\in\mathbb{R}^2} \kappa_t(x,y) \times f(y)\ \mathrm{d}y,\\ \kappa_t(x,y) &= \frac{1}{2\pi t}\exp\left(-\frac{1}{2}\frac{\|x-y\|^2}{t}\right),\end{aligned} \tag{7.76}$$

which correspond to solutions of

$$\frac{\partial f}{\partial t} = \frac{1}{2}\Delta f. \tag{7.77}$$

The scaling power is $\alpha = 2$.

b) The (quadratic) root scale-spaces over the root semifields R_p:

$$\begin{aligned}(\Phi_t f)(x) &= \oint_{y\in\mathbb{R}^2}^{R_p} \kappa_t(x,y) \times f(y),\\ \kappa_t(x,y) &= \frac{1}{\sqrt[p]{2\pi t}}\exp\left(-\frac{1}{2p}\frac{\|x-y\|^2}{t}\right),\end{aligned} \tag{7.78}$$

which correspond to solutions of

$$\frac{\partial f}{\partial t} = \frac{p-1}{f}\frac{1}{2}\|\nabla f\|^2 + \frac{1}{2}\Delta f. \tag{7.79}$$

The scaling power is $\alpha = 2$.

c) The (quadratic) logarithmic scale-spaces over the logarithmic semifields L_μ:

$$\begin{aligned}(\Phi_t f)(x) &= \oint_{y\in\mathbb{R}^2}^{L_\mu} \kappa_t(x,y) + f(y),\\ \kappa_t(x,y) &= -\frac{1}{\mu}\ln(2\pi t) - \frac{1}{2\mu}\frac{\|x-y\|^2}{t},\end{aligned} \tag{7.80}$$

which correspond to solutions of

$$\frac{\partial f}{\partial t} = \mu \frac{1}{2} \|\nabla f\|^2 + \frac{1}{2} \Delta f. \tag{7.81}$$

The scaling power is $\alpha = 2$.

d) The α-dilation scale-space over the tropical max semifield T_+:

$$\begin{aligned} (\Phi_t f)(x) &= \sup_{y \in \mathbb{R}^2} \kappa_t(x, y) + f(y), \\ \kappa_t(x, y) &= -\frac{t}{\beta} \left(\frac{\|x - y\|}{t} \right)^\beta, \end{aligned} \tag{7.82}$$

with $\frac{1}{\alpha} + \frac{1}{\beta} = 1$, which correspond to (viscosity) solutions of

$$\frac{\partial f}{\partial t} = \frac{1}{\alpha} \|\nabla f\|^\alpha. \tag{7.83}$$

The scaling power is α.

e) The α-erosion scale-space over the tropical min semifield T_-:

$$\begin{aligned} (\Phi_t f)(x) &= \inf_{y \in \mathbb{R}^2} \kappa_t(x, y) + f(y), \\ \kappa_t(x, y) &= \frac{t}{\beta} \left(\frac{\|x - y\|}{t} \right)^\beta, \end{aligned} \tag{7.84}$$

with $\frac{1}{\alpha} + \frac{1}{\beta} = 1$, which correspond to (viscosity) solutions of

$$\frac{\partial f}{\partial t} = -\frac{1}{\alpha} \|\nabla f\|^\alpha. \tag{7.85}$$

The scaling power is α.

The operators Φ_t above solve the corresponding PDEs and one readily checks that the kernels $\kappa_t(\cdot, y)$ satisfy the PDE for all $y \in \mathbb{R}^2$. For example, consider the quadratic ($\alpha = 2$) dilation scale-space and $k_t(x) := \kappa_t(x, 0) = -\frac{1}{2} \frac{\|x\|^2}{t}$:

$$\frac{\partial k_t}{\partial t} = \frac{1}{2} \frac{\|x\|^2}{t^2}, \quad \|\nabla k_t\|^2 = \frac{\|x\|^2}{t^2}. \tag{7.86}$$

So, indeed, k_t satisfies the dilation PDE. The same check can be done for the other scale-spaces.

Remark 7.4: The Schrödinger equation also generates a scale-space representation in the space $L^2(\mathbb{R}^2; \mathbb{C})$, in the sense that it satisfies the linearity axiom and axioms 2-6. However, it does not fit in the theory here as the complex numbers do not form a one-dimensional semifield, and the corresponding kernel $\kappa_t(x, \cdot)$ is not square integrable. In Kraakman (2023) PDE-based neural networks using the Schrödinger equation are investigated and implemented. They require more memory than our classical PDE-Based CNNs for only a small accuracy gain in practice so far.

7.4.3 ISOMORPHIC SCALE-SPACES

In Proposition 7.1 we saw that the nonnegative linear, root, and logarithmic semifields are isomorphic, with the same being true for the tropical ones. It seems natural then that there also exist isomorphisms between the corresponding semifield scale-spaces. Let us start by clarifying what we mean by two semifield scale-spaces being isomorphic.

Definition 7.29 (Semifield Scale-space Isomorphism): Let R and $\tilde{R}$ be two semifields, and let Φ_t and $\tilde{\Phi}_t$ be two semifield scale-spaces over R and $\tilde{R}$ respectively. We say the two scale-spaces are isomorphic if there exists a semifield isomorphism $\varphi : \tilde{R} \to R$ such that

$$\Phi_t \circ \mathcal{P}_\varphi = \mathcal{P}_\varphi \circ \tilde{\Phi}_t, \tag{7.87}$$

where $\mathcal{P}_\varphi : F(\mathbb{R}^2, \tilde{R}) \to F(\mathbb{R}^2, R)$ is the pointwise operator (7.7).

Indeed, one can check that in this sense the scale-spaces of interest are isomorphic in the following way, akin to Proposition 7.1.

Proposition 7.2 (Scale-Space Isomorphisms):

- The quadratic root scale-spaces over the root semifields R_p are isomorphic to the Gaussian scale-space over the nonnegative linear semifield $L_{\geq 0}$.
- The quadratic logarithmic scale-spaces over the logarithmic semifields L_μ are isomorphic to the Gaussian scale-space over the nonnegative linear semifield $L_{\geq 0}$.
- The α-dilation scale-space over the tropical max semifield T_+ is isomorphic to the α-erosion scale-space over the tropical min semifield T_-.
- Informally, in the limit $\mu \to \pm\infty$ the quadratic logarithmic scale-spaces over the logarithmic semifields L_μ "converge" to the quadratic ($\alpha = 2$) dilation and erosion scale-spaces of the tropical semifields $T_\pm$.

Definition 7.29 also gives us a way to create new (isomorphic) semifield scale-spaces from existing ones in the following way. Take any existing semifield scale-space Φ_t over a semifield R, and let $\varphi : \tilde{R} \to R$ be any semifield isomorphism. We then simply define the new scale-space $\tilde{\Phi}_t = \mathcal{P}_{\varphi^{-1}} \circ \Phi_t \circ \mathcal{P}_\varphi$ over the semifield $\tilde{R}$.

In Florack (2001) nonlinear scale-spaces are created in exactly this way by performing a pointwise transformation on the Gaussian scale-space and deducing what nonlinear PDE corresponds to the obtained evolution.

More specifically, let $\varphi : R \to L$ be a monotonic twice continuously differentiable transformation, where $R \subseteq \mathbb{R}$ is some subset of the reals. We start with the isotropic diffusion PDE on $\mathbb{R}^2$:

$$\frac{\partial f}{\partial t} = \frac{1}{2} \Delta f. \tag{7.88}$$

We then define a new evolution $g(x, t) := \varphi^{-1}(f(x, t))$. Let us derive what PDE g obeys. Using the shorthand $\varphi(g) := \mathcal{P}_\varphi(g)$ for a moment, it follows from the equalities:

$$\begin{aligned} \frac{\partial g}{\partial t} &= \frac{\partial}{\partial t}\varphi^{-1}(f) = \frac{1}{\varphi'(g)}\frac{\partial f}{\partial t}, \\ \Delta f &= \Delta(\varphi(g)) = \varphi''(g)\|\nabla g\|^2 + \varphi'(g)\Delta g, \end{aligned} \tag{7.89}$$

that the PDE that describes the evolution of g is:

$$\frac{\partial g}{\partial t} = \frac{\varphi''(g)}{\varphi'(g)}\frac{1}{2}\|\nabla g\|^2 + \frac{1}{2}\Delta g. \tag{7.90}$$

Florack suggests setting $\mu := \varphi''/\varphi' = (\ln\varphi')'$ as a constant, as the class of non-trivial (that being non-affine) φ's is then:

$$\varphi(x) = e^{\mu x} \quad \text{with} \quad \mu \neq 0, \mu \in \mathbb{R}, \tag{7.91}$$

up to affine transformations. This transformation of the PDE is known as the *Cole-Hopf transformation* (Evans (2010), p.195). This is exactly the isomorphism between the logarithmic and linear semifield as seen in Proposition 7.1, and indeed, with this φ we get the quadratic logarithmic scales spaces:

$$\frac{\partial g}{\partial t} = \frac{\mu(1)}{2}\|\nabla g\|^2 + \frac{1}{2}\Delta g. \tag{7.92}$$

In the extreme cases of the transformation, that being $\mu = \pm\infty$, the diffusion part becomes negligible in comparison to the erosion/dilation part, and one can say that the morphological scale-spaces arise.

If we instead choose $\varphi(x) = x^p$ the quadratic root scale-spaces arise:

$$\frac{\partial g}{\partial t} = \frac{p-1}{g}\frac{1}{2}\|\nabla g\|^2 + \frac{1}{2}\Delta g. \tag{7.93}$$

7.5 CONSEQUENCES

In this section we explore the consequences of the semifield scale-spaces axioms (Definition 7.27).

We start by showing that the equivariance axioms of the scale-space representation lead to invariance properties of the kernel $\kappa_t : \mathbb{R}^2 \times \mathbb{R}^2 \to R$. For example, the translation equivariance of Φ_t (Axiom 7.5) implies that the kernel κ_t (Axiom 7.1) is translation invariant in the sense that $\kappa_t(v+x, v+y) = \kappa_t(x, y)$ for all $x, y, v \in \mathbb{R}^2$.

We then show that, due to the translation equivariance (Axiom 7.5), the semifield scale-space can be written as a semifield convolution with a reduced kernel $k_t : \mathbb{R}^2 \to R$. From there on out we show Theorem 7.1 which gives an explicit form of the reduced kernel k_t in the semifield Fourier domain, this being the main theorem.

7.5.1 Equivariance of Operator becomes Invariance of Kernel

In this subsection we show how the translation, rotoreflection, and scaling equivariance axioms on the operator family Φ_t translate to corresponding invariances on the kernel κ_t. The upcoming three lemmas are straightforward, generally known, and basically identical in proof.

Lemma 7.5 (Translation Invariance): From the integral operator (Axiom 7.1) and translation equivariance (Axiom 7.5) it follows that the kernel is translation invariant, that is:

$$\kappa_t(v+x, v+y) = \kappa_t(x, y), \tag{7.94}$$

for all for all $x, y, v \in \mathbb{R}^2$ and $t > 0$.

Proof: We rewrite the translation equivariance (Axiom 7.5) as

$$\mathcal{T}_{-v} \circ \Phi_t \circ \mathcal{T}_v = \Phi_t. \tag{7.95}$$

We apply some dummy function $f \in \mathcal{H}$ and evaluate it at some dummy position $x \in \mathbb{R}^2$:

$$((\mathcal{T}_{-v} \circ \Phi_t \circ \mathcal{T}_v)(f))(x) = (\Phi_t(f))(x). \tag{7.96}$$

Using the definition of the translation operator $\mathcal{T}_v$ (7.4) and the integral operator axiom we expand this into:

$$\oint_{y \in \mathbb{R}^2} \kappa_t(v+x, y) \otimes f(-v+y) = \oint_{y \in \mathbb{R}^2} \kappa_t(x, y) \otimes f(y). \tag{7.97}$$

Using the translation invariance property (7.19) of the semifield integration gives:

$$\oint_{y \in \mathbb{R}^2} \kappa_t(v+x, v+y) \otimes f(y) = \oint_{y \in \mathbb{R}^2} \kappa_t(x, y) \otimes f(y). \tag{7.98}$$

Given that this should hold for all $f \in \mathcal{H}$ we can conclude:

$$\kappa_t(v+x, v+y) = \kappa_t(x, y). \tag{7.99}$$

□

The last step of the proof can be understood as a semifield version of *Fundamental lemma of the calculus of variations*. There are several versions of this lemma but generally they are of the form

$$\forall g : \int fg = 0 \Rightarrow f = 0, \tag{7.100}$$

with some assumptions on the nature of f and g. Using the substitution $f = f_1 - f_2$ we can equivalently write

$$\forall g : \int f_1 g = \int f_2 g \Rightarrow f_1 = f_2. \tag{7.101}$$

Because a semifield does necessarily have additive inverses, the second form (7.101) is the natural one in our context. The lemma in its second form (7.101) is used in the last step of the proof of Lemma 7.5, specifically when transitioning from (7.98) to (7.99). We will also apply it in the upcoming proofs of Lemma 7.6 and Lemma 7.7. The proof of the lemma is straightforward in our setting; the kernel κ_t is assumed to be continuous (Axiom 7.1), and the space $\mathcal{H}$ contains all indicator functions, meaning that a standard "concentration" argument works.

Lemma 7.6 (Rotoreflection Invariance): From the integral operator (Axiom 7.1) and the rotoreflection equivariance (Axiom 7.6) it follows that the kernel is rotoreflection invariant, that is:

$$\kappa_t(Qx, Qy) = \kappa_t(x, y), \tag{7.102}$$

for all orthonormal $Q \in \mathbb{R}^{2\times 2}$, $x, y \in \mathbb{R}^2$, and $t > 0$.

Proof: We rewrite the rotoreflection equivariance (Axiom 7.6) as

$$\mathcal{R}_{Q^{-1}} \circ \Phi_t \circ \mathcal{R}_Q = \Phi_t. \tag{7.103}$$

We apply some dummy function $f \in \mathcal{H}$ and evaluate it at some dummy position $x \in \mathbb{R}^2$:

$$\big((\mathcal{R}_{Q^{-1}} \circ \Phi_t \circ \mathcal{R}_Q)(f)\big)(x) = (\Phi_t(f))(x). \tag{7.104}$$

Using the definition of the rotoreflection operator $\mathcal{R}_Q$ (7.5) and integral operator axiom we expand this to:

$$\oint_{y\in\mathbb{R}^2} \kappa_t(Qx, y) \otimes f(Q^{-1}y) = \oint_{y\in\mathbb{R}^2} \kappa_t(x, y) \otimes f(y). \tag{7.105}$$

Using the rotoreflection invariance property (7.20) of the semifield integration gives:

$$\oint_{y\in\mathbb{R}^2} \kappa_t(Qx, Qy) \otimes f(y) = \oint_{y\in\mathbb{R}^2} \kappa_t(x, y) \otimes f(y). \tag{7.106}$$

Given that this should hold for all $f \in \mathcal{H}$ we can conclude:

$$\kappa_t(Qx, Qy) = \kappa_t(x, y). \tag{7.107}$$

□

Lemma 7.7 (Scale Invariance): From the integral operator (Axiom 7.1) and scaling equivariance (Axiom 7.4) it follows that the kernel is scale invariant, that is:

$$\chi(s) \otimes \kappa_{s^\alpha t}(sx, sy) = \kappa_t(x, y). \tag{7.108}$$

for all $x, y \in \mathbb{R}^2$ and $s, t > 0$.

Proof: We rewrite the scaling equivariance (Axiom 7.4) as

$$\mathcal{S}_{\frac{1}{s}} \circ \Phi_{s^{\alpha}t} \circ \mathcal{S}_s = \Phi_t. \tag{7.109}$$

We apply some dummy function $f \in \mathcal{H}$ and evaluate it at some dummy position $x \in \mathbb{R}^2$:

$$\left(\left(\mathcal{S}_{\frac{1}{s}} \circ \Phi_{s^{\alpha}t} \circ \mathcal{S}_s\right)(f)\right)(x) = (\Phi_t(f))(x). \tag{7.110}$$

Using the definition of the scaling operator $\mathcal{S}_s$ (7.6) and integral operator axiom we expand this to:

$$\oint_{y\in\mathbb{R}^2} \kappa_{s^{\alpha}t}(sx, y) \otimes f\left(\frac{y}{s}\right) = \oint_{y\in\mathbb{R}^2} \kappa_t(x, y) \otimes f(y). \tag{7.111}$$

Using the scaling property (7.21) of the semifield integration gives:

$$\chi(s) \otimes \oint_y \kappa_{s^{\alpha}t}(sx, sy) \otimes f(y) = \oint_y \kappa_t(x, y) \otimes f(y). \tag{7.112}$$

Given that this should hold for all $f \in \mathcal{H}$ we conclude:

$$\chi(s) \otimes \kappa_{s^{\alpha}t}(sx, sy) = \kappa_t(x, y). \tag{7.113}$$

□

Consider now Lemma 7.5. Because we can freely choose the translation v, we can also choose $v = -y$:

$$\kappa_t(x, y) = \kappa_t(x - y, 0) =: k_t(x - y). \tag{7.114}$$

We thus see that κ_t is completely characterized by its behavior on $\kappa_t(\cdot, 0)$, which we define as the reduced kernel $k_t : \mathbb{R}^2 \to R$. Plugging this newfound knowledge back into the integral operator axiom we get the following result.

Lemma 7.8 (Translation Equivariance implies Semifield Convolution): Consider the integral operator (Axiom 7.1) and translation equivariance (Axiom 7.5). Define the reduced kernel

$k_t(x) = \kappa_t(x, 0)$. We can write the scale-space operator Φ_t as a semifield convolution:

$$\Phi_t f = k_t \circledast f. \tag{7.115}$$

Proof: This follows immediately from Lemma 7.5.

$$\begin{aligned}
(\Phi_t f)(x) &= \oint_{y \in \mathbb{R}^2} \kappa_t(x, y) \otimes f(y) \\
&= \oint_{y \in \mathbb{R}^2} \kappa_t(x - y, 0) \otimes f(y) \\
&= \oint_{y \in \mathbb{R}^2} k_t(-y + x) \otimes f(y) \\
&= (k_t \circledast f)(x).
\end{aligned} \tag{7.116}$$

□

Lemma 7.9 (Convolution Property of Reduced Kernel): From the integral operator (Axiom 7.1), the one-parameter semigroup (Axiom 7.2), the strong continuity (Axiom 7.3), and the translation equivariance (Axiom 7.5), it follows that the reduced kernel $k_t(x) = \kappa_t(x, 0)$ satisfies:

$$k_s \circledast k_t = k_{s+t} \quad \text{for all} \quad t, s > 0. \tag{7.117}$$

Proof: We have already seen in Lemma 7.8 that the integral operator axiom and the translation equivariance axiom imply that

$$\Phi_t f = k_t \circledast f. \tag{7.118}$$

If we use this formula together with the one-parameter semigroup property axiom we get

$$k_s \circledast (k_t \circledast f) = k_{s+t} \circledast f. \tag{7.119}$$

Using associativity of semifield convolution (7.31) on the l.h.s.:

$$(k_s \circledast k_t) \circledast f = k_{s+t} \circledast f. \tag{7.120}$$

We are free to choose $f = k_\varepsilon$ for $\varepsilon > 0$:

$$(k_s \circledast k_t) \circledast k_\varepsilon = k_{s+t} \circledast k_\varepsilon. \tag{7.121}$$

From the strong continuity axiom we know that $\lim_{\varepsilon \to 0} k_\varepsilon \circledast k_t = k_t$, thus, after taking this limit, we can conclude:

$$k_s \circledast k_t = k_{s+t}. \tag{7.122}$$

□

7.5.2 Towards the Semifield Fourier Domain

We see that to move forward we need a way to efficiently work with semifield convolutions. The semifield Fourier transform (Definition 7.25) has the important property of turning convolutions into much more wieldy pointwise multiplication. This is why we translate all previous lemmas to the semifield Fourier domain using the semifield Fourier transform.

Lemma 7.10: Consider all axioms of a semifield scale-space (Definition 7.27). The reduced kernel $k_t(x) := \kappa_t(\cdot, 0)$ in the semifield Fourier domain $\hat{k}_t = \mathcal{F}_R(k_t)$ satisfies

$$\hat{k}_t(Q\omega) = \hat{k}_t(\omega), \tag{7.123}$$

$$\hat{k}_{s^\alpha t}\left(\frac{\omega}{s}\right) = \hat{k}_t(\omega), \tag{7.124}$$

$$\hat{k}_s(\omega) \otimes \hat{k}_t(\omega) = \hat{k}_{s+t}(\omega), \tag{7.125}$$

for all orthonormal $Q \in \mathbb{R}^{2\times 2}$, $\omega \in \mathbb{R}^2$, and $s, t > 0$.

Proof: Lemma 7.6 tells us that

$$\kappa_t(Qx, Qy) = \kappa_t(x, y). \tag{7.126}$$

Translating this to the reduced kernel k_t gives

$$k_t(Qx) = k_t(x). \tag{7.127}$$

Taking the semifield Fourier transform on both sides, and using the rotoreflection equivariance property of the Fourier transform (7.46), we get:

$$\hat{k}_t(Q\omega) = \hat{k}_t(\omega). \tag{7.128}$$

Lemma 7.7 also tells us that

$$\chi(s) \otimes \kappa_{s^\alpha t}(sx, sy) = \kappa_t(x, y). \tag{7.129}$$

Translating this to the reduced kernel k_t gives

$$\chi(s) \otimes k_{s^\alpha t}(sx) = k_t(x). \tag{7.130}$$

Taking the semifield Fourier transform on both sides, and using the scaling equivariance property of the Fourier transform (7.47), we get:

$$\chi(s) \otimes \chi\left(\frac{1}{s}\right) \otimes \hat{k}_{s^\alpha t}\left(\frac{\omega}{s}\right) = \hat{k}_t(\omega). \tag{7.131}$$

Using that χ is a homomorphism we know that $\chi(s) \otimes \chi\left(\frac{1}{s}\right) = \chi\left(\frac{s}{s}\right) = \chi(1) = \mathbb{1}$, so we can simplify this to

$$\hat{k}_{s^\alpha t}\left(\frac{\omega}{s}\right) = \hat{k}_t(\omega). \tag{7.132}$$

Lemma 7.9 tells us that

$$k_s \circledast k_t = k_{s+t}. \tag{7.133}$$

Taking the semifield Fourier transform on both sides, and using the convolution property of the Fourier transform (7.45), we get:

$$\hat{k}_s \otimes \hat{k}_t = \hat{k}_{s+t}. \tag{7.134}$$

□

Let us check if the reduced kernels of the scale-spaces of interest indeed satisfy the properties listed in the previous lemma by inspecting their Fourier transforms.

Proposition 7.3 (Semifield Fourier Transform of Kernels of Interest): Consider the employed semifield Fourier transforms (Definition 7.26) and the semifield scale-spaces of interest (Definition 7.28). Let $k_t(x) := \kappa_t(\cdot, 0)$ be the reduced kernel and $\hat{k}_t = \mathcal{F}_R(k_t)$ its semifield Fourier transform.

a) For the Gaussian scale-space over the linear semifield L we have:

$$\hat{k}_t^L(\omega) = \exp\left(-\frac{1}{2}t\|\omega\|^2\right). \tag{7.135}$$

b) For the quadratic root scale-spaces over the root semifields R_p we have:

$$\hat{k}_t^{R_p}(\omega) = \exp\left(-\frac{1}{2p}t\|\omega\|^2\right). \tag{7.136}$$

c) For the quadratic logarithmic scale-spaces over the logarithm semifields L_μ we have:

$$\hat{k}_t^{L_\mu}(\omega) = -\frac{1}{2\mu}t\|\omega\|^2. \tag{7.137}$$

d) For the α-dilation scale-space over the tropical max semifield T_+ we have:

$$\hat{k}_t^{T_+}(\omega) = \frac{1}{\alpha}t\|\omega\|^\alpha. \tag{7.138}$$

e) For the α-erosion scale-space over the tropical min semifield T_- we have:

$$\hat{k}_t^{T_-}(\omega) = -\frac{1}{\alpha}t\|\omega\|^\alpha. \tag{7.139}$$

7.5.3 EXPLICIT FORM OF THE REDUCED KERNEL

With the results derived above we are now ready to derive the explicit form of the reduced kernel in the Fourier domain $\hat{k}_t$, and, in turn, an expression for the reduced kernel k_t. But to succinctly state this explicit form we need one extra ingredient: semifield exponentiation, i.e. a generalization of repeated semifield multiplication.

Definition 7.30 (Semifield Exponentiation): Let $R = (R, \mathbb{0}, \mathbb{1}, \oplus, \otimes)$ be a one-dimensional metric semifield. The *semifield exponentiation* $\exp_R : \mathbb{R} \to R$ is defined as the (up to time scaling unique) map that satisfies:

$\exp_R(s) \otimes \exp_R(t) = \exp_R(s+t)$ for all $s, t \in \mathbb{R}$,
$\exp_R(0) = \mathbb{1}$,
$\lim_{t\to\infty} \exp_R(t) = \mathbb{0}$.

To distinguish the semifield exponentiation from regular exponentiation we always indicate the former with the semifield in the subscript, and the latter without any subscript.

Definition 7.31 (Employed Semifield Exponentiation):

a) In the (nonnegative) linear semifield L case we have $\exp_L(t) = \exp(-t)$.
b) In the root semifields R_p case we have $\exp_{R_p}(t) = \exp(-t)$.
c) In the logarithmic semifields L_μ case we have $\exp_{L_\mu}(t) = -\operatorname{sign}(\mu)t$.
d) In the tropical max semifield T_+ case we have $\exp_{T_+}(t) = -t$.
e) In the tropical min semifield T_- case we have $\exp_{T_-}(t) = t$.

Theorem 7.1 (Explicit form Reduced Kernel): Let R be a one-dimensional metric semifield, $\mathcal{F}_R$ the semifield Fourier transform (Definition 7.25), and $\exp_R$ the semifield exponentiation (Definition 7.30). Consider all axioms of a semifield scale-space (Definition 7.27). We have that the reduced kernel k_t is (up to a time scaling) equal to:

$$\begin{aligned} \hat{k}_t(\omega) &= \exp_R(\|\omega\|^\alpha t), \\ k_t(x) &= \left(\mathcal{F}_R^{-1}\hat{k}_t\right)(x), \end{aligned} \tag{7.140}$$

where α is the scaling power (Axiom 7.4).

Before we continue with the proof, we can check that, indeed, all semifield Fourier transforms of the reduced kernels, as listed in Proposition 7.3, have this stated form.

Proof: Consider the reduced kernel $\hat{k}_t$ in the Fourier domain and all its properties as listed in Lemma 7.10. Due to the rotoreflectional symmetry (7.125) we will abuse notation slightly and write

$$\hat{k}_t(\omega) = \hat{k}_t(\|\omega\|) = \hat{k}_t(r), \tag{7.141}$$

where $r = \|\omega\|$. Taking $s = r > 0$ in the scaling invariance (7.123) we get

$$\hat{k}_t(r) = \hat{k}_{r^\alpha t}(1) \quad \text{for} \quad r > 0. \tag{7.142}$$

Due to the one-parameter semigroup property (7.124) and $t \mapsto \hat{k}_t(\omega)$ being continuous, we have (up to a time scaling)

$$\hat{k}_t(1) = \exp_R(t) \quad \text{or} \quad \hat{k}_t(1) = \mathbb{0}. \tag{7.143}$$

We are not interested in the $\hat{k}_t(1) = \mathbb{0}$ case as it would imply, together with the previous equation, that $\hat{k}_t$ is identically zero, corresponding to a non-relevant scale-space. Combining the equations found so far we get

$$\hat{k}_t(r) = \exp_R(r^\alpha t) \quad \text{for} \quad r > 0. \tag{7.144}$$

which we can extend to $r = 0$ by continuity. Taking the inverse semifield Fourier transform concludes the proof. □

The up-to-a-time-scaling non-uniqueness is something we can *not* avoid as every semifield scale-space Φ_t corresponds to an infinite family of scale-spaces $\tilde{\Phi}_t = \Phi_{st}$ for every $s > 0$.

The above theorem shows that every (one dimensional metric) semifield R corresponds to a unique one-parameter family of semifield scale-spaces, where the scaling power α acts as the parameter.

There is one caveat here though, and that is that not every α necessarily results in a $\tilde{k}_t$ which is in the domain of the used inverse semifield Fourier transform. For example, in the tropical semifields the case $\alpha = 1$ results in a $\tilde{k}_t$ which we cannot insert into the inverse transforms listed in Definition 7.26.

7.6 ARCHITECTURE

In this section, we will briefly discuss the PDE-CNN architecture by defining the PDE sublayers corresponding to semifield scale-spaces (7.147), the PDE sublayer corresponding to convection (7.149), and the affine sublayer (7.150).

Consider any one-dimensional metric semifield R and a corresponding semifield scale-space Φ_t. As Lemma 7.8 shows, we can write the scale-space $\Phi_t f$ of a (appropriate) function $f : \mathbb{R}^2 \to R$ as $\Phi_t f = k_t \circledast f$ where $k_t : \mathbb{R}^2 \to R$ is the reduced kernel, and $\circledast$ is the semifield convolution. We want to implement these semifield scale-spaces to use them within the design of our PDE-CNNs. However, in practice, we cannot work with general signals defined on the continuum of $\mathbb{R}^2$, and we need to discretize our setting.

As is usual in machine learning we imagine images $f : \mathbb{R}^2 \to R$ as sampled images $\tilde{f} : Z \to R$ on a grid $Z \subset \mathbb{R}^2$ such that $\tilde{f}(z) = f(z)$ for all grid points $z \in Z$. We idealize the grid as the infinite integer grid $\mathbb{Z}^2$ here for the sake of simplicity (we have no boundary concerns). Let $\tilde{f}, \tilde{k} : \mathbb{Z}^2 \to R$ be the discretized versions of an image f and any kernel k. The discrete semifield convolution is defined as

$$\left(\tilde{k} \circledast \tilde{f}\right)[i,j] = \bigoplus_{(m,n)\in\mathbb{Z}^2} \tilde{k}[-m+i,-n+j] \otimes \tilde{f}[m,n], \tag{7.145}$$

where we have used the notation $[\cdot,\cdot]$ to emphasize the discrete nature.

With the discrete semifield convolution we can write down the formula for a PDE sublayer in the PDE-CNN. The input of a PDE sublayer consists of signals $\sim f_i : \mathbb{Z}^2 \to R$ and matrices $H_i \in \mathbb{R}^{2\times 2}$ with $i = 1, ..., C$, and C being the number of channels. The matrices H_i act as the learnable parameters of the layer. We consider the continuous scale-space kernel $k_t : \mathbb{R}^2 \to R$ and create the discretized kernels $\tilde{k}_i : \mathbb{Z}^2 \to R$ by defining

$$\tilde{k}_i(x) = k_1(H_i x). \tag{7.146}$$

Without loss of generality we may take $t = 1$ in our scale-space kernel k_t as a scaling in t can be captured in H_i. We then perform the discrete semifield convolutions to acquire our outputs $\sim g_i : \mathbb{Z}^2 \to R$:

$$\text{PDE sublayer:} \quad \tilde{g}_i = \tilde{k}_i \circledast \tilde{f}_i. \tag{7.147}$$

The matrices H_i require some explanation. As already touched upon in Subsection 7.4.1, in the semifield scale-space axioms we implicitly assumed the standard inner product $\mathcal{G}(x, y) = x^\top y$ on $\mathbb{R}^2$, but this is not the only one we can choose. By choosing the inner product to be

$$\mathcal{G}(x, y) = x^\top G y, \quad G = H^\top H \tag{7.148}$$

for any matrix H we get a scale-space representation that is completely identical, albeit “stretched” with respect to the coordinates. The Gram matrix $G = H^\top H$ is by construction symmetric positive definite (SPD) as required, and relieves us from coding a SPD constraint. By considering kernels of the form $k_t(Hx)$ we effectively include the possibility of processing the image with a different inner product. The “stretching” induced by the matrix H in the Gaussian scale-space case corresponds to the *affine Gaussian scale-space* studied in LINDEBERG AND GÅRDING (1997).

The PDE sublayer, as described in formula (7.147), seems to have nothing to do with a PDE at first glance. However, remember that every semifield scale-space Φ_t over a semifield R can be associated with a PDE, and the PDE sublayer is effectively a solver for the corresponding initial value problem. The examples in Definition 7.28 clarify this.

Alongside the PDE sublayers that correspond to semifield scale-spaces we also have the convection PDE sublayer. The convection sublayer effectively solves the convection PDE by translating images. The input of a convection PDE sublayer consists of images $\tilde{f}_i : \mathbb{Z}^2 \to R$ and vectors $v_i \in \mathbb{R}^2$ with $i = 1, \dots, C$, and C being the number of channels. The vectors v_i act as the learnable parameters of the layer. The output signals $\tilde{g}_i : \mathbb{Z}^2 \to R$ are obtained through bilinear interpolating (Interp) the inputs at the appropriate positions:

$$\begin{aligned}&\text{Convection sublayer:}\\&\tilde{g}_i[m,n] = \operatorname{Interp}\left(\tilde{f}_i, m - (v_i)_1, n - (v_i)_2\right).\end{aligned} \tag{7.149}$$

The final ingredient we need is the affine layer which is defined as follows. The input consists of channels $\sim f_i : \mathbb{Z}^2 \to \mathbb{R}$, weights $w_{ij} \in \mathbb{R}$, and biases $b_j \in \mathbb{R}$, with $i = 1, \dots, C_i$, $j = 1, \dots, C_o$, and C_i, C_o being the number of input and output channels respectively. The weights w_{ij} and biases b_j act as the learnable parameters of the layer. We then perform the following pointwise computation to get the C_o outputs $\sim g_j : \mathbb{Z}^2 \to R$

$$\text{Affine layer:} \quad \tilde{g}_j = b_j + \sum_{i=1}^{C_i} w_{ij} \tilde{f}_i. \tag{7.150}$$

By concatenating various PDE sublayers, that being either a sublayer that corresponds to a scale-space or a convection sublayer, with an affine layer at the end we form a PDE layer. Multiple PDE layers after each other with an affine layer at the start and end creates a PDE-CNN.

7.7 EXPERIMENTS

This section presents two experiments. The first assesses how adding additional scale-space–generating PDEs to the PDE-CNN architecture affects its accuracy. The second evaluates the architecture's data efficiency on the DRIVE dataset. The code for these experiments is available in the GitLab repository SEMIFIELD PDE-CNNS.

7.7.1 Including New PDEs in PDE-CNNs

Current PDE-based neural networks employ three PDEs that generate scale-spaces: diffusion, dilation, and erosion, which correspond, respectively, to the linear semifield L, the tropical min semifield T_-, and the tropical max semifield T_+. Our theory reveals at least two PDEs that have not yet been used in PDE-based neural networks: the PDE that generates the root scale-space (7.79) and the PDE that generates the logarithmic scale-space (7.81), which arise naturally from the root semifield R_p and logarithm semifield L_μ. Our first experiment will examine how the inclusion of these PDEs affects the accuracy of the PDE-CNN architecture.

The networks that we consider always include the convection PDE sublayer at the start of the PDE layer. The PDE-CNNs will consist of 6 PDE layers, 32 channels, and have an average parameter count of approximately 9 500, which changes with the amount of PDEs we add to its PDE layers.

We will test the networks on the DRIVE dataset (Subsection 2.8.1). We selected the DRIVE dataset because its images contain vessels at varying scales, making it well-suited for applying scale-space techniques, including PDE-based neural networks.

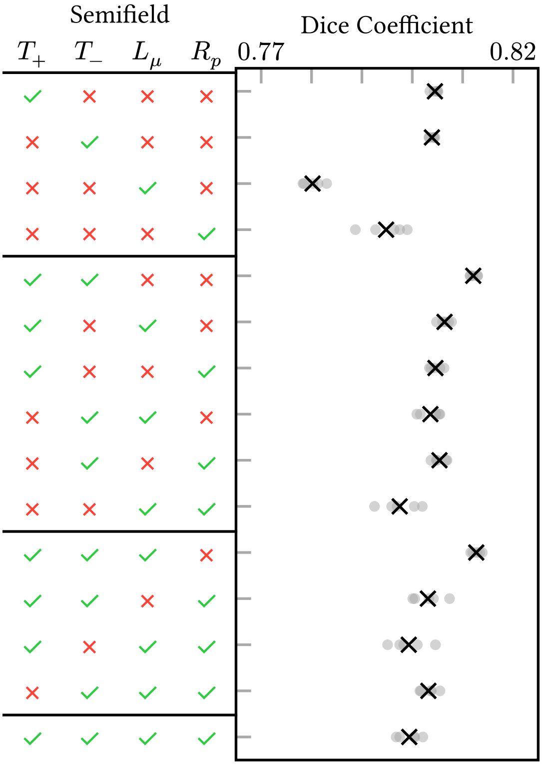
Since the goal is retinal vessel segmentation, we will be training using the Dice loss (2.59). We train on batches consisting of 8 patches. We empirically found that a higher batch size results in a worse test set accuracy. We use the AdamW optimizer with an initial learning rate of 0.01 that decays linearly to 0.001 over the first 1 000 batches. The beta, epsilon and weight decay parameters of AdamW are kept at their (PyTorch) default values of $(0.9, 0.999)$, 10^{-8}, and 0.01. During training we keep track of the Dice coefficient on the test set and the best one is stored. We train until the Dice coefficient on the test set no longer increases, which happens within 20 000 batches. We then repeat this 5 times for every possible situation.

We empirically found that adding the linear semifield L, that is we add a PDE sublayer corresponding to the Gaussian scale-space, does *not* affect the accuracy of any of the networks. This can be explained by noticing that such a PDE sublayer can be emulated completely and effectively by the convection sublayer (7.149) together with the affine sublayer (7.150) , which are always components of the PDE-CNNs we consider here. For this reason we have omitted the linear semifield PDE sublayers altogether. This is in agreement with the results found in Castella (2021), p.28.

The result can be found in Figure 7.1. We observe multiple things:

- Adding semifields to the existing PDE-CNN architecture, which only employ the tropical semifields and convection, may enhance accuracy, albeit not significantly, as observed in the case of going from $\{T_+, T_-\}$ to $\{T_+, T_-, L_\mu\}$.
- The inclusion of the tropical min semifield T_- *always* increases accuracy, most starkly seen when going from $\{L_\mu\}$ to $\{T_-, L_\mu\}$.
- Adding semifields does *not* necessarily improve accuracy, as is evident from the last row when compared to the two-semifield models in the middle rows.
- The inclusion of the root semifield R_p seems to make the training less stable, as indicated by the increase in spread within the scatter plot at the respective rows.

It is worth mentioning however that these results might be specific to the DRIVE dataset.



T_+	T_-	L_μ	R_p
✓	✗	✗	✗
✗	✓	✗	✗
✗	✗	✓	✗
✗	✗	✗	✓
✓	✓	✗	✗
✓	✗	✓	✗
✓	✗	✗	✓
✗	✓	✓	✗
✗	✓	✗	✓
✗	✗	✓	✓
✓	✓	✓	✗
✓	✓	✗	✓
✓	✗	✓	✓
✗	✓	✓	✓
✓	✓	✓	✓

Figure 7.1: A scatterplot of the accuracy of a 6-layer PDE-CNN on the DRIVE dataset, with various designs of the PDE layer as indicated in the table on the left. The crosses indicate the mean. The rows are organized according to the amount of semifields included in the model.

7.7.2 Data Efficiency of PDE-CNNs

The data efficiency of PDE-G-CNNs on $\mathbb{M}_2$ is already verified in Pai, Bellaard, Smets and Duits (2023), but whether this desirable property holds in the PDE-CNN case is still left untested. Our second experiment is therefore testing the data efficiency of a PDE-CNN on the DRIVE dataset.

As a baseline we consider a CNN with 31 488 parameters, and compare this against a PDE-CNN with 5 280 parameters. To make the comparison fair both networks have 6 layers and 24 channels. The only difference between the CNN and PDE-CNN is the kind of layer that is used. In the CNN we use a standard 2D convolutional module with 3×3 kernels together with a nonlinear activation function. In the PDE-CNN we use a PDE layer, as described in Section 7.6, that employs three PDEs: convection, dilation, and erosion. The size of the networks has been chosen this way such that both give a satisfactory Dice coefficient of $\gtrapprox 0.80$ on the test set when trained on the complete training set.

Following the method in Pai, Bellaard, Smets and Duits (2023), we randomly take 1% to 100% of the training data. Other than that our methodology is identical to Subsection 7.7.1.

The result can be found in Figure 7.2. We see that on the DRIVE dataset, in comparison with a standard CNN, the PDE-CNN not only features fewer parameters but also showcases competitive accuracy and increased data efficiency. This mirrors the results found in Pai, Bellaard, Smets and Duits (2023), but this time for a PDE-CNN instead of the $M = \mathbb{M}_2$ PDE-G-CNN considered there.

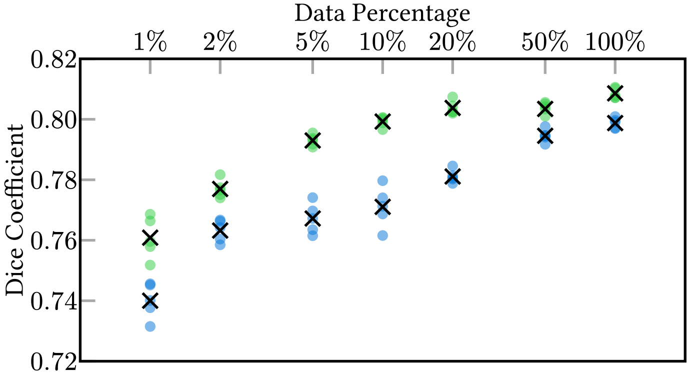


Figure 7.2: A scatterplot of the accuracy of a 6-layer 24-channel CNN (31 488 parameters) in blue and PDE-CNN (5 280 parameters) in green on the DRIVE dataset, when trained multiple times on varying amounts of training data. The crosses indicate the mean.

7.8 CONCLUSION

PDE-CNNs are an interesting alternative to CNNs in the sense that their constituents, this being solvers of PDEs that generate scale-spaces, are geometrically meaningful and interpretable.

The existing PDE-CNN framework uses four PDEs: convection, diffusion, dilation, and erosion. Through the introduction of semifield scale-spaces (Definition 7.27) we demonstrate the presence of a broad class of PDEs that remain unused within the PDE-CNN paradigm.

The theory of semifield scale-spaces is expressive and encapsulates a large class of known scale-spaces. Theorem 7.1 shows that every semifield gives rise to a one-parameter family of semifield scale-spaces. This indicates that the generalization to semifields is one that is not too general and definitely fruitful.

In Subsection 7.7.2 we empirically verified on the DRIVE dataset that PDE-CNNs, just like PDE-G-CNNs, when compared to traditional CNNs, require less training data, have fewer parameters, and increased accuracy.

In Subsection 7.7.1 we experimented on the inclusion of various semifields and their corresponding scale-spaces within PDE layers of a PDE-CNN. We see that the thought “more semifields means better accuracy” is incorrect, and that it is not clear if the addition of more semifields into the already existing PDE-CNN framework is worth the effort. However, in all cases inclusion of the tropical semifield improved the result, advocating for tropical algebras in PDE-based neural networks.

Publications

Gijs Bellaard, 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. Volume: 65. Issue: 6. Pages: 819-843. DOI: `10.1007/s10851-023-01147-w`. Cited on pages: 11, 14 and 49.

Gijs Bellaard, Gautam Pai, Javier Oliván Bescós and Remco Duits. 2023. *“Geometric Adaptations of PDE-G-CNNs”*. In: Scale Space and Variational Methods in Computer Vision. Pages: 538-550. DOI: `10.1007/978-3-031-31975-4_41`. Cited on pages: 11, 14 and 49.

Gijs Bellaard, Sei Sakata, Bart M. N. Smets and Remco Duits. 2025. *“PDE-CNNs: Axiomatic Derivations and Applications”*. In: Journal of Mathematical Imaging and Vision. Volume: 67. Issue: 2. DOI: `10.1007/s10851-025-01230-4`. Cited on pages: 11, 15, 49 and 144.

Gijs Bellaard and Bart M. N. Smets. 2025. *“Roto-Translation Invariant Metrics on Position-Orientation Space”*. In: Geometric Science of Information. Pages: 34-42. DOI: `10.1007/978-3-032-03918-7_4`. Cited on pages: 12 and 14.

Gijs Bellaard, Bart M. N. Smets and Remco Duits. 2025. *“Universal Collection of Euclidean Invariants between Pairs of Position-Orientations”*. In: Geometric Science of Information. Pages: 24-33. DOI: `10.1007/978-3-032-03918-7_3`. Cited on pages: 12 and 14.

Daan Bon, Gautam Pai, **Gijs Bellaard**, Olga Mula and Remco Duits. 2025. *“Optimal Transport on the Lie Group of Roto-translations”*. In: SIAM Journal on Imaging Sciences. Volume: 18. Issue: 2. Pages: 789-821. DOI: `10.1137/24M1641531`. Cited on pages: 13 and 15.

Remco Duits, **Gijs Bellaard** and Alice Barbora Tumpach. 2025. *“Analysis and Computation of Geodesic Distances on Reductive Homogeneous Spaces”*. In: Geometric Science of Information. Pages: 13-23. DOI: `10.1007/978-3-032-03918-7_2`. Cited on page: 15.

Gautam Pai, **Gijs Bellaard**, Rick Sengers, Luc Florack and Remco Duits. 2025. *“Entropic Optimal Transport with*

Data-Driven Metrics on the Roto-translation Group". In: Scale Space and Variational Methods in Computer Vision. DOI: `10.1007/978-3-031-92369-2_27`. Cited on pages: 13 and 15.

Gautam Pai, **Gijs Bellaard**, Bart M. N. Smets and Remco Duits. 2023. *"Functional Properties of PDE-Based Group Equivariant Convolutional Neural Networks"*. In: Geometric Science of Information. Pages: 63-72. DOI: `10.1007/978-3-031-38271-0_7`. Cited on pages: 11, 15, 49, 180 and 181.

BIBLIOGRAPHY

ANDREI AGRACHEV, DAVIDE BARILARI AND UGO BOSCAIN. 2019. *"A Comprehensive Introduction to Sub-Riemannian Geometry"*. DOI: 10.1017/9781108677325. Cited on page: 23.

LUIS ALVAREZ, FRÉDÉRIC GUICHARD, PIERRE-LOUIS LIONS AND JEAN-MICHEL MOREL. 1993. *"Axioms and fundamental equations of image processing"*. In: Archive for Rational Mechanics and Analysis. Volume: 123. Issue: 3. Pages: 199-257. DOI: 10.1007/BF00375127. Cited on page: 131.

BRANDON ANDERSON, TRUONG-SON HY AND RISI KONDOR. 2019. *"Cormorant: Covariant Molecular Neural Networks"*. In: Neural Information Processing Systems. Volume: 32. http://papers.neurips.cc/paper/2019/hash/03573b32b2746e6e8ca98b9123f2249b-Abstract.html. Cited on page: 53.

JONAS AUGUST AND STEVEN W. ZUCKER. 2003. *"Sketches with curvature: the curve indicator random field and Markov processes"*. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume: 25. Issue: 4. Pages: 387-400. DOI: 10.1109/TPAMI.2003.1190567. Cited on page: 119.

ERIK J. BEKKERS, DA CHEN AND JORG M. PORTEGIES. 2018. *"Nilpotent Approximations of Sub-Riemannian Distances for Fast Perceptual Grouping of Blood Vessels in 2D and 3D"*. In: Journal of Mathematical Imaging and Vision. Volume: 60. Issue: 6. Pages: 882-899. DOI: 10.1007/s10851-018-0787-z. Cited on pages: 66 and 67.

ERIK J. BEKKERS, 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. Volume: 8. Issue: 4. Pages: 2740-2770. DOI: 10.1137/15M1018460. Cited on pages: 12 and 59.

ERIK J. BEKKERS, SHARVAREE VADGAMA, ROB HESSELINK, PUTRI A. VAN DER LINDEN AND DAVID W. ROMERO. 2024. *"Fast, Expressive SE(3) Equivariant Networks through Weight-Sharing* *in Position-Orientation Space"*. In: International Conference on Learning Representations. `https://openreview.net/forum?id=dPHLbUqGbr`. Cited on pages: 12, 53, 102, 108 and 116.

Nicky J. van den Berg, 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. Volume: 66. Issue: 2. Pages: 198-230. DOI: `10.1007/s10851-023-01170-x`. Cited on page: 12.

Jonathan M. Borwein and Adrian S. Lewis. 2006. *"Convex Analysis and Nonlinear Optimization"*. DOI: `10.1007/978-0-387-31256-9`. Cited on page: 153.

Roger W. Brockett and Petros Maragos. 1992. *"Evolution equations for continuous-scale morphology"*. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. Volume: 3. Pages: 125-128. DOI: `10.1109/ICASSP.1992.226260`. Cited on pages: 42 and 131.

Michael M. Bronstein, Joan Bruna, Taco Cohen and Petar Veličković. 2021. *"Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges"*. arXiv: `2104.13478`. Cited on page: 11.

Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam and Pierre Vandergheynst. 2017. *"Geometric Deep Learning: Going beyond Euclidean data"*. In: IEEE Signal Processing Magazine. Volume: 34. Issue: 4. Pages: 18–42. DOI: `10.1109/MSP.2017.2693418`. Cited on page: 10.

Joan Bruna and Stephane Mallat. 2013. *"Invariant Scattering Convolution Networks"*. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume: 35. Issue: 8. Pages: 1872-1886. DOI: `10.1109/TPAMI.2012.230`. Cited on page: 11.

Bernhard Burgeth and Joachim Weickert. 2005. *"An Explanation for the Logarithmic Connection between Linear and Morphological System Theory"*. In: International Journal of Computer Vision. Volume: 64. Issue: 2. Pages: 157-169. DOI: `10.1007/s11263-005-1841-z`. Cited on page: 131.

Adrian Castella. 2021. *"Introduction to Shift-Invariant PDE-based Convolutional Neural Networks and experimentation with Vessel Segmentation"*. https://research.tue.nl/en/studentTheses/introduction-to-shift-invariant-pde-based-convolutional-neural-ne. Cited on page: 179.

Fernando Cervantes-Sanchez, Ivan Cruz-Aceves, Arturo Hernandez-Aguirre, Martha A. Hernandez-Gonzalez and Sergio E. Solorio-Meza. 2019. *"Automatic Segmentation of Coronary Arteries in X-ray Angiograms Using Multiscale Analysis and Artificial Neural Networks"*. In: Applied Sciences. Volume: 9. Issue: 24. DOI: 10.3390/app9245507. Cited on page: 55.

Antonin Chambolle and Thomas Pock. 2019. *"Total Roto-Translational Variation"*. In: Numerische Mathematik. Volume: 142. Issue: 3. Pages: 611-666. DOI: 10.1007/s00211-019-01026-w. Cited on page: 13.

François Chollet. 2017. *"Xception: Deep Learning with Depthwise Separable Convolutions"*. arXiv: 1610.02357. Cited on page: 48.

Taco S. Cohen. 2021. *"Equivariant Convolutional Networks"*. https://hdl.handle.net/11245.1/0f7014ae-ee94-430e-a5d8-37d03d8d10e6. Cited on pages: 10 and 47.

Taco S. Cohen, Mario Geiger and Maurice Weiler. 2019. *"A General Theory of Equivariant CNNs on Homogeneous Spaces"*. In: Advances in Neural Information Processing Systems. Volume: 32. https://papers.neurips.cc/paper_files/paper/2019/hash/b9cfe8b6042cf759dc4c0cccb27a6737-Abstract.html. Cited on page: 47.

Taco S. Cohen, Maurice Weiler, Berkay Kicanaoglu and Max Welling. 2019. *"Gauge Equivariant Convolutional Networks and the Icosahedral CNN"*. In: Proceedings of the 36th International Conference on Machine Learning. Volume: 97. Pages: 1321-1330. https://proceedings.mlr.press/v97/cohen19d.html. Cited on pages: 10, 52 and 155.

Taco S. Cohen and Max Welling. 2016. *"Group Equivariant Convolutional Networks"*. In: International Conference on Machine Learning. Volume: 48. Pages: 2990-2999. `https://proceedings.mlr.press/v48/cohenc16.html`. Cited on pages: 10 and 47.

Taco S. Cohen and Max Welling. 2017. *"Steerable CNNs"*. In: International Conference on Learning Representations. `https://openreview.net/forum?id=rJQKYt5ll`. Cited on page: 11.

Leo Dorst and Rein van den Boomgaard. 1994. *"Morphological signal processing and the slope transform"*. In: Signal Processing. Volume: 38. Issue: 1. Pages: 79-98. DOI: `10.1016/0165-1684(94)90058-2`. Cited on page: 131.

Remco Duits. 2005. *"Perceptual Organization in Image Analysis: A Mathematical Approach Based on Scale, Orientation and Curvature"*. DOI: `10.6100/IR595308`. Cited on pages: 38 and 40.

Remco Duits, Ugo Boscain, Francesco Rossi and Yuri L. Sachkov. 2014. *"Association Fields via Cuspless Sub-Riemannian Geodesics in SE(2)"*. In: Journal of Mathematical Imaging and Vision. Volume: 49. Issue: 2. Pages: 384-417. DOI: `10.1007/s10851-013-0475-y`. Cited on page: 72.

Remco Duits, 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. Volume: 17. Issue: 1. Pages: 42-75. DOI: `10.1134/S1054661807010063`. Cited on page: 38.

Remco Duits, Michael Felsberg, Gösta Granlund and Bart ter Haar Romeny. 2007. *"Image Analysis and Reconstruction using a Wavelet Transform Constructed from a Reducible Representation of the Euclidean Motion Group"*. In: International Journal of Computer Vision. Volume: 72. Issue: 1. Pages: 79-102. DOI: `10.1007/s11263-006-8894-5`. Cited on page: 38.

Remco Duits, Luc Florack, Jan de Graaf and Bart ter Haar Romenij. 2004. *"On the Axioms of Scale Space Theory"*. In: Journal of Mathematical Imaging and Vision. Volume: 20. Issue:

3. Pages: 267-298. DOI: `10.1023/B:JMIV.0000024043.96722.aa`. Cited on pages: 130.

Remco Duits and Erik Franken. 2010. *"Left-Invariant Parabolic Evolutions on SE(2) and Contour Enhancement via Invertible Orientation Scores Part I: Linear Left-Invariant Diffusion Equations on SE(2)"*. In: Quarterly of Applied Mathematics. Volume: 68. Issue: 2. Pages: 255-292. `http://www.jstor.org/stable/43638929`. Cited on pages: 13 and 66.

Remco Duits and Erik Franken. 2011. *"Left-Invariant Diffusions on the Space of Positions and Orientations and their Application to Crossing-Preserving Smoothing of HARDI images"*. In: International Journal of Computer Vision. Volume: 92. Issue: 3. Pages: 231-264. DOI: `10.1007/s11263-010-0332-z`. Cited on pages: 13 and 93.

Remco Duits, Stephan P. L. Meesters, Jean-Marie 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. Volume: 60. Issue: 6. Pages: 816-848. DOI: `10.1007/s10851-018-0795-z`. Cited on pages: 12 and 59.

Remco Duits, Bart M. N. Smets, Erik Bekkers and Jacobus W. Portegies. 2021. *"Equivariant Deep Learning via Morphological and Linear Scale Space PDEs on the Space of Positions and Orientations"*. In: Scale Space and Variational Methods in Computer Vision. Pages: 27-39. DOI: `10.1007/978-3-030-75549-2_3`. Cited on page: 11.

Remco Duits, Etienne St-Onge, Jacobus W. Portegies and Bart M. N. Smets. 2019. *"Total Variation and Mean Curvature PDEs on the Space of Positions and Orientations"*. In: Scale Space and Variational Methods in Computer Vision. Pages: 211-223. DOI: `10.1007/978-3-030-22368-7_17`. Cited on pages: 13 and 93.

A. F. M ter Elst and Derek W. Robinson. 1998. *"Weighted Subcoercive Operators on Lie Groups"*. In: Journal of Functional Analysis. Volume: 157. Issue: 1. Pages: 88-163. DOI: `10.1006/jfan.1998.3259`. Cited on page: 66.

Lawrence C. Evans. 2010. *“Partial Differential Equations 2nd Edition”*. Volume: 19. Cited on pages: 43, 131 and 164.

David J. Field, Anthony Hayes and Robert F. Hess. 1993. *“Contour integration by the human visual system: Evidence for a local “association field””*. In: Vision Research. Volume: 33. Issue: 2. Pages: 173-193. DOI: `10.1016/0042-6989(93)90156-Q`. Cited on page: 56.

Luc Florack. 2001. *“Non-Linear Scale-Spaces Isomorphic to the Linear Case with Applications to Scalar, Vector and Multispectral Images”*. In: International Journal of Computer Vision. Volume: 42. Issue: 1. Pages: 39-53. DOI: `10.1023/A:1011185417206`. Cited on pages: 131 and 163.

William T. Freeman and Edward H. Adelson. 1991. *“The design and use of steerable filters”*. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume: 13. Issue: 9. Pages: 891-906. DOI: `10.1109/34.93808`. Cited on page: 11.

Fabian Fuchs, Daniel Worrall, Volker Fischer and Max Welling. 2020. *“SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks”*. In: Neural Information Processing Systems. Volume: 33. Pages: 1970-1981. `https://proceedings.neurips.cc/paper/2020/hash/15231a7ce4ba789d13b722cc5c955834-Abstract.html`. Cited on page: 53.

Johannes Gasteiger, Florian Becker and Stephan Günnemann. 2021. *“GemNet: Universal Directional Graph Neural Networks for Molecules”*. In: Neural Information Processing Systems. `https://openreview.net/forum?id=HS_sOaxS9K-`. Cited on page: 41.

Frédéric Guichard and Jean-Michel Morel. 1997. *“Partial differential equations and image iterative filtering”*. In: The State of the Art in Numerical Analysis. Volume: 63. Pages: 252-562. DOI: `10.1093/oso/9780198500148.003.0020`. Cited on page: 131.

Brian C. Hall. 2015. *“Lie Groups, Lie Algebras, and Representations: An Elementary Introduction”*. DOI: `10.1007/978-3-319-13467-3`. Cited on page: 23.

William B. Heard. 2005. *"Rigid Body Mechanics: Mathematics, Physics and Applications"*. DOI: `10.1002/9783527618811`. Cited on page: 100.

Henk J.A.M. Heijmans and Rein van den Boomgaard. 2002. *"Algebraic Framework for Linear and Morphological Scale-Spaces"*. In: Journal of Visual Communication and Image Representation. Volume: 13. Issue: 1. Pages: 269-301. DOI: `10.1006/jvci.2001.0480`. Cited on pages: 131.

Taizo Iijima. 1959. *"Basic theory of pattern observation"*. In: Papers of Technical Group on Automata and Automatic Control. Cited on pages: 42 and 130.

Jörn-Henrik Jacobsen, Jan van Gemert, Zhongyou Lou and Arnold W. M. Smeulders. 2016. *"Structured Receptive Fields in CNNs"*. In: IEEE Conference on Computer Vision and Pattern Recognition. Pages: 2610-2619. DOI: `10.1109/CVPR.2016.286`. Cited on pages: 11 and 132.

Michiel H. J. Janssen, Remco Duits and Marcel Breeuwer. 2015. *"Invertible Orientation Scores of 3D Images"*. In: Scale Space and Variational Methods in Computer Vision. Pages: 563-575. DOI: `10.1007/978-3-319-18461-6_45`. Cited on page: 39.

Michiel H. J. Janssen, Augustus J. E. M. Janssen, Erik J. Bekkers, Javier Oliván Bescós and Remco Duits. 2018. *"Design and Processing of Invertible Orientation Scores of 3D Images"*. In: Journal of Mathematical Imaging and Vision. Volume: 60. Issue: 9. Pages: 1427-1458. DOI: `10.1007/s10851-018-0806-0`. Cited on page: 39.

Stiliyan N. Kalitzin, Bart M. ter Haar Romeny and Max A. Viergever. 1997. *"Invertible orientation bundles on 2D scalar images"*. In: Scale-Space Theory in Computer Vision. Pages: 77-88. DOI: `10.1007/3-540-63167-4_41`. Cited on page: 38.

Stiliyan N. Kalitzin, Bart M. ter Haar Romeny and Max A. Viergever. 1999. *"Invertible apertured orientation filters in image analysis."*. In: International Journal of Computer Vision. Volume: 31. Pages: 145-158. DOI: `10.1023/A:1008013815039`. Cited on page: 38.

Jan J. Koenderink. 1984. *"The structure of images"*. In: Biological Cybernetics. Volume: 50. Issue: 5. Pages: 363-370. DOI: `10.1007/BF00336961`. Cited on pages: 11, 42, 130 and 157.

Vassili N. Kolokoltsov and Victor P. Maslov. 1997. *"Idempotent Analysis and Its Applications"*. DOI: `10.1007/978-94-015-8901-7`. Cited on page: 157.

Tim B. Kraakman. 2023. *"On the Construction of Classical and Quantum PDE-Based Neural Networks"*. `https://research.tue.nl/en/studentTheses/on-the-construction-of-classical-and-quantum-pde-based-neural-net`. Cited on page: 162.

Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. 2012. *"ImageNet Classification with Deep Convolutional Neural Networks"*. In: Advances in Neural Information Processing Systems. Volume: 25. Pages: . `https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html`. Cited on page: 9.

John M. Lee. 2012. *"Introduction to Smooth Manifolds"*. DOI: `10.1007/978-1-4419-9982-5`. Cited on pages: 18, 23, 110 and 111.

John M. Lee. 2019. *"Introduction to Riemannian Manifolds"*. DOI: `10.1007/978-3-319-91755-9`. Cited on page: 21.

Tony Lindeberg. 1997. *"On the Axiomatic Foundations of Linear Scale-Space"*. In: Gaussian Scale-Space Theory. Pages: 75-97. DOI: `10.1007/978-94-015-8802-7_6`. Cited on page: 157.

Tony Lindeberg. 2022. *"Scale-Covariant and Scale-Invariant Gaussian Derivative Networks"*. In: Journal of Mathematical Imaging and Vision. Volume: 64. Issue: 3. Pages: 223-242. DOI: `10.1007/s10851-021-01057-9`. Cited on pages: 132 and 154.

Tony Lindeberg and Jonas Gårding. 1997. *"Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure"*. In: Image and Vision Computing. Volume: 15. Issue: 6. Pages: 415-434. DOI: `10.1016/S0262-8856(97)01144-X`. Cited on page: 176.

LI LIU, MINGZHU WANG, SHUWANG ZHOU, MINGLEI SHU, LAURENT D. COHEN AND DA CHEN. 2023. *"Curvilinear Structure Tracking Based on Dynamic Curvature-penalized Geodesics"*. In: Pattern Recognition. Volume: 134. DOI: `10.1016/j.patcog.2022.109079`. Cited on page: 120.

SHANGZHEN LUAN, CHEN CHEN, BAOCHANG ZHANG, JUNGONG HAN AND JIANZHUANG LIU. 2018. *"Gabor Convolutional Networks"*. In: IEEE Transactions on Image Processing. Volume: 27. Issue: 9. Pages: 4357-4366. DOI: `10.1109/TIP.2018.2835143`. Cited on page: 132.

STÉPHANE MALLAT. 2012. *"Group Invariant Scattering"*. arXiv: `1101.2286`. Cited on page: 11.

DANIEL S. MARGULIES, JOACHIM BÖTTGER, AIMI WATANABE AND KRZYSZTOF J. GORGOLEWSKI. 2013. *"Visualizing the human connectome"*. In: NeuroImage. Volume: 80. Pages: 445-461. DOI: `10.1016/j.neuroimage.2013.04.111`. Cited on pages: 13 and 205.

MATEMATLETA. 2018. *"a proof that the pointswise limits of lower semicontinuous (lsc) functions is lsc"*. `https://math.stackexchange.com/q/2968282`. Cited on page: 141.

MIRGAHNEY MOHAMED, GABRIELE CESA, TACO S. COHEN AND MAX WELLING. 2020. *"A Data and Compute Efficient Design for Limited-Resources Deep Learning"*. arXiv: `2004.09691`. Cited on pages: 10, 52 and 155.

IGOR MOISEEV AND YURI L. SACHKOV. 2010. *"Maxwell Strata in Sub-Riemannian Problem on the Group of Motions of a Plane"*. In: ESAIM: Control, Optimisation and Calculus of Variations. Volume: 16. Issue: 2. Pages: 380-399. DOI: `10.1051/cocv/2009004`. Cited on pages: 72 and 76.

PHILIPPOS MORDOHAI AND GÉRARD MEDIONI. 2006. *"Tensor Voting: A Perceptual Organization Approach to Computer Vision and Machine Learning"*. DOI: `10.1007/978-3-031-02242-5`. Cited on page: 75.

ERIC J. PAUWELS, LUC J. VAN GOOL, PETER FIDDELAERS AND THEO MOONS. 1995. *"An extended class of scale-invariant and recursive* *scale space filters"*. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume: 17. Issue: 7. Pages: 691-701. DOI: 10.1109/34.391411. Cited on pages: 130 and 154.

Gabriel Peyré and Marco Cuturi. 2020. *"Computational Optimal Transport"*. arXiv: 1803.00567. Cited on page: 13.

Silvia L. Pintea, Nergis Tomen, Stanley F. Goes, Marco Loog and Jan C. van Gemert. 2021. *"Resolution Learning in Deep Convolutional Networks Using Scale-Space Theory"*. In: IEEE Transactions on Image Processing. Volume: 30. Pages: 8342-8353. DOI: 10.1109/TIP.2021.3115001. Cited on page: 132.

Jorg M. Portegies. 2018. *"PDEs on the Lie group SE(3) and their application in diffusion-weighted MRI"*. https://research.tue.nl/en/publications/pdes-on-the-lie-group-se3-and-their-application-in-diffusion-weig. Cited on pages: 13 and 93.

Jorg M. Portegies, Rutger H. J. 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. Volume: 10. Issue: 10. DOI: 10.1371/journal.pone.0138122. Cited on pages: 13 and 93.

Jorg M. Portegies, Gonzalo R. Sanguinetti, Stephan P. L. Meesters and Remco Duits. 2015. *"New Approximation of a Scale Space Kernel on SE(3) and Applications in Neuroimaging"*. In: Scale Space and Variational Methods in Computer Vision. Pages: 40-52. DOI: 10.1007/978-3-319-18461-6_4. Cited on pages: 13, 93 and 99.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp and O. Anatole von Lilienfeld. 2014. *"Quantum chemistry structures and properties of 134 kilo molecules"*. In: Scientific Data. Volume: 1. Issue: 1. DOI: 10.1038/sdata.2014.22. Cited on page: 57.

Bart M. ter Haar Romeny. 2003. *"Front-End Vision and Multi-Scale Image Analysis"*. DOI: 10.1007/978-1-4020-8840-7. Cited on page: 42.

David W. Romero, Robert-Jan Bruintjes, Jakub M. Tomczak, Erik J. Bekkers, Mark Hoogendoorn and Jan C. van Gemert. 2022. *"FlexConv: Continuous Kernel Convolutions with Differentiable Kernel Sizes"*. In: 10th International Conference on Learning Representations. `https://openreview.net/forum?id=3jooF27-0Wy`. Cited on page: 132.

Olaf Ronneberger, Philipp Fischer and Thomas Brox. 2015. *"U-Net: Convolutional Networks for Biomedical Image Segmentation"*. In: Medical Image Computing and Computer-Assisted Intervention. Pages: 234-241. DOI: `10.1007/978-3-319-24574-4_28`. Cited on page: 15.

Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum and Jean-Louis Reymond. 2012. *"Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17"*. In: Journal of Chemical Information and Modeling. Volume: 52. Issue: 11. Pages: 2864-2875. DOI: `10.1021/ci300415d`. Cited on page: 57.

Nikhil Saldanha, Silvia L. Pintea, Jan C. van Gemert and Nergis Tomen. 2021. *"Frequency learning for structured CNN filters with Gaussian fractional derivatives"*. arXiv: `2111.06660`. Cited on page: 132.

Mateus Sangalli, Samy Blusseau, Santiago Velasco-Forero and Jesús Angulo. 2021. *"Scale Equivariant Neural Networks with Morphological Scale-Spaces"*. In: Discrete Geometry and Mathematical Morphology. Pages: 483-495. DOI: `10.1007/978-3-030-76657-3_35`. Cited on pages: 132 and 154.

Filippo Santambrogio. 2015. *"Optimal Transport for Applied Mathematicians"*. DOI: `10.1007/978-3-319-20828-2`. Cited on page: 13.

Guillermo Sapiro. 2001. *"Geometric Partial Differential Equations and Image Analysis"*. DOI: `10.1017/CBO9780511626319`. Cited on page: 131.

saz. 2014. *“Measurability of a pointwise limit of measurable functions”*. `https://math.stackexchange.com/q/921576`. Cited on page: 139.

Martin Schmidt and Joachim Weickert. 2016. *“Morphological Counterparts of Linear Shift-Invariant Scale-Spaces”*. In: Journal of Mathematical Imaging and Vision. Volume: 56. Issue: 2. Pages: 352-366. DOI: `10.1007/s10851-016-0646-8`. Cited on pages: 130, 131 and 153.

Finn M. Sherry, Chase van de Geijn, Erik J. Bekkers and Remco Duits. 2025. *“Orientation Scores should be a Piece of Cake”*. In: Geometric Science of Information. Pages: 224–233. DOI: `10.1007/978-3-032-03924-8_23`. Cited on page: 39.

Laurent Sifre and Stéphane Mallat. 2013. *“Rotation, Scaling and Deformation Invariant Scattering for Texture Discrimination”*. In: IEEE Conference on Computer Vision and Pattern Recognition. Pages: 1233-1240. DOI: `10.1109/CVPR.2013.163`. Cited on page: 11.

Bart M. N. Smets, Jacobus W. Portegies, Erik J. Bekkers and Remco Duits. 2023. *“PDE-Based Group Equivariant Convolutional Neural Networks”*. In: Journal of Mathematical Imaging and Vision. Volume: 65. Issue: 1. Pages: 209-239. DOI: `10.1007/s10851-022-01114-x`. Cited on pages: 11, 49, 60, 63, 64, 70, 87, 93, 97 and 119.

Bart M. N. Smets, Jacobus W. 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. Volume: 63. Issue: 2. Pages: 237-262. DOI: `10.1007/s10851-020-00991-4`. Cited on pages: 13, 93 and 119.

Ivan Sosnovik, Michał Szmaja and Arnold Smeulders. 2020. *“Scale-Equivariant Steerable Networks”*. In: International Conference on Learning Representations. `https://openreview.net/forum?id=HJgpugrKPS`. Cited on pages: 132 and 154.

Joes Staal, 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. Volume: 23. Issue: 4. Pages: 501-509. DOI: `10.1109/TMI.2004.825627`. Cited on page: 54.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff and Patrick Riley. 2018. *"Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds"*. arXiv: `1802.08219`. Cited on page: 53.

Nergis Tomen, Silvia-Laura Pintea and Jan van Gemert. 2021. *"Deep Continuous Networks"*. In: Proceedings of the 38th International Conference on Machine Learning. Volume: 139. Pages: 10324-10335. `https://proceedings.mlr.press/v139/tomen21a.html`. Cited on page: 132.

David S. Tuch, Timothy G. Reese, Mette R. Wiegell, Nikos Makris, John W. Belliveau and J. Van Wedeen. 2002. *"High angular resolution diffusion imaging reveals intravoxel white matter fiber heterogeneity"*. In: Magnetic Resonance in Medicine. Volume: 48. Issue: 4. Pages: 577-582. DOI: `10.1002/mrm.10268`. Cited on pages: 13 and 205.

Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, Sameer Velankar, Gerard J. Kleywegt, Alex Bateman, Richard Evans, Alexander Pritzel, Michael Figurnov, Olaf Ronneberger, Russ Bates, Simon A. A. Kohl, Anna Potapenko, Andrew J. Ballard, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Ellen Clancy, David Reiman, Stig Petersen, Andrew W. Senior, Koray Kavukcuoglu, Ewan Birney, Pushmeet Kohli, John Jumper and Demis Hassabis. 2021. *"Highly accurate protein structure prediction for the human proteome"*. In: Nature. Volume: 596. Issue: 7873. Pages: 590-596. DOI: `10.1038/s41586-021-03828-1`. Cited on page: 9.

Ashish Vaswani, 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. Volume: 30. https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html. Cited on page: 9.

Joachim Weickert, Seiji Ishikawa and Atsushi Imiya. 1999. *“Linear Scale-Space has First been Proposed in Japan”*. In: Journal of Mathematical Imaging and Vision. Volume: 10. Issue: 3. Pages: 237-252. DOI: 10.1023/A:1008344623873. Cited on pages: 42 and 130.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma and Taco S. Cohen. 2018. *“3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data”*. In: Neural Information Processing Systems. Volume: 31. https://papers.nips.cc/paper/2018/hash/488e4104520c6aab692863cc1dba45af-Abstract.html. Cited on page: 53.

Martin Welk. 2003. *“Families of Generalised Morphological Scale Spaces”*. In: Scale Space Methods in Computer Vision. Pages: 770-784. DOI: 10.1007/3-540-44935-3_54. Cited on page: 130.

Andrew P. Witkin. 1983. *“Scale-space filtering”*. In: 8th International Joint Conference on Artificial Intelligence. Pages: 1019-1022. https://www.ijcai.org/Proceedings/83-2/Papers/091.pdf. Cited on pages: 11, 42 and 130.

Daniel Worrall and Max Welling. 2019. *“Deep Scale-spaces: Equivariance Over Scale”*. In: Advances in Neural Information Processing Systems. Volume: 32. https://papers.nips.cc/paper/2019/hash/f04cd7399b2b0128970efb6d20b5c551-Abstract.html. Cited on pages: 11, 132 and 154.

Jiong Zhang, Behdad Dashtbozorg, Erik J. Bekkers, Josien P. W. Pluim, Remco Duits and Bart M. ter Haar Romeny. 2016. *“Robust Retinal Vessel Segmentation via Locally Adaptive Derivative Frames in Orientation Scores”*. In: IEEE Transactions on Medical Imaging. Volume: 35. Issue: 12. Pages: 2631-2644. DOI: 10.1109/TMI.2016.2587062. Cited on page: 120.

Repositories

Erik Bekkers. 2023. *“✨🐴🔥 PONITA”*. Contributors: David Wessels. https://github.com/ebekkers/ponita. Cited on page: 53.

Gijs Bellaard. 2024. *“Semifield PDE-CNNs”*. https://gitlab.com/gijsbel/semifield-pde-cnns/. Cited on page: 177.

Gijs Bellaard. 2025. *“PONITA Invariants”*. Contributors: Bart M. N. Smets. https://gitlab.com/gijsbel/ponita_invariants. Cited on pages: 102 and 116.

Bart M. N. Smets. 2020. *“LieTorch”*. Contributors: Gijs Bellaard, Gautam Pai and Finn Sherry. https://gitlab.com/bsmetsjr/lietorch. Cited on pages: 53, 85, 119 and 125.

INDEX

FIGURES

TABLES

Acknowledgements

First and foremost I have to thank Remco Duits for being my supervisor these past four years. Without you I would not have had the opportunity to pursue a PhD, and this thesis would not have existed.

Being at the university has been a joy due to my immediate colleagues: Bart Smets, Finn Sherry, Gautam Pai, Lars Smolders, Leanne Vis, Nicky van den Berg, Rick Sengers, and Sjors Heefer. I enjoyed all our activities, conversations, meetings, conferences, collaborations, breaks, lunches, walks, and, of course, drinks.

Lucy Verberk. Eight years we have already been together. That time includes nearly all of our Bachelor's, and all of our Master's and PhD. It's impossible to separate those years from you. Thank you for being there throughout.

Thanks to all my friends and family for everything we have shared during my PhD: vacations, trips, friends' weekends, festivals, Carnaval, diners, poker, *borrels*, parties, bouldering, Dungeons & Dragons sessions, and countless other moments. Here's to many more to come!

Many thanks to my committee members prof. dr. Christoph Brune, prof. dr. Victorita Dolean-Maini, prof. dr. Simon F. Portegies Zwart, and dr. Erik J. Bekkers for reading this thesis and participating in the defense.

About the Author

Gijs Bellaard was born on December 4, 1997, in Geertruidenberg. He completed his secondary education at Dongemond College Raamsdonksveer in 2016. He obtained a bachelor's degree (2019) and master's degree (2021) in Applied Mathematics from Eindhoven University of Technology (TU/e). In 2021, he started his PhD at TU/e under the supervision of Remco Duits.