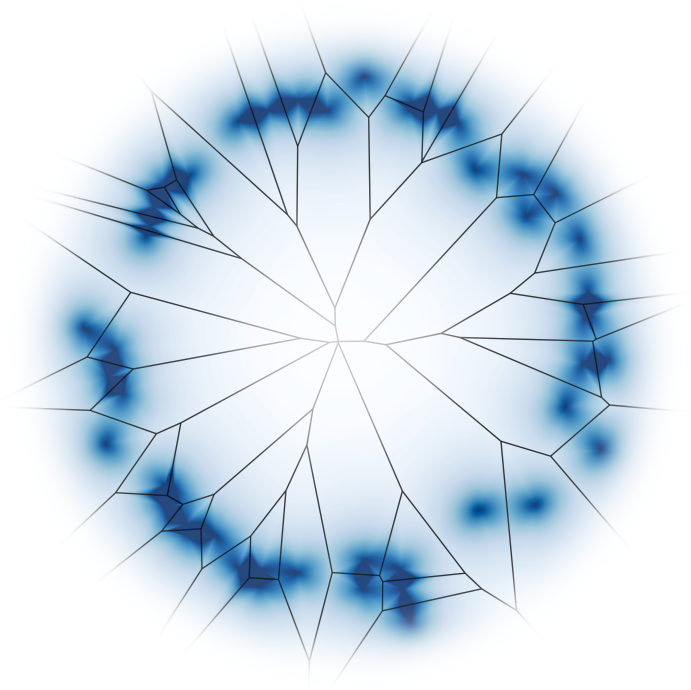


On Symmetries and Metrics in Geometric Inference

GIOVANNI LUCA MARCHETTI

Doctoral Thesis
Stockholm, Sweden 2023

Division of Robotics, Perception and Learning
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
SE-100 44 Stockholm, Sweden

Public defense:
April 9th, 2024
F3 (Flodis)

Tryck: Universitetsservice US AB

Abstract

Spaces of data naturally carry intrinsic geometry. Statistics and machine learning can leverage on this rich structure in order to achieve efficiency and semantic generalization. Extracting geometry from data is therefore a fundamental challenge which by itself defines a statistical, computational and unsupervised learning problem. To this end, symmetries and metrics are two fundamental objects which are ubiquitous in continuous and discrete geometry. Both are suitable for data-driven approaches since symmetries arise as interactions and are thus collectable in practice while metrics can be induced locally from the ambient space. In this thesis, we address the question of extracting geometry from data by leveraging on symmetries and metrics. Additionally, we explore methods for statistical inference exploiting the extracted geometric structure. On the metric side, we focus on Voronoi tessellations and Delaunay triangulations, which are classical tools in computational geometry. Based on them, we propose novel non-parametric methods for machine learning and statistics, focusing on theoretical and computational aspects. These methods include an active version of the nearest neighbor regressor as well as two high-dimensional density estimators. All of them possess convergence guarantees due to the adaptiveness of Voronoi cells. On the symmetry side, we focus on representation learning in the context of data acted upon by a group. Specifically, we propose a method for learning equivariant representations which are guaranteed to be isomorphic to the data space, even in the presence of symmetries stabilizing data. We additionally explore applications of such representations in a robotics context, where symmetries correspond to actions performed by an agent. Lastly, we provide a theoretical analysis of invariant neural networks and show how the group-theoretical Fourier transform emerges in their weights. This addresses the problem of symmetry discovery in a self-supervised manner.

Sammanfattning

Datamängder innehar en naturlig inneboende geometri. Statistik och maskininlärning kan dra nytta av denna rika struktur för att uppnå effektivitet och semantisk generalisering. Att extrahera geometri ifrån data är därför en grundläggande utmaning som i sig definierar ett statistiskt, beräkningsmässigt och oövervakat inlärningsproblem. För detta ändamål är symmetrier och metriker två grundläggande objekt som är allestädes närvarande i kontinuerlig och diskret geometri. Båda är lämpliga för datadrivna tillvägagångssätt eftersom symmetrier uppstår som interaktioner och är därmed i praktiken samlingsbara medan metriker kan induceras lokalt ifrån det omgivande rummet. I denna avhandling adresserar vi frågan om att extrahera geometri ifrån data genom att utnyttja symmetrier och metriker. Dessutom utforskar vi metoder för statistisk inferens som utnyttjar den extraherade geometriska strukturen. På den metriska sidan fokuserar vi på Voronoi-tessellationer och Delaunay-trianguleringar, som är klassiska verktyg inom beräkningsgeometri. Baserat på dem föreslår vi nya icke-parametriska metoder för maskininlärning och statistik, med fokus på teoretiska och beräkningsmässiga aspekter. Dessa metoder inkluderar en aktiv version av närmaste grann-regressorn samt två högdimensionella täthetsskattare. Alla dessa besitter konvergensgarantier på grund av Voronoi-cellernas anpassningsbarhet. På symmetrisidan fokuserar vi på representationsinlärning i sammanhanget av data som påverkas av en grupp. Specifikt föreslår vi en metod för att lära sig ekvivarianta representationer som garanteras vara isomorfa till datarummet, även i närvaro av symmetrier som stabiliserar data. Vi utforskar även tillämpningar av sådana representationer i ett robotiksammanhang, där symmetrier motsvarar handlingar utförda av en agent. Slutligen tillhandahåller vi en teoretisk analys av invarianta neuronnät och visar hur den gruppteoretiska Fouriertransformen framträder i deras vikter. Detta adresserar problemet med att upptäcka symmetrier på ett självövervakat sätt.

Acknowledgements

This thesis is a by-product of an adventurous journey. I wish to express my gratitude to the many people who have played an essential role.

I am grateful to my supervisors, whose support and advice have guided me through the tangles of research:

- Thank you, Danica, for the vision, the freedom, and all the help. You kept this journey focused, converting challenges into opportunities. This has resulted in personal growth and fulfillment beyond any expectation.

- Thank you, Anastasiia, for being a constant mentor, walking me across a challenging field.

I am grateful to my collaborators, whose expertise and enthusiasm have represented a morning star to me:

- Thank you, Gustaf, for introducing me to the fundamentals. You made the first steps in this journey feel like a breeze.

- Thank you, Vladislav, for always prioritizing research that is pure and authentic, despite the compromises.

- Thank you, Alfredo, for your kaleidoscopic knowledge and for your commitment. This journey has been grounded in what I have learned from you.

- Thank you, Florian, Luis, Alex, Chris, and Sophia, for the exciting research together. Your bright insights have lit the path all the way.

I am grateful to the places I have traversed:

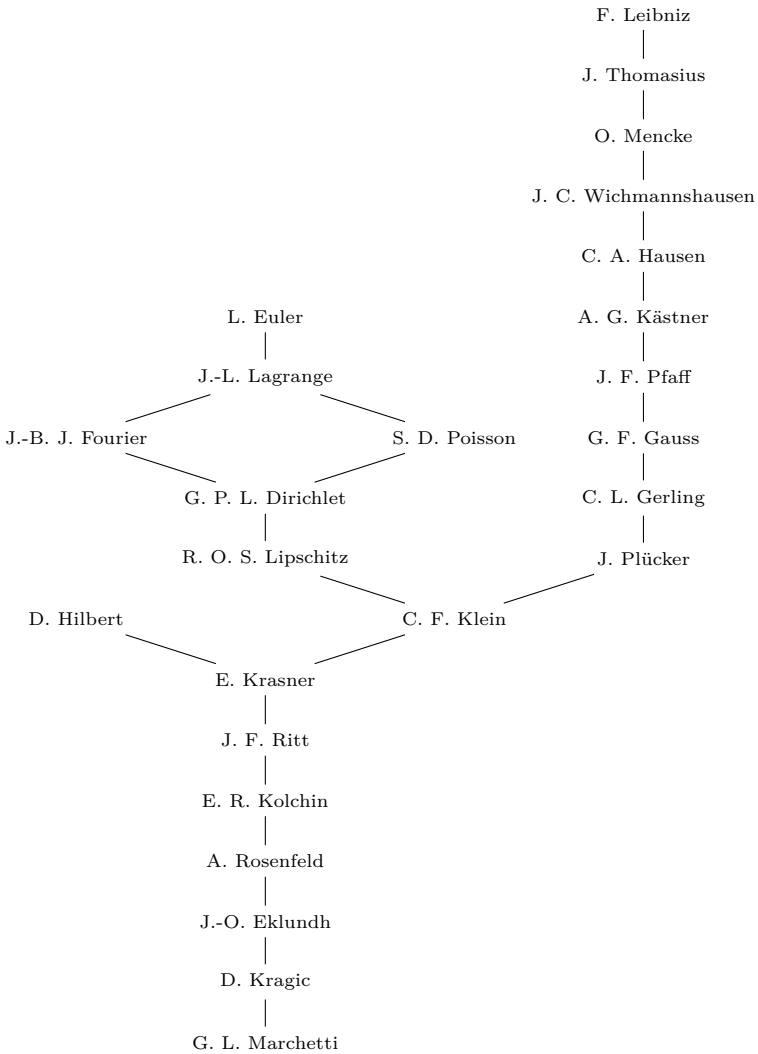
- Thank you, division of Robotics, Perception and Learning, for your vibrant environment promoting discussion, dissemination, and development.

- Thank you, Qualcomm, for hosting me in a warm inspiring Summer.

Above all, thank you, Serena, for so many things that I am unable to explain in lines. No journey would be possible without your presence, and no journey would matter if not together.

Giovanni Luca Marchetti
Somewhere in $\mathbb{RP}^3 \simeq \mathrm{SO}(3)$
January 2024

Academic Ancestors



Contents

Part I

Overview

Chapter 1

Introduction

In this work more questions arise than answers given, for which of course we do not apologize.

–Abstract of [1]

1.1 The Geometry of Data

Let us start with an illustrative example. Consider a dataset of images depicting a scene from different points of view. The corresponding data lie in an ambient space $\mathbb{R}^P$, where P is the finite set of sensor units of the camera – a combination of pixels and channels accounting for color, depth, and so on. Such ambient space is Euclidean of dimension $|P| \gg 0$ proportional to the resolution of the camera. However, the images depicting the scene can possibly belong only to a specific, tiny submanifold $\mathcal{X} \subseteq \mathbb{R}^P$. Indeed, since the camera is constrained to translate and rotate in the three-dimensional physical world, $\mathcal{X}$ is diffeomorphic to the Cartesian product $\mathbb{R}^3 \times \mathrm{SO}(3)$. In the latter, the first factor accounts for translations while $\mathrm{SO}(3)$ is the space of three-dimensional rotations. Therefore, the intrinsic space of data $\mathcal{X}$ is 6-dimensional independently of the resolution, and its topology is non-trivial. Even further, the space $\mathbb{R}^3 \times \mathrm{SO}(3)$ – and especially its translation component $\mathbb{R}^3$ – carries a meaningful geometric structure: it is an isometric copy of the three-dimensional Euclidean world.

The above example illustrates a general phenomenon concerning the nature of data. Even though datasets typically manifest discretely in high-dimensional ambient spaces, they are constrained to low-dimensional submanifolds. Intuitively, the ambient features – pixel-wise color values in the case of images – are highly correlated by an (unknown) entangling mechanism underlying reality. The latter constrains the intrinsic degrees of freedom of the space of 'natural' data, while globally deforming its topology. This defines a basic meta-statistical principle known

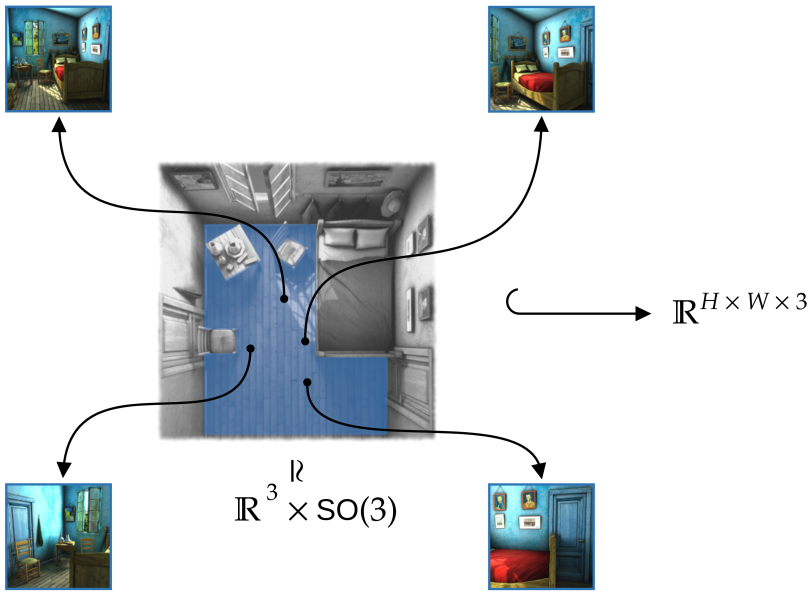


Figure 1.1: A space of images of a scene from different points of view is intrinsically diffeomorphic to $\mathbb{R}^3 \times \mathrm{SO}(3)$.

as *manifold hypothesis* [2]. Crucially, data manifolds are typically equipped with an additional intrinsic structure: they are *geometric* spaces. This means that features such as distances and angles between data should be regarded as semantically meaningful and can be leveraged for statistical inference. In the above example, the metric structure of the translational component $\mathbb{R}^3$ encodes the geometry of the world. Being able to isometrically map data to $\mathbb{R}^3 \times \mathrm{SO}(3)$ enables the determination of the (relative) pose of the camera. If we imagine the images to be collected by an autonomous agent such as a human or a mobile robot, an isometric map enables the latter to localize itself in the environment and to navigate therein. Understanding the geometry of the world is a fundamental perceptual aspect that serves as a basis for intelligent behavior, including reasoning and planning.

All this motivates the problem of extracting the geometry of the data manifold. Therefore, the following general question will be the central focus of this thesis.

Question 1.1.1. *How can the geometric structure of the data manifold be extracted statistically and exploited for inference?*

We refer to the problem raised by the question above as *geometric inference*. This is in line with the terminology from deep learning literature [3], but abstracted to a more general context. Indeed, Question 1.1.1 is vague and can be potentially addressed via a wide spectrum of approaches, ranging from (high-dimensional)

statistics, to computational geometry, to machine learning. To begin with, a foundational challenge is clarifying the notion of 'manifold' on which geometric inference is based upon. To this end, we isolate the following classes of notions, all of which will be explored in this thesis.

- *Discrete manifolds.* In this case, a manifold is a discrete geometric object representing the combinatorial incarnation of a space – typically a graph or, more generally, a simplicial complex. Extracting geometry means constructing a simplicial complex from a given dataset P. Methods in this direction are mostly non-parametric, meaning that the simplicial complex is computed directly from data without optimizing parameters of a statistical model.

- *Fuzzy manifolds.* Here, the notion of a manifold is relaxed in a probabilistic sense i.e., it overlaps with that of a probability distribution. Datapoints in P are interpreted as samples from the latter. This reduces geometry extraction to a (high-dimensional) density estimation problem, which lies at the heart of inferential statistics. A solution can be achieved via non-parametric methods as well as parametric ones, in which case the probabilistic model is fit to data via optimization.

- *Smooth manifolds.* This is the notion at the core of differential geometry, which we consider here at an informal level. In this case, the data manifold is modelled as an ideal continuous object, on which the dataset lies. Geometric inference consists of finding a data *representation* i.e., a map $\varphi\colon \mathcal{X} \to \mathcal{Z}$ from the ambient space of data to a copy of the data manifold $\mathcal{Z}$, referred to as latent space. The aim for the representation is preserving a given geometric structure of data, making it emerge in the latent space. Once extracted, such geometric structure can be exploited for addressing specific tasks over $\mathcal{Z}$. As compared to the original unstructured space $\mathcal{X}$, inference in $\mathcal{Z}$ is potentially geometry-aware and, therefore, significantly more reliable, robust, and efficient. Methods adhering to this paradigm are usually highly parametric and involve optimizing over (the parameters of) φ. The latter is typically a powerful machine learning model, such as a deep neural network. The problem of inferring φ is known in the literature as *representation learning* [4].

Extracting geometry from data rises several challenges, which are entailed in the paradigms mentioned above. First, *computational* aspects are crucial since, generally speaking, the cost of computing geometric quantities is expected to increase drastically as the (intrinsic) dimensionality grows. This phenomenon is known as the *curse of dimensionality* [5], and is ubiquitous across statistics. In order to benefit from geometric tools, it is therefore necessary to mitigate their computational cost, especially in high dimensions. This applies mainly to non-parametric methods, where inference is performed directly. On the other hand, optimization and even data-driven supervision can mitigate the cost by circumventing the computation of raw geometric features. The computational efficiency of simplicial complexes –

together with their associated geometric and topological quantities — lies at the heart of the fields of computational geometry and topology [6], respectively.

Second, it is necessary to clarify what kind of geometric structure needs to be extracted and how. This depends inextricably on the type of data under consideration. In some scenarios, the data manifold is naturally embedded in an ambient space from which structure can be induced directly, such as distances between points. In other instances, this is either unfeasible or leads to undesired results. In the example discussed above, no obvious distance in the ambient space $\mathbb{R}^P$ mimics the intrinsic Euclidean geometry of $\mathcal{X}$ induced by the real world. This raises the need for additional information to perform geometric inference, which can either derive from prior knowledge around data or can be provided via some source of *supervision*. To this end, collecting geometric information is realistic in several practical scenarios, especially in the context of robotics. Indeed, autonomous agents such as robots can discover the geometry of data via specific sensors. In the example discussed above, a mobile robot collecting the images can access the geometry of the world – and therefore of the data manifold – by odometric and internal measurements. These provide the distances and even the actions performed by the robot while navigating, representing a supervisory signal which is rich with intrinsic geometric information. Extracting structure by leveraging on the information collected via interaction with the environment is a central aspect of embodied intelligence known in the literature as *interactive perception* [7].

1.2 Symmetries and Metrics

In this thesis, we will address Question 1.1.1 by discussing a variety of methods revolving around two concepts: *symmetries* and *metrics*. These are mathematical tools that have been regarded as fundamental objects in geometry ever since Euclid's seminal treatise *The Elements* [8]. Both can be leveraged upon in order to understand and extract the geometric structure of data. However, they profoundly differ in their nature, leading to methods that diverge in context, assumptions, and scope.

Metrics are quantities representing relative distances between datapoints. Therefore, they are basic objects that can be exploited for geometric inference for several reasons. First, distances can be induced on data directly from their ambient space – at least locally – assuming the latter is a metric space (e.g., Euclidean), as commonly happens. This avoids the need for supervision, making it possible to design fully unsupervised metric-based methods. Nonetheless, in specific practical scenarios, supervised methods are still relevant since distances can be realistically collected, for example, as signals from sensors. Second, due to the elementary nature of metrics, the latter are suitable for direct computations, leading to non-parametric methods. Specifically, distances are core ingredients for several non-parametric constructions,

for example of simplicial complexes and density estimators. These will be discussed extensively in the present thesis.

Symmetries, on the other hand, represent invertible transformations of the data manifold. Therefore, they are geometric objects exhibiting interesting algebraic structures. The latter are described via the formalism of *groups* and group actions, which lie at the foundation of the mathematical theory of symmetries. Due to their rich structure, groups and symmetries have played a central role in the history and philosophy of mathematics. Most notably, they were highlighted in Klein's *Erlangen Program* [9] – a visionary perspective on geometry elevating symmetries as the main component of the notion of a space. Intuitively, symmetries determine the quantities that are invariant with respect to them, which in turn can be regarded as the only features that are intrinsic to the geometry considered. In other words, symmetries encode the entire geometric structure of spaces. This principle at the core of the Erlangen Program inspires and motivates the effort of designing symmetry-based methods for geometric inference. However, several challenges arise from a statistical perspective. First, differently from metrics, symmetries are typically unfeasible to infer from datapoints or their ambient space. Therefore, either prior knowledge has to be assumed around the geometry of data – such as in the case of group-theoretical convolutional neural networks [10] – or symmetry information needs to be conveyed via supervision. This challenge is evident in the motivating example at the beginning of this chapter, where symmetries are given by change of perspective, resulting in highly non-linear and unpredictable image transformations (see Figure 1.1). Yet, such transformations are intrinsically described as rotations and translations, which, as mentioned before, can be collected by an agent via sensors and leveraged as a supervisory signal for inference. Second, in order to extract geometry via symmetries, it is necessary to deploy powerful statistical tools such as deep neural networks. As a consequence, most symmetry-based methods are highly parametric and fit into the paradigm of representation learning discussed above. Specifically, the aim of representations in this context is to extract geometry by preserving symmetries of data – a property known as *equivariance*. This motivates the recent field of equivariant representation learning, which will be a core focus of this thesis.

1.3 Contributions of the Thesis

In this thesis, we answer Question 1.1.1 by leveraging on metrics and symmetries of data. We explore a variety of approaches and perform both theoretical and empirical analyses. Our contributions range from non-parametric methods in high-dimensional computational geometry and statistics (**Papers A, B, C** below), to deep learning methods for equivariant representation learning (**Papers D, E, F** below), to theoretical analysis of invariant learners (**Paper G** below). The following is the list of papers presented in this thesis. The symbol * denotes equal

contribution.

A: A. Kravberg*, **G. L. Marchetti***, V. Polianskii* A. Varava, F. T. Pokorny, D. Kragic. *Active Nearest Neighbor Regression Through Delaunay Refinement*. In International Conference on Machine Learning (ICML), 2022, [11].

B: V. Polianskii*, **G. L. Marchetti***, A. Kravberg, A. Varava, F. T. Pokorny, D. Kragic. *Voronoi Density Estimator for High-Dimensional Data: Computation, Compactification and Convergence*. In Uncertainty in Artificial Intelligence (UAI), 2022, [12][1].

C: **G. L. Marchetti**, V. Polianskii, A. Varava, F. T. Pokorny, D. Kragic. *An Efficient and Continuous Voronoi Density Estimator*. In International Conference on Artificial Intelligence and Statistics[2] (AISTATS), 2023, [13]. **Notable Paper Award**[2].

D: **G. L. Marchetti***, G. Tegnér*, A. Varava, D. Kragic. *Equivariant Representation Learning via Class-Pose Decomposition*. In International Conference on Artificial Intelligence and Statistics (AISTATS), 2023, [14].

E: L. A. P. Rey*, **G. L. Marchetti***, D. Kragic, D. Jarnikov, M. Holenderski. *Equivariant Representation Learning in the Presence of Stabilizers*. In European Conference on Machine Learning, (ECML-PKDD), 2023, [15].

F: A. Reichlin, **G. L. Marchetti**, H. Yin, A. Ghadirzadeh, D. Kragic. *Back to the Manifold: Recovering from Out-of-Distribution States*. In International Conference on Intelligent Robots and Systems (IROS), 2022, [16].

G: **G. L. Marchetti**, C. Hillar, D. Kragic, S. Sanborn. *Harmonics of Learning: Universal Fourier Features Emerge in Invariant Networks*. Preprint, 2024, [17].

The following is a list of additional papers contributed by the author but not included in this thesis. **Paper X-2** below is a preliminary version of **Paper E** above.

X-1: **G. L. Marchetti***, G. Tegnér*, M. Moletta*, P. Shi, A. Varava, A. Kravberg, D. Kragic. *Learning Coarsened Dynamic Graph Representations for*

Deformable Object Manipulation. In International Conference on Advanced Robotics (ICAR), 2021.

X-2: L. A. P. Rey*, **G. L. Marchetti***, D. Kragic, D. Jarnikov, M. Holenderski. *Equivariant Representations for Non-Free Group Actions*. In NeurIPS Workshop on Symmetries and Geometry in Neural Representations (NeurReps), 2022.

X-3: A. Reichlin*, **G. L. Marchetti***, H. Yin, A. Varava, D. Kragic. *Learning Geometric Representations of Objects via Interaction*. In European Conference on Machine Learning (ECML-PKDD), 2023.

X-4: **G. L. Marchetti**, G. Cesa, K. Pratik, A. Behboodi. *Neural Lattice Reduction: A Self-Supervised Geometric Deep Learning Approach*. In NeurIPS Workshop on Symmetries and Geometry in Neural Representations (NeurReps), 2023[3].

X-5: A. G. Castellanos*, A. A. Medbouhi*, **G. L. Marchetti**, D. Kragic. *HyperSteiner: Computing Heuristic Hyperbolic Steiner Minimal Trees*. Preprint, 2024.

[1]The version included in this thesis contains errata of the published one.
[2]Top $\sim 5\%$ of accepted papers.
[3]Work done during an internship at Qualcomm AI Research, Amsterdam.

Chapter 2

Metric-Based Approaches

Geometry is the art of correct reasoning from incorrectly drawn figures.
–Henri Poincaré

Ever since the earliest approaches to geometry, distances have emerged as an essential tool in the study of spaces. Indeed, a basic model of a geometry is a set equipped with a distance function i.e., a *metric space*.

Definition 2.0.1. A *metric space* is a set $\mathcal{X}$ equipped with a *distance* function $d: \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ such that for all $x, y, z \in \mathcal{X}$:

Definiteness	*Reflexivity*	*Triangle Inequality*
$d(x,y) = 0$ iff $x = y$	$d(x,y) = d(y,x)$	$d(x,z) \leq d(x,y) + d(y,z)$

A map $\varphi: \mathcal{X} \to \mathcal{X}'$ between metric spaces is an *isometry* if it preserves distances i.e., $d(\varphi(x), \varphi(y)) = d(x,y)$ for all $x, y \in \mathcal{X}$.

Distances are ubiquitous in mathematics, appearing in an extremely vast range of scenarios. The most classical example of a metric space is the Euclidean one, given by $\mathcal{X} = \mathbb{R}^n$ equipped with the l_2 distance. More generally, a fundamental class of metric spaces is given by (complete) Riemann manifolds equipped with the geodesic distance i.e., the length of the shortest path between two points.

In the following sections, we will outline methods in discrete mathematics, statistics and machine learning aiming to extract geometry from data by leveraging on metrics. More specifically, we will review constructions of simplicial complexes, density estimators and representation learners in metric spaces. In the former two cases, we will focus on *non-parametric* approaches avoiding parameter optimization or Bayesian inference. The motivation is that optimization, although powerful, is typically computationally expensive and rarely prone to mathematical analysis. Instead, metrics enable explicit optimization-free construction due to their simplicity and flexibility. This will enable us to analyze in detail the theoretical aspects

of the non-parametric methods presented, such as computational complexity and convergence. Lastly, for the most intricate constructions we will only discuss the Euclidean case for simplicity. We envision that these approaches generalize to arbitrary Riemannian manifolds, but leave it for future investigation as outlined in Section 4.2.

2.1 Simplicial Complexes from Metrics

A natural way of answering Question 1.1.1 is by extracting from data a combinatorial structure representing the discrete analogue of a manifold. Such structure is known as *simplicial complex*. The latter is the central object of study of combinatorial and computational topology, and has found major applications in topological data analysis [6].

Definition 2.1.1. A *simplicial complex* with vertices P is a collection Σ of subsets $\sigma \subseteq P$ deemed *simplices* such that:

- All the singletons are simplices i.e., $\{p\} \in \Sigma$ for all $p \in P$,
- Σ is closed w.r.t. taking subsets i.e., if $\tau \subseteq \sigma \in \Sigma$ then $\tau \in \Sigma$.

The *dimension* of a simplex $\sigma \in \Sigma$ is $|\sigma| - 1$.

Intuitively, simplices are combinatorial atomic blocks from which the simplicial complex is built as a whole. When all the simplices have dimension at most 1, simplicial complexes coincide with (undirected) graphs.

In this section, starting from a finite dataset $P \subseteq \mathcal{X}$ in a metric space $\mathcal{X}$, we aim to define a simplicial complex with vertices P. A first example of a canonical construction is the *Vietoris-Rips complex*, which depends on a parameter $h \in \mathbb{R}_{>0}$. The simplices of the Vietoris-Rips complex are the subsets $\sigma \subseteq P$ of diameter less than h i.e., such that $d(x, y) < h$ for all $x, y \in \sigma$ (see Figure 2.2, left). Alternatively, it is possible to build a simplicial complex by assigning a neighborhood to each datapoint and by considering the resulting dual structure – a construction referred to as *Čech complex*.

Definition 2.1.2. Fix a neighborhood $p \in U_p \subseteq \mathcal{X}$ for each $p \in P$. The associated *Čech complex* is the simplicial complex whose simplices are the subsets $\sigma \subseteq P$ such that

$$\bigcap_{p \in \sigma} U_p \neq \emptyset. \tag{2.1}$$

A basic example of neighborhoods in a metric space are *balls*, defined for a point $p \in P$ and a radius $h \in \mathbb{R}_{>0}$ as $B(p, h) = \{x \in \mathcal{X} \mid d(x, p) < h\}$. The Čech complex of balls for a given radius defines another standard construction of a simplicial complex in a metric space (see Figure 2.2, center). This complex is relevant since,

under technical assumptions on $\mathcal{X}$, it is homotopically equivalent to $\cup_p U_p \subseteq \mathcal{X}$ in an appropriate sense by the Nerve Theorem [18]. However, balls are uninformative neighborhoods of datapoints from a geometric perspective. They do not adapt around data and their geometry is controlled by the global radius parameter h. A more adaptive and parameter-free alternative is given by *Voronoi cells*.

Definition 2.1.3. The *Voronoi cell* of $p \in P$ is:

$$C(p) = \{x \in \mathbb{R}^n \mid \forall q \in P \; d(x,q) \geq d(x,p)\}. \tag{2.2}$$

The Voronoi cells intersect at the boundary and cover the ambient space $\mathcal{X}$. The collection $\{C(p)\}_{p\in P}$ is referred to as *Voronoi tessellation* generated by P (see Figure 2.1). The Čech complex of the Voronoi tessellation is referred to as *Delaunay triangulation* (see Figure 2.2, right).

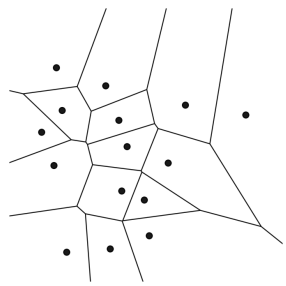


Figure 2.1: An example of a Voronoi tessellation.

Voronoi tessellations are a central object of study in computational geometry. Although their systematic analysis in arbitrary dimensions dates to the beginning of the 20^{th} century [19], they sparingly appear in Descartes' and Dirichlet's work [20, 21]. From a broader perspective, Voronoi tessellations represent the geometric structure underlying the nearest neighbor search, sometimes referred to as 'post-office problem' [22]. Therefore, they manifest – implicitly or explicitly – in methods involving nearest neighbors, a classical examples of which is k-nearest neighbor classification and regression [23]. For $k = 1$, the latter is defined as follows. Given ground-truth values $\{y_p\}_{p\in P}$ for datapoints, the unknown function generating the values is estimated as:

$$\widetilde{f}(x) = y_{\overline{p}}, \qquad \overline{p} = \operatorname*{argmin}_{p\in P} d(x,p). \tag{2.3}$$

Therefore, the nearest neighbor regressor is locally constant over the Voronoi cells and is undefined at their boundary. Overall, it constitutes an example of a simple yet effective metric-based non-parametric regressor leveraging on the geometry of Voronoi tessellations.

Despite their simplicity and ubiquity, Voronoi tessellations and Delaunay triangulations exhibit remarkable geometric properties. In order to illustrate this, from now on we focus on Euclidean ambient spaces $\mathcal{X} = \mathbb{R}^n$ for simplicity. In this case, the Voronoi cells are arbitrary n-dimensional convex polytopes whose k-dimensional boundaries are given by points in $\mathbb{R}^n$ equidistant to $n-k+1$ datapoints. For generic datasets P, this implies that the Delaunay triangulation is an embedded simplicial complex, meaning that all the simplices have dimension at most n and the convex hulls $\mathrm{Conv}(\sigma)$ of the n-dimensional simplices intersect only at their boundaries. Even further, the Delaunay triangulation satisfies the following fundamental property.

Proposition 2.1.1 (Empty Sphere Property; [24])**.** *For generic datasets $P \subseteq \mathbb{R}^n$, the sphere passing through the vertices of an n-dimensional Delaunay simplex (i.e., the circumsphere) does not contain other datapoints in its interior. The Delaunay triangulation is the only embedded simplicial complex covering* $\mathrm{Conv}(P)$ *with this property.*

The intuition behind the empty sphere property is that Delaunay simplices tend to avoid acute angles, since the latter would increase the radius of the corresponding circumsphere. This geometric regularity of the Delaunay triangulation, together with the fact that it comes with no hyperparameters, has motivated its usage in a variety of constructions and methods. For example, the Delaunay triangulation can be deployed to define a piece-wise linear interpolator by extending given scalar values $\{y_p\}_{p\in P} \subseteq \mathbb{R}$ linearly over simplices. This interpolator is optimal in a certain sense among all the piece-wise linear ones obtained from triangulations of $\mathrm{Conv}(P)$ [25, 26]. Moreover, low-dimensional ($n = 2, 3$) Delaunay triangulations have found applications in computer graphics for the purpose of meshing and *mesh refinement*. The goal of the latter is to progressively add vertices to a mesh in order to increase the rendering quality. Ruppert's algorithm [27] and Chew's second algorithm [28] are popular solutions to this problem, both relying on the idea of adding circumcenters (i.e., vertices of Voronoi cells) of poorly-behaved Delaunay triangles. Due to the Empty Sphere Property, these algorithms possess strong halting guarantees.

However, all the above-mentioned constructions of simplicial complexes raise a computational challenge, especially in terms of memory. The number of simplices in the Vietoris-Rips complex and the Čech complex of balls ranges from $|P|$ for $h = 0$ to $2^{|P|}$ for $h \gg 0$, of which the number of k-dimensional ones ranges from 0 to $\binom{|P|}{k+1} = \mathcal{O}(|P|^{k+1})$. For the Delaunay triangulation, the number of simplices is significantly lower, but still unfeasible to store in memory as the dimension n grows. Namely, the number of n-dimensional simplices in the Delaunay triangulation is bounded by (see [29]):

$$\binom{|P| - \lfloor \frac{n+1}{2} \rfloor}{|P| - n} + \binom{|P| - \lfloor \frac{n+2}{2} \rfloor}{|P| - n} = \mathcal{O}\left(|P|^{\lceil \frac{n}{2} \rceil}\right). \tag{2.4}$$

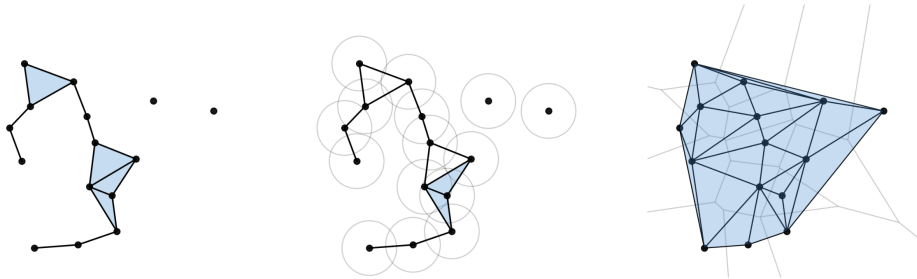


Figure 2.2: Three constructions of simplicial complexes from metric spaces.

The bound is strict since it is achieved when P lies on the n-dimensional moment curve $\{(t, t^2, \cdots, t^n)\}_{t\in\mathbb{R}} \subseteq \mathbb{R}^n$. From an algorithmic perspective, the most popular method to compute the Delaunay triangulation is by lifting P to the standard paraboloid in $\mathbb{R}^{n+1}$, where simplices cover the lower convex hull [30]. The latter can be computed by standard techniques from computational geometry, which however scale exponentially in time complexity w.r.t. n [31]. Alternatively, a ray-casting Monte Carlo Markov Chain approach has been recently proposed [32]. It consists of randomly walking across the boundaries of Voronoi cells, collecting the vertices found along the way together with their corresponding Delaunay simplices. This yields an approximate stochastic technique with variable memory and time complexity depending on the length and number of random walks performed.

In **Paper A**, we leverage on the geometry of the Delaunay triangulation for the purpose of non-parametric *active learning*. The aim of the latter is updating the dataset P iteratively via a querying procedure in order to improve the performance of a given learner – see [33] for a survey. Specifically, we propose an active querying procedure for the nearest neighbor regressor (Equation 2.3). Note that non-parametric methods are particularly suitable for active learning since they are unaffected by catastrophic forgetting – a persistent challenge of parametric learners [34]. This enables to query datapoints based on the geometry of the current dataset alone, without retraining the given learner. Our method – deemed Active Nearest Neighbor Regressor (ANNR) – is based on the intuition that the most informative regions are the ones where the graph of the (estimated) function exhibits the most variation. The latter is measured by the volume of the function's graph, discretized via the Delaunay triangulation. Formally, given the current dataset P at some iteration of the algorithm, ANNR queries the circumcenter of the Delaunay simplex σ maximizing (see Figure 2.3):

$$\mathrm{Vol}\left(\mathrm{Conv}(\hat{\sigma})\right), \tag{2.5}$$

where Conv denotes the convex hull and $\hat{\sigma} \subseteq \mathbb{R}^{n+1}$ is the lifting of σ to the graph

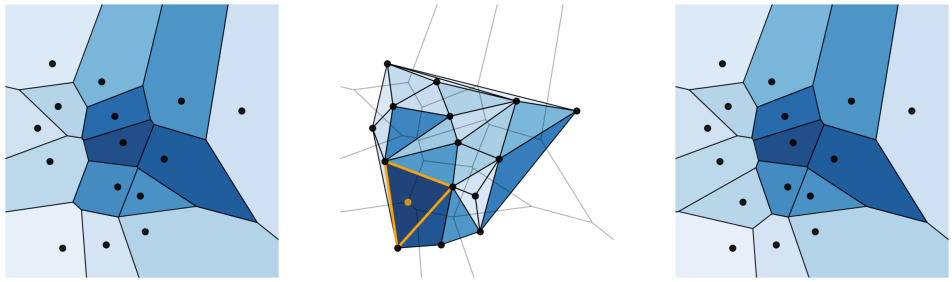


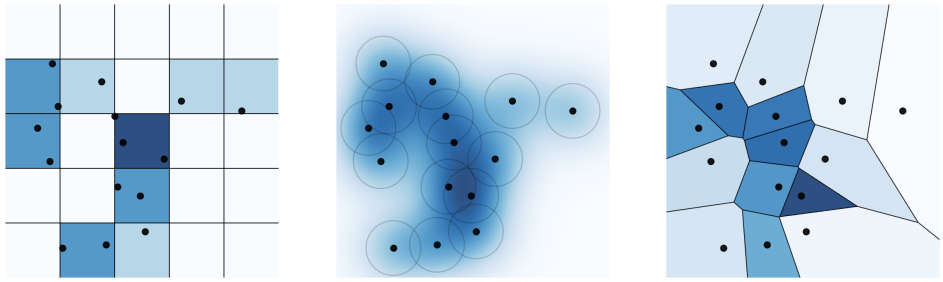
Figure 2.3: Illustration of the querying procedure of ANNR. The simplex maximizing Equation 2.5 and its circumcenter are highlighted in orange.

of (a multiple of) the ground-truth function i.e., $\hat{\sigma} = \{(p, \lambda y_p)\}_{p \in \sigma}$. Here, $\lambda \in \mathbb{R}_{>0}$ is a hyperparameter regulating the exploration-exploitation tradeoff typical of active learning. We compute the volume in Equation 2.5 via the Cayley-Menger determinant [35], while we sample the Delaunay simplices (together with their circumcenters) via the approach from [32]. Note that the circumcenters of Delaunay simplices correspond to the vertices of Voronoi cells, which in turn represent the points where the nearest neighbor regressor is maximally discontinuous. ANNR is inspired by the above-mentioned Ruppert's and Chew's algorithms for mesh refinement in computer graphics. Intuitively speaking, collecting information for the purpose of learning is analogous to refining the (high-dimensional) mesh given by the graph of the estimated function. Similarly to mesh refinement, we prove halting guarantees for ANNR by leveraging on the Empty Sphere Property of the Delaunay triangulation.

2.2 Non-Parametric Density Estimation

An alternative way to address Question 1.1.1 is by extracting a *fuzzy* manifold from data i.e., a probability density function. This results in a statistical rephrasing of the question, reducing it to a density estimation problem. More precisely, assuming that the metric space $\mathcal{X}$ is equipped with a Borel measure, we think of P as being sampled from an absolutely continuous probability distribution with unknown density. The goal is recovering the latter by associating to $P \subseteq \mathcal{X}$ a probability density function $\rho_P \in \mathrm{L}^1(\mathcal{X})$. This problem lies at the heart of inferential statistics and can be approached from several perspectives. Similarly to the case of simplicial complexes, we aim to discuss non-parametric approaches, together with their theoretical properties. This means that we focus on density estimators providing ρ_P in closed form by leveraging the geometric structure of $\mathcal{X}$ in an efficient manner.

The first historical example of a non-parametric density estimator is given by *histograms*, dating back to the early days of statistics [36]. Given a tessellation of



Histograms KDE VDE

Figure 2.4: Three classical non-parametric density estimators.

the ambient space $\mathcal{X} = \cup_i C_i$ into closed cells of finite volume intersecting at their boundary, histograms estimate the density as

$$\rho_P(x) = \frac{|P \cap C|}{|P|\mathrm{Vol}(C)}, \tag{2.6}$$

where C is the cell containing x. The density is therefore locally constant on the cells and is undefined at their boundaries (which are assumed to be negligible). A typical choice for the tessellation in the case $\mathcal{X} = \mathbb{R}^n$ is the regular partition into hypercubes of a given side length (see Figure 2.4, left). Histograms are computationally efficient: both the evaluation of the estimated density and sampling from the latter take linear time w.r.t. the dataset size $|P|$. However, they do not enjoy any geometrical nor statistical property because of their basic nature.

Arguably, the most widespread density estimator is the *Kernel Density Estimator* (KDE) [37, 38]. Given a kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ which is integrable in the second variable, the estimated density is defined as

$$\rho_P(x) = \frac{1}{|P|Z_p} \sum_{p \in P} K(p, x), \tag{2.7}$$

where $Z_p = \int_{\mathcal{X}} K(p, y)\ \mathrm{d}y$. In other words, KDE is a uniform mixture of the (unnormalized) densities $K(\cdot, p)$ for $p \in P$, and in particular it can be evaluated and sampled from in linear time w.r.t. $|P|$. A typical expression for the kernel is in the form $K(p, x) = K\left(\frac{d(x,p)}{h}\right)$, where $K : \mathcal{X} \to \mathbb{R}_{\geq 0}$ denotes by abuse of notation a function with appropriate integrability properties while $h \in \mathbb{R}_{>0}$ is a hyperparameter deemed *bandwidth*. For example, a standard choice is given by the Gaussian kernel (see Figure 2.4, center):

$$K(d) = e^{-d^2}. \tag{2.8}$$

The kernel represents a prior choice of a local geometry around data, which is then averaged in order to obtain the estimated density. Intuitively speaking, KDE can indeed be seen as the fuzzy analogue of the Čech complex, where the neighborhoods U_p are replaced by the soft neighborhoods given by the (unnormalized) density $K(\cdot, p)$. To visualize this, compare the centers of Figure 2.2 and 2.4. The bandwidth plays a role analogue to the radius in the Čech complex of balls i.e., it controls the scale of the local geometry. The value of h crucially affects the estimator by distributing the mass of the density around data or away from it. Bandwidth selection is therefore a fundamental problem, and several heuristic rules have been proposed to address it [39, 40]. Nonetheless, KDE suffers from the prior choice of local geometry given by the kernel, negatively biasing the estimated density. An exemplary mathematical consequence of this is the lack of convergence of KDE to the ground truth density. Assuming P is sampled from ρ, ρ_P converges (in an appropriate stochastic sense) as $|P| \to +\infty$ to the convolution between ρ and K. Therefore, KDE does not converge to the ground-truth density unless the bandwidth is chosen adaptively to P such that $h \to 0$ at an appropriate rate [41]. The latter intuitively annihilates the local geometric bias of KDE, recovering convergence.

Analogously to simplicial complexes, in order to design a geometrically adaptive density estimator it is convenient to deploy Voronoi tessellations. To this end, the *Voronoi Density Estimator* (VDE) has been proposed [42]. The estimated density is defined as inversely proportional to the volumes of Voronoi cells i.e.,

$$\rho_P(x) = \frac{1}{|P|\mathrm{Vol}(C(p))}, \tag{2.9}$$

where $C(p)$ is the Voronoi cell containing x. The density is locally constant on Voronoi cells and undefined at their (negligible) boundary. Therefore, VDE can be intuitively understood as an adaptive version of histograms (compare with Equation 2.6). However, despite its geometric structure, VDE comes with a number of shortcomings:

- Since the Voronoi cells can be unbounded, it is necessary to assume that the measure over $\mathcal{X}$ is finite. This is not the case for the Lebesgue measure over $\mathcal{X} = \mathbb{R}^n$, which is usually circumvented by restricting the measure to a chosen compact subset $A \subseteq \mathcal{X}$ containing P.

- The volume of Voronoi cells is challenging to compute. Assuming $\mathcal{X} = \mathbb{R}^n$, the latter are arbitrary convex polytopes, and determining their volume is therefore a hard computational problem [43].

In **Paper B**, we introduce a new version of VDE addressing the challenges above. The proposed density estimator – deemed *Compactified Voronoi Density*

Estimator (CVDE) – is defined as:

$$\rho_P(x) = \frac{K(p,x)}{|P|Z_p}, \tag{2.10}$$

where K is a kernel, $Z_p = \int_{C(p)} K(p,y)\ \mathrm{d}y$ and $C(p)$ is the Voronoi cell containing x. Therefore, CVDE amends for unbounded cells by deploying the local (unnormalized) density $K(p,\cdot)$, bridging the gap between KDE and VDE (see Figure 2.6, left). In order to address the computational challenge of VDE, we propose to estimate the volumes of Voronoi cells approximally via a Monte Carlo ray-casting approach. To this end, we again focus on the Euclidean space $\mathcal{X} = \mathbb{R}^n$ and rephrase the desired integral in spherical coordinates as

$$\int_{C(p)} K(p,y)\ \mathrm{d}y = \int_{\mathbb{S}^{n-1}} \underbrace{\int_0^{l(p+\sigma)} t^{n-1} K(p, p+t\sigma)\ \mathrm{d}t}_{\text{Conical Integral}}\, \mathrm{d}\sigma \tag{2.11}$$

and approximate the right-hand side by sampling rays uniformly on the sphere $\mathbb{S}^{n-1}$. The conical integral in Equation 2.11 is performed over the ray originating from p in the direction of σ and ending at the boundary of the Voronoi cell. Its length is given by:

$$l(x) = \sup\left\{ t \geq 0 \mid p + t\frac{x-p}{d(x,p)} \in C(p) \right\} = \min_{q \neq p,\ l^q(x) \geq 0} l^q(x), \tag{2.12}$$

where

$$l^q(x) = \frac{d(q,p)^2}{2\left\langle \frac{x-p}{d(x,p)}, q-p \right\rangle}. \tag{2.13}$$

We refer to Figure 2.5 for an illustration. Therefore, $l(x)$ is computable in linear

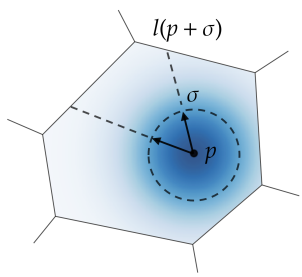
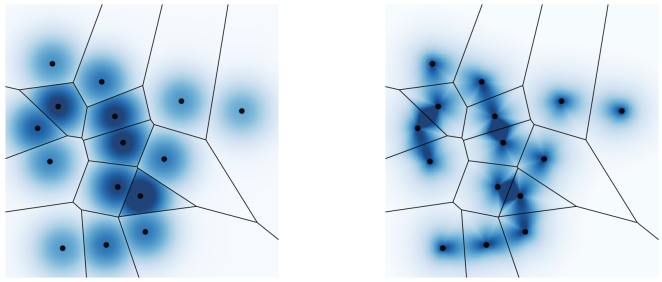


Figure 2.5: Depiction of the rays and their lengths, involved in the construction of CVDE and RVDE.

time w.r.t. $|P|$, making the cost of evaluating CVDE at a given point $x \in \mathbb{R}^n$ linear w.r.t. both $|P|$ and the number of Monte Carlo samples on the sphere. A similar



CVDE RVDE

Figure 2.6: The two Voronoi density estimators introduced in this thesis.

technique known as *hit-and-run* can be used to sample from the estimated density. Hit-and-run iteratively produces points on rays towards random directions, resulting in a Markov Chain stabilizing at the distribution given by the restriction of K to $C(p)$. These Monte Carlo ray-casting methods are analogous – and actually a particular case of – the methods from [32] deployed to compute the Delaunay triangulation in ANNR. In summary, these approximations mitigate the computational burden of VDE, enabling its usage on datasets of large cardinality.

From a theoretical perspective, the main contribution of **Paper B** is a fundamental convergence result for estimators that are based on Voronoi tessellations such as VDE and CVDE. By combining tools from high-dimensional Euclidean geometry and measure theory, we formally prove the following.

Theorem 2.2.1. *Suppose that $P \mapsto \rho_P$ is a density estimator over $\mathbb{R}^n$ satisfying for all $p \in P$:*

$$\int_{C(p)} \rho_P(x)\ \mathrm{d}x = \frac{1}{|P|}. \tag{2.14}$$

Suppose moreover that the ground-truth density ρ has full support and consider the estimated density ρ_P as being random in P sampled from ρ. Then as $|P| \to +\infty$, ρ_P converges to ρ in distribution w.r.t. x and in probability w.r.t. P.

The above result highlights the geometric advantages of Voronoi tessellations in terms of density estimation. Since the Voronoi cells adapt geometrically around data, they force convergence to the ground-truth density as long as equal mass is distributed on each cell by the estimator. Compared to KDE, this guarantees convergence without the requirement for the bandwidth of the kernel to vanish since, intuitively, the local geometry of the estimator adapts automatically via the Voronoi cells.

While CVDE improves the computational cost of VDE, it is still more expensive than KDE due to its dependency on the sampled rays over the sphere. Moreover, it

exhibits the same jumping discontinuity as VDE at the boundary of Voronoi cells. As a consequence, the estimated density is unstable and unsuitable for gradient-based parametric extensions. Therefore, a natural challenge is designing a (non-parametric) density estimator satisfying Equation 2.14 which is continuous and more computationally efficient. To this end, in **Paper C** we propose a novel approach deemed *Radial Voronoi Density Estimator* (RVDE). Our core idea is to obtain the desired condition expressed in Equation 2.14 by forcing the conical integral in Equation 2.11 to be constant. To this end, we fix a radial kernel $K : \mathbb{R} \to \mathbb{R}_{\geq 0}$ and introduce (the inverse of) a *radial bandwidth* $\beta : \mathbb{R}_{>0} \to \mathbb{R}$ defined implicitly by the integral equation:

$$\int_0^l t^{n-1} K(\beta(l)t) \, \mathrm{d}t = \alpha, \tag{2.15}$$

where $\alpha > 0$ is a hyperparameter. Assuming K satisfies a few technical conditions, this equation possesses unique solution $\beta(l)$ for all $l > 0$ which can be computed numerically via, for example, the Newton-Raphson method. The estimated density is then defined as (see Figure 2.6, right):

$$\rho_P(x) = \frac{K(\beta(l(x)) \; d(x,p))}{\alpha |P| w_n}, \tag{2.16}$$

where $w_n = \mathrm{Vol}(\mathbb{S}^{n-1})$. Since $l(x)$ is continuous in x and computable in linear time w.r.t. $|P|$, the same holds for the estimated density. Additionally, it is possible to sample from the latter in linear time w.r.t. $|P|$ by sampling uniformly along rays around datapoints. Overall, RVDE simultaneously satisfies the aforementioned continuity, efficiency, and convergence properties. One downside is the presence of the hyperparameter α, which is analogous to the bandwith in KDE and requires to be tuned. However, α comes with a geometric interpretation in that it controls the distribution of the modes of RVDE. Indeed, we prove that the latter lie either on the datapoints or on (the midpoints of) the edges of the Gabriel graph – an easily-computable subgraph of the Delaunay graph – based on a threshold monotonically determined by α. This inspires a hyperparameter selection procedure based on the statistics of the Gabriel graph.

2.3 Metric and Contrastive Learning

Lastly, we discuss the representation learning methods relying on the metric structure of data. As anticipated in Chapter 1, a *representation* is simply a map $\varphi \colon \mathcal{X} \to \mathcal{Z}$, where $\mathcal{Z}$ is a space referred to as latent. Ideally, the aim of a representation is forcing some intrinsic structure of data to emerge in the latent space. Such structure can be leveraged upon for subsequent inference over $\mathcal{Z}$. In *metric learning*, the aim is preserving distances in $\mathcal{X}$ or, in other words, encouraging φ to

be (as close as possible to) an isometry. Metric learning can be formulated as an optimization problem as follows. Given a dataset $P \subseteq \mathcal{X}$ in a metric space and a parametrized class of functions φ_θ with parameter $\theta \in \Theta$, such as deep neural networks, the prototypical objective of metric learning is minimizing:

$$\mathcal{L}(\theta) = \frac{1}{|P|^2} \sum_{(p,q)\in P\times P} \mid d(p,q) - d(\varphi_\theta(p),\ \varphi_\theta(q)) \mid. \tag{2.17}$$

In the above, d denotes by abuse of notation the distance on both $\mathcal{X}$ and $\mathcal{Z}$. Indeed, $\mathcal{L}(\theta) = 0$ if and only if φ_θ is an isometry when restricted to P. Several variations of this objective have been proposed, for example Sammon's loss incorporating scale-invariance [44] and the triplet loss leveraging on three datapoints simultaneously [45]. We refer the reader to [46] for a survey on metric learning. Crucially, the latent space $\mathcal{Z}$ together with its metric is set a priori and constitutes the fundamental design choice of metric learning. A variety of metrics beyond the Euclidean one have been proposed for this purpose, for example hyperbolic metrics for hierarchically-structured data [47] or the Cayley-Klein metric incorporating all the uniform non-Euclidean geometries simultaneously [48]. On the other hand, the distances between datapoints in P are typically induced from the ambient space of data [49], after an eventual preprocessing involving building a simplicial complex and taking geodesic distances over the latter [50]. As an alternative, in specific practical scenarios distances can originate from some form of supervision such as measurements collected via a sensor.

A popular representation learning paradigm related to metric learning is *contrastive learning* – see [51] for a survey. The overall aim is inferring representations such that similar data lie close in the latent space while dissimilar ones lie far apart. To this end, contrastive learning assumes that the dataset is organized in two sets $\mathcal{D}_+, \mathcal{D}_- \subseteq P\times P$, corresponding to similar and dissimilar pairs of datapoints respectively. The desired structure in the representation is then enforced by minimizing an objective in the following form:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{D}_+|} \sum_{(p,q)\in\mathcal{D}_+} d(\varphi_\theta(p),\varphi_\theta(q)) - \frac{1}{|\mathcal{D}_-|} \sum_{(p,q)\in\mathcal{D}_-} d(\varphi_\theta(p),\varphi_\theta(q)). \tag{2.18}$$

The above objective is classically referred as 'siamese' since it requires the same function φ to be computed twice for each pair. The framework was originally introduced in [52] and later expanded in [53]. Contrastive learning bears similarity to metric learning in that the metric d over P is replaced by two binary relations $\mathcal{D}_\pm$ indicating, intuitively, whether data are close or far from each other. In several scenarios, only a notion of similarity is available, raising the need to design a contrastive learning framework relying on $\mathcal{D}_+$ alone. To this end, the second summand of Equation 2.18 is replaced by a term encouraging data to spread apart in $\mathcal{Z}$ or, in other words, φ_θ to be injective. This is necessary in order to avoid φ_θ

collapsing to trivial degenerate solutions such as constant maps. A popular loss for this purpose is a discrete entropy term, for example the (negative) soft minimum distance between datapoints [54]:

$$\log \sum_{(p,q)\in P\times P} e^{-d(\varphi_\theta(p),\ \varphi_\theta(q))}. \tag{2.19}$$

Alternatives to the above term have been designed, including margin penalties over the minimum distance – see [55] for an overview. Since entropy-like objectives are well-behaved on bounded metric spaces, it is customary to deploy a compact latent space $\mathcal{Z}$, for example by normalizing the output of φ_θ on a sphere and by deploying cosine similarity as the latent metric [56]. Lastly, it has been recently argued that the additional term such as the one in Equation 2.18 might be completely omitted by bootstrapping the learning dynamics [57].

Analogously to metric learning, the similar/dissimilar pairs in contrastive learning can be inferred unsupervisedly via clustering techniques or, more commonly, can come from forms of supervision or domain knowledge. Data might be defined to be similar if they belong to the same semantic class [52] or if they are obtained from the same datum via a priorly-selected set of augmentations [57, 58]. Since augmentations often correspond to (invertible) transformations of the data space, in the latter case φ_θ learns a mapping which is *invariant* to such transformations. This draws a connection between contrastive learning and symmetry-based machine learning methods, which we will discuss in the next chapter.

Chapter 3

Symmetry-Based Approaches

All of mathematics is a tale about groups.
–Henri Poincaré

In this chapter, we discuss the problem of extracting geometry from data (Question 1.1.1) by leveraging on their *symmetries*. Intuitively speaking, symmetries are invertible transformations of a given space. Despite such an elementary premise, they constitute a fundamental structure capturing the intrinsic geometry of the data manifold. This has elevated symmetries as a central object of study in geometry, paving the way to their usage in the contemporary fields of statistics and machine learning.

The importance of symmetries in mathematics has been highlighted in Klein's *Erlangen Program* – a foundational perspective initiated by a pioneering essay [9]. The grounding principle is that features and observables are relevant for a given geometry if, and only if, they are invariant to the underlying symmetries. In other words, symmetries characterize everything that is relevant to the geometric structure and, therefore, determine the geometry itself. For example, distances and angles between points in a Euclidean space are relevant for rigid geometry but not for linear algebra since they are invariant to isometries but not to all the linear transformations. This has historically shifted the focus from a point-set view on geometry to a symmetry-based one, ultimately leading to the development of the theory of categories – an abstract approach to mathematics juxtaposing transformations of objects to the objects themselves [59]. In physics, an earlier consequence of the Erlangen Program is marked by the discovery of special relativity. The latter was based on the observation that the principles of electromagnetism are not invariant to the symmetries of the Euclidean geometry, but to the ones of the (non-positive) four-dimensional Minkowsky metric. That is, the transformational structure of physical equations implies that spacetime is naturally equipped with a non-Euclidean metric, highlighting how symmetries determine geometry and not

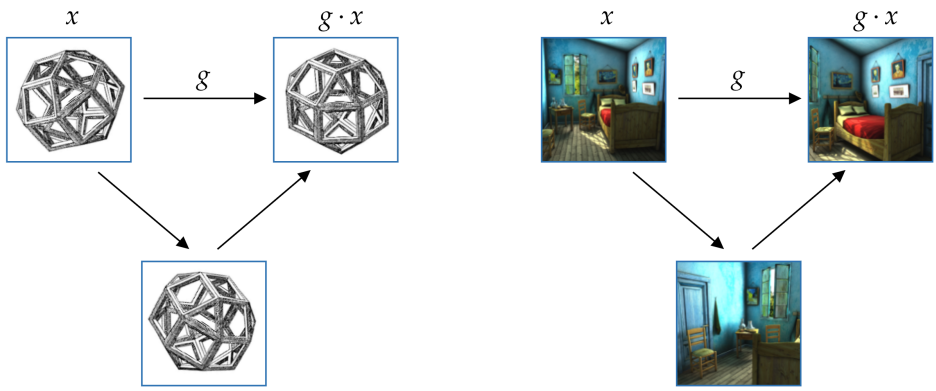


Figure 3.1: Two examples of group actions over image spaces.

vice versa.

Differently from Chapter 2, in what follows we will focus mainly on representation learning approaches. Indeed, symmetries are more elaborate mathematical objects than metrics, involving sets of transformations with algebraic structure. Therefore, they are less prone to non-parametric approaches, typically requiring powerful machine learning tools such as deep neural networks. We will therefore avoid discussing the computational properties of the methods presented, focusing instead on foundational mathematical aspects. For example, we will address the well-posedness of symmetry-based learning, providing theoretical guarantees for the learning problems based on the nature of the available data.

3.1 The Mathematics of Symmetries

We start by introducing the mathematical formulation of the core concepts around symmetries. Instead of defining the latter as single entities, it is convenient to formalize the operations involving them. Indeed, symmetries can be inverted and combined via composition in (at least) pairs. These operations and their properties are axiomatized by the abstract algebraic object known as *group*.

Definition 3.1.1. A group is a set G equipped with a *composition map* $G \times G \to G$ denoted by $(g, h) \mapsto gh$, an *inversion map* $G \to G$ denoted by $g \mapsto g^{-1}$, and a distinguished *identity element* $1 \in G$ such that for all $g, h, k \in G$:

Associativity	*Inversion*	*Identity*
$g(hk) = (gh)k$	$g^{-1}g = gg^{-1} = 1$	$g1 = 1g = g$

A map $\rho : G \to G'$ between groups is called *homomorphism* if $\rho(gh) = \rho(g)\rho(h)$ for all $g, h \in G$.

Abstractly speaking, a symmetry is by definition an element of a group. Groups satisfying $gh = hg$ for all $g, h \in G$ deemed *commutative* or, alternatively, *Abelian*. Classical examples of discrete groups include the modular integers $\mathbb{Z}/N$ equipped with addition (deemed the cyclic groups) and the permutations of a finite set equipped with composition (deemed the symmetric group). A differentiable manifold equipped with a group structure (with differentiable composition and inversion maps) is deemed *Lie group*. Examples of the latter include the invertible operators $\mathrm{GL}(V)$ of a finite-dimensional vector space V, together with its subgroup of orthogonal operators $\mathrm{O}(V)$ when V is a real Euclidean space and the subgroup of unitary operators $\mathrm{U}(V)$ when V is a complex Hilbert space.

Another fundamental concept in group theory is the one of group *actions*, which formalize the concept of a space $\mathcal{X}$ having a given group of symmetries.

Definition 3.1.2. An action by a group G on a set $\mathcal{X}$ is a map $G \times \mathcal{X} \to \mathcal{X}$ denoted by $(g, x) \mapsto g \cdot x$, satisfying for all $g, h \in G,\ x \in \mathcal{X}$:

Associativity	*Identity*
$g \cdot (h \cdot x) = (gh) \cdot x$	$1 \cdot x = x$

In general, the following actions can be defined for arbitrary groups: G acts on any set *trivially* by $g \cdot x = x$, and G acts on itself seen as a set via (left) *multiplication* by $g \cdot h = gh$. Further examples are $\mathrm{GL}(V)$ and $\mathrm{U}(V)$ acting on V by evaluating operators. Group actions induce classes in $\mathcal{X}$ by identifying points related by a symmetry.

Definition 3.1.3. Consider the equivalence relation on $\mathcal{X}$ given by deeming x and y equivalent if $y = g \cdot x$ for some $g \in G$. The induced equivalence classes are called *orbits*, and the set of orbits is denoted by $\mathcal{X}/G$.

For example, the orbits of the trivial action are singletons, while the multiplication action has a single orbit. Intuitively, an orbit may be interpreted as an invariant, maximal class of data induced by the symmetry structure.

Maps between spaces acted upon by the same group that preserve the corresponding symmetries are deemed *equivariant*.

Definition 3.1.4. A map $\varphi\colon\ \mathcal{X} \to \mathcal{Z}$ between sets acted upon by G is called *equivariant* if $\varphi(g \cdot x) = g \cdot \varphi(x)$ for all $g \in G, x \in \mathcal{X}$ or, alternatively, if the following diagram commutes for all $g \in G$:

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{g\cdot} & \mathcal{X} \\ {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle\varphi} \\ \mathcal{Z} & \xrightarrow[g\cdot]{} & \mathcal{Z} \end{array}$$

An equivariant map φ is called *invariant* if G acts trivially on $\mathcal{Z}$ or, explicitly, if $\varphi(g \cdot x) = \varphi(x)$. It is called *isomorphism* if it is bijective.

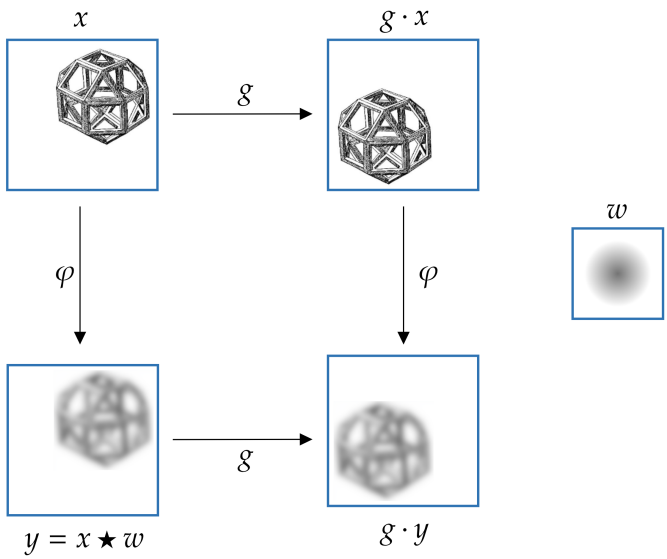


Figure 3.2: Convolutions are equivariant linear maps.

3.2 Group Convolutions

In this section, we discuss a fundamental class of linear equivariant maps, on which several modern machine learning models are based upon. To begin with, we assume that G is finite in order to avoid subtleties arising in functional analysis and measure theory. Everything that follows can be extended to general compact groups with their corresponding Haar measure [60].

Consider the data space $\mathcal{X} = \mathbb{R}^G$ i.e., the free vector space generated by G. Its elements $(x_g)_{g \in G}$ can be interpreted as (scalar) *signals* over the symmetry group. Such spaces are ubiquitous in applications since they encompass data ranging from images, to sound, to colored meshes. For example, if G is a product of two cyclic groups of modular integers $G = (\mathbb{Z}/H) \times (\mathbb{Z}/W)$, $\mathcal{X}$ models the space of (grayscale) images of resolution $H \times W$. The fundamental algebraic aspect of $\mathcal{X}$ is that it is equipped with the *convolution* product, given by:

$$(x \star y)_g = \sum_{h \in G} x_h y_{h^{-1} g}. \tag{3.1}$$

The convolution is associative, and it is commutative if, and only if, G is. Moreover, G acts on $\mathcal{X}$ via $g \cdot x = \delta_g \star x = (x_{g^{-1}h})_{h \in G}$, where δ_g is the canonical basis vector. In other words, G permutes the domain coordinates of the signal. It is straightforward to see that for a fixed w, the map $x \mapsto x \star w$ is equivariant (see Figure 3.2), and it can be proven that any linear equivariant map $\mathcal{X} \to \mathcal{X}$ is of this form for an appropriate w [61]. This has motivated the deployment of convolutions as (linear) layers in machine learning models, with w as an optimizable parameter. In the case of images, the original proposal dates back to [62] and has been popularized by its usage in the ground-breaking image classifier AlexNet [63]. Convolutional neural network layers for general groups have been introduced in [10, 64]. These models have inspired a plethora of extensions and variations, most notably graph neural networks [65] – models equivariant to automorphisms of the input graph. This has ultimately led to attention mechanisms, which are equivariant to permutations and are nowadays widely applied in various domains, ranging from natural language processing [66] to vision [67]. From a broader perspective, the history of deep learning suggests that the the principle of (linear) equivariance has been the driving force in the development of contemporary neural architectures.

Even though convolutional layers are simple and effective machine learners, they present a number of drawbacks. First, convolutions are (equivariant) transformations of $\mathcal{X}$. As a consequence, the dimensionality of the (ambient space of) data can not be reduced, limiting the expressivity and the compression capabilities of the model. This is typically amended by introducing pooling operations [63], which however break equivariance to an extent. Second, convolutions apply exclusively to spaces of signals, on which G acts by permuting coordinates. This is not the case in several scenarios, since the group action over data can be more complex and even *unknown* a priori. For example, consider the scenario presented in Section 1.1 involving the group $G = \mathbb{R}^3 \times \mathrm{SO}(3)$ acting on the space of images of a scene by change of perspective (see Figure 3.1, right). Such an action deviates significantly from pixel permutation: the colors change unpredictably as the field of view of the image shifts across the environment. In this case, it is unreasonable to assume the group action to be understood a priori since it would require perfect knowledge of the given scene.

3.3 Equivariant Representation Learning

Based on the discussion in the previous section, we focus on the scenario where the group action over $\mathcal{X}$ is unknown a priori and therefore needs to be inferred from data. Indeed, we assume that the dataset $\mathcal{D}$ consists of a finite number of samples from the group action i.e., triples of the form $(x, g, y) \in \mathcal{X} \times G \times \mathcal{X}$ with $y = g \cdot x$. The overall goal is learning a representation $\varphi : \mathcal{X} \to \mathcal{Z}$ which is (approximately) equivariant. Here, $\mathcal{Z}$ the latent space equipped with a group action chosen a priori. Given a parametrized class of functions φ_θ, $\theta \in \Theta$, such as deep neural networks,

the prototypical objective of equivariant representation learning is minimizing:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(x,g,y)\in\mathcal{D}} d(g \cdot \varphi_\theta(x),\ \varphi_\theta(x)). \tag{3.2}$$

In the above, d is a metric on $\mathcal{Z}$ or, more generally, a positive definite function. The objective defined by Equation 3.2 is similar to the one of metric learning (Equation 2.17). Indeed, both the representation learning frameworks aim to preserve geometric structure in the latent space – metrics and symmetries respectively. Note that while the group action over G is assumed to be unknown, the group G itself, together with the chosen latent action, is known and exploited for extracting the representation. In other words, the algebraic structure of (latent) symmetries behaves as a prior in equivariant representation learning. This information is often available in practice, as evident for example in the case of images of scenes (example from Section 1.1), where the group is simply $G = \mathbb{R}^3 \times \mathrm{SO}(3)$, independently from the scene considered.

Several works have explored variations of Equation 3.2 and of the choice of the latent group action. The first instance of equivariant representation learning dates back to the introduction of *Transforming Autoencoders* [68]. In this case, the latent space $\mathcal{Z} = G^d$ consists of a Cartesian product of several copies of G deemed 'capsules', on which the group acts via multiplication. Even though the original work focuses on $G = \mathbb{R}^2$ acting on images by translations of the visual plane, the principle is general and has been subsequently abstracted to arbitrary Lie groups [69]. As an alternative, a number of works have focused on Euclidean latent spaces $G = \mathbb{R}^d$ on which G acts by linear or affine transformations [70–72]. Even though linearity results in simple representations, it limits their expressivity since the original group action over $\mathcal{X}$ is rarely (isomorphic to) a linear one. Moreover, group convolutions (see Section 3.2) have been in some cases deployed as a latent group action [73,74]. Finally, we mention that is possible learn a representation with an objective analogous to Equation 3.2 without any prior knowledge around the group by training an additional transition model $T : \mathcal{Z} \times G \to \mathcal{Z}$ replacing the latent group action [75,76]. This, however, not only requires careful regularization in order to avoid trivial solutions (e.g., constant φ), but lacks the geometric properties and guarantees deriving from a group structure, which is crucial for our purposes.

In **Paper D**, we introduce a representation learner deemed *Equivariant Isomorphic Network*[1] (EquIN) which is general and theoretically grounded. Specifically, the foundational result is the following elementary yet profound fact from group theory.

Proposition 3.3.1. *Suppose that the action is* free *i.e., $g \cdot x = x$ implies $g = 1$. Then the following holds:*

[1] This nomenclature is introduced in Paper E, while the model is left unnamed elsewhere.

- *There is an equivariant isomorphism*

$$\mathcal{X} \simeq (\mathcal{X}/G) \times G, \tag{3.3}$$

where G acts trivially on the orbits and by multiplication on itself. In other words, each orbit can be identified equivariantly with the group itself.

- *Any equivariant map $\varphi : (\mathcal{X}/G) \times G \to (\mathcal{X}/G) \times G$ is a right multiplication on each orbit i.e., for each orbit $O \in \mathcal{X}/G$ there is an $h_O \in G$ such that $\varphi(O, g) = (O', gh_O)$ for all $g \in G$. In particular, if φ induces a bijection on orbits then it is an isomorphism.*

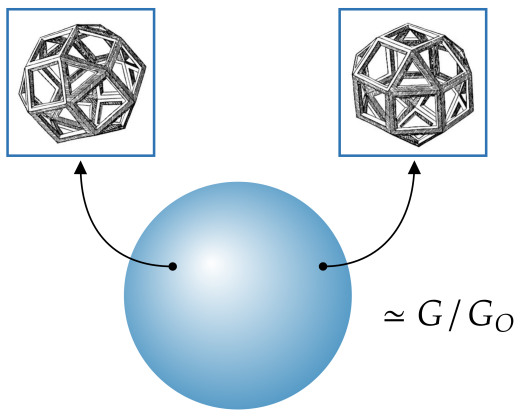


Figure 3.3: Each orbit O is isomorphic to the coset space G/G_O of the corresponding stabilizer subgroup. In particular, for free actions the orbit is isomorphic to the group G itself.

The above result not only describes $\mathcal{X}$ up to equivariant isomorphism, but enables to learn an isomorphic representation due to its second claim. Based on the latter, in EquIN we propose to set $\mathcal{Z} = \mathcal{E} \times G$, where $\mathcal{E}$ is a space responsible for representing orbit information while G is responsible for representing each orbit individually. Since orbits are invariant while G acts on itself via multiplication, given metrics $d_\mathcal{E}$ and d_G on $\mathcal{E}$ and G respectively, the objective from Equation 3.2 translates to:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(x,g,y)\in\mathcal{D}} \left(\underbrace{d_\mathcal{E}\left(\varphi^\mathcal{E}_\theta(y),\ \varphi^\mathcal{E}_\theta(x)\right)}_{Invariant} + \underbrace{d_G\left(\varphi^G_\theta(y),\ g\varphi^G_\theta(x)\right)}_{Multiplication-Equivariant} \right). \tag{3.4}$$

Here $\varphi^\mathcal{E}$ and φ^G denote the projections of φ to the corresponding latent Cartesian factors. Note that φ is effectively a contrastive learner (see Section 2.3), whose

positive pairs are data related by a symmetry or, in other words, belonging to the same orbit. Therefore, in order to encourage injectivity of $\varphi^{\mathcal{E}}$ – which is also required by the second claim of Proposition 3.3.1 – it is necessary to introduce an additional objective term such as the one from Equation 2.19.

The maps $\varphi^{\mathcal{E}}, \varphi^{G}$ are implemented as deep neural networks, and typically consist of two output branches of the same shared model. Since the output space of a neural network is Euclidean, a challenge is to design φ^{G} to produce elements of G, which can have non-trivial topology. We address this by assuming that G is a Lie group i.e., a differentiable manifold. This enables us to consider the *Lie algebra* $\mathfrak{g}$, defined as the tangent space at $1 \in G$. The Lie algebra maps to G via the *exponential map* $\mathfrak{g} \to G$, denoted by $v \mapsto e^{v}$. Although the exponential map can be defined for general Lie groups by an appropriate ordinary differential equation, for linear groups it is computed simply via the (Taylor-expanded) matrix exponential. Based on this, φ^{G} first outputs elements of $\mathfrak{g}$, that are then mapped to G via the exponential map. The same parametrization has been deployed in [77].

Proposition 3.3.1 requires the group action to be free. Even though this is common in practice, it may occur that group elements stabilize datapoints. For example, in the case of images depicting symmetric objects, rotations by specific angles produce (almost) identical data (see Figure 3.4). This leads to the following group-theoretical definition.

Definition 3.3.1. The *stabilizer* subgroup of a point $x \in \mathcal{X}$ w.r.t. an action by G on $\mathcal{X}$ is:

$$G_x = \{g \in G \mid g \cdot x = x\}. \tag{3.5}$$

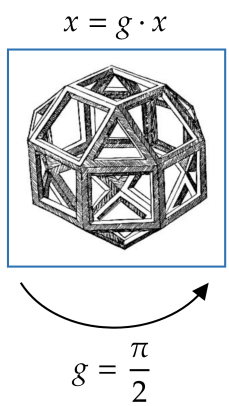


Figure 3.4: An example of a stabilizer.

Stabilizers of elements in the same orbit are conjugate, meaning that for each x, y belonging to the same orbit O there exists $h \in G$ such that $G_y = hG_xh^{-1}$. By

abuse of notation, we refer to the conjugacy class G_O of stabilizers for $O \in \mathcal{X}/G$. The notion of a free action can be rephrased as the condition that all the stabilizers are trivial, i.e., $G_O = \{1\}$ for every O. If the action over $\mathcal{Z}$ is not free while the one over $\mathcal{X}$ is, there exists no equivariant map $\varphi : \mathcal{X} \to \mathcal{Z}$. This raises the necessity of designing a more general equivariant representation learning framework in the presence of stabilizers. We address this in **Paper E** by extending EquIN to non-free actions. The core idea is to rely on the following generalization of Proposition 3.3.1.

Proposition 3.3.2. *The following holds:*

- *Each orbit O is isomorphic to the set of (left)* cosets $G/G_O = \{gG_O \mid g \in G\}$. *In other words, there is an isomorphism:*

$$\mathcal{X} \simeq \coprod_{O \in \mathcal{X}/G} G/G_O \quad \subseteq 2^G \times \mathcal{X}/G, \tag{3.6}$$

  where 2^G denotes the power-set of G on which G acts by left multiplication i.e., $g \cdot A = \{ga \mid a \in A\}$.

- *Any equivariant map*

$$\varphi : \mathcal{X} \to \coprod_{O \in \mathcal{X}/G} G/G_O \tag{3.7}$$

  inducing a bijection on orbits is an isomorphism.

This suggests a generalization of EquIN capable of taking stabilizers into account. Namely, instead of producing elements of G, φ_G is designed to output cosets of stabilizers. This extends EquIN to non-free actions while preserving all of its the theoretical guarantees. However, since the action is unknown a priori, such are the stabilizer subgroups. In order to circumvent the explicit modeling of the stabilizers, we propose to output arbitrary (finite) subsets of G. In other words, we replace the component G of $\mathcal{Z}$ in the original version of EquIN with the (finite) power-set 2^G. One can easily prove that if such subsets are *minimal*, they are guaranteed to coincide with cosets of stabilizers, as desired. We enforce minimality by an additional loss term penalizing (discrete) entropy. Finally, the metric d_G is chosen as a distance between sets e.g., the Chamfer or the Hausdorff distance over G. Overall, this describes our extension of EquIN to non-free group actions.

In the following sections, we draw both formal and informal connections between equivariant representation learning and related fields of statistics and robotics.

3.4 Relation to Disentanglement and Causality

Equivariance is closely related to the concept of *disentanglement* in representation learning. Intuitively, a representation is considered disentangled if a variation of a

distinguished semantic factor in data is reflected by a change of a single component in the latent space. For example, in the case of images depicting human portraits, a semantic aspect might be hair and skin color, or the azimuth angle of the head. The idea of disentanglement has been introduced informally in [4], followed by several attempts of formalizing the notion rigorously. A popular line of research initiated by [78] has proposed a statistical definition of disentanglement as the independence of the marginals of the distribution induced by data on the latent space via the representation. However, it has been shown that this notion is unsatisfactory since it makes disentanglement mathematically impossible in an unsupersived and unbiased manner [79]. The intuition is that statistical independence (of semantic factors) is not an *intrinsic* structure of data since it is not invariant under the symmetries of the distribution.

An alternative formulation of disentanglement based on symmetries and group theory has been addressed in [80, 81]. The presence of multiple semantic factors in the data is formalized as an action on $\mathcal{X}$ by a decomposed group

$$G = G_1 \times \cdots \times G_n, \tag{3.8}$$

where each of the Cartesian components G_i is responsible for the variation of a single factor. This leads to the following symmetry-based definition of disentanglement.

Definition 3.4.1. A representation $\varphi : \mathcal{X} \to \mathcal{Z}$ is *disentangled* if:

- There is a Cartesian decomposition $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_n$, where each $\mathcal{Z}_i$ is acted upon trivially by the factors G_j with $j \neq i$,
- φ is equivariant.

The definition above reduces the concept of disentanglement to the one of equivariance, corroborating the principle that preservation of symmetries leads to the extraction of meaningful geometric structure in data. The symmetry-based approach has led to frameworks for learning and evaluating disentangled representations [82, 83]. EquIN adheres to this line of research since it automatically infers disentangled representations in the sense of Definition 3.4.1. Indeed, assuming a group as in Equation 3.8 acts freely on $\mathcal{X}$, we set $\mathcal{Z}_i = G_i$. In order to account for the remaining latent factor $\mathcal{Z}_0 = \mathcal{E}$, a copy of the trivial group $G_0 = \{1\}$ can be added to G without altering it up to isomorphism. The group $G \simeq G_0 \times \cdots \times G_n$ acts on $\mathcal{Z} = \mathcal{E} \times G = \mathcal{Z}_0 \times \cdots \times \mathcal{Z}_n$ as required for a disentangled latent space. An analogous consideration applies to non-free actions, but requires the additional hypothesis that the stabilizer subgroups also factorize into a product of subgroups of the G_i's, similarly to Equation 3.8.

From a broader perspective, the relation between equivariant representation learning and disentanglement can be seen as an instance of a deeper analogy to *causal inference*. The latter is an area of statistics aiming to discover and exploit

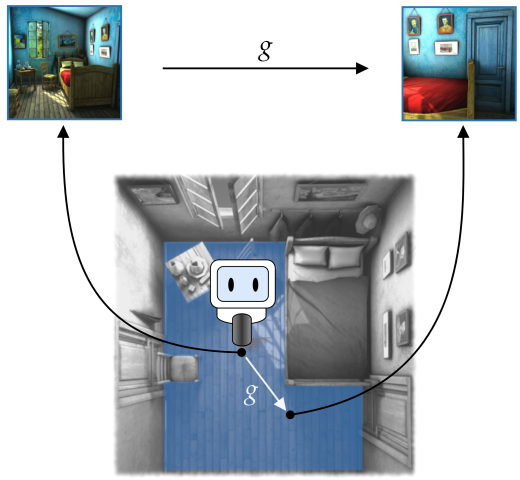


Figure 3.5: In several scenarios, actions performed by an agent define a group action on the space of perceived data.

the causal relations between factors in data. Classical principles from the field have seen extensions and deployment in the context of representation learning – a paradigm known as *causal representation learning* [84]. Now, the statistical formulation of disentanglement can be seen as an elementary instance of causality since a factorization into independent variables corresponds to a trivial causal graph with no edges. The reformulation of the notion of disentangled representations via symmetries and equivariance is analogous to the usage of *interventions* – a central tool for causal inference developed in Pearl's do-calculus [85]. The latter is grounded in the principle that interaction with the world is necessary in order to infer causal relations. Proposition 3.3.1 and 3.3.2 are parallel to *identifiability* results in causal representation learning [86, 87], which guarantee that the causal factors are inferrable by observing interventions and their consequences. The parallel between interventions and symmetries in the context of causal and equivariant representation learning respectively has been recently explored in the literature [87, 88].

3.5 Relation to Robotics and Interactive Perception

In this section, we draw connections between equivariant representation learning and robotic perception. To begin with, the focus of robotics are scenarios involving one or more agents (robots) that interact with the world, collect observations via sensing and, typically, aim to solve a specified task. The problem of extracting semantically-meaningful information from observations is known in robotics as *perception*. At its essence, the goal of the latter is enabling the agent(s) to understand the world, serving as a foundation for informed decision-making. Perception

is therefore the core component of intelligent behavior and, perhaps counterintuitively, is regarded as more challenging than reasoning itself – a principle originally formulated by Moravec in his celebrated paradox [89]. From a machine learning perspective, the problem of perception corresponds to the one of representation learning since both aim at extracting semantic structure from data. Now, the ability of interacting with the environment is crucial to the end of perception. Observing consequences of its own actions enables the agent to gather insights into the world and its structure. In particular, being able to predict the effects of interaction enables the agent to plan, which is essential to solve decision-making tasks. This principle has been explored in the robotics literature under the name of *interactive perception* [7]. The latter, similarly to the above-mentioned do-calculus, highlights the fundamental role of interaction for the purpose of semantic understanding.

The connection to equivariant representation learning is grounded in the fact that symmetries can be seen as a form of interaction. In this sense, the group action plays the role of the transition map for the environment's states and, therefore, of the observations perceived by the agent. For example, consider a locomotor robot that can navigate an environment by translating and rotating its camera along a single axis. The actions performed by the robot be seen as elements of the Lie group $\mathbb{R}^2 \times \mathrm{SO}(2)$, defining a group action on the space of images perceived by the robot while navigating – see Figure 3.5 for an illustration. In a sense, this point of view on symmetries is dual to the traditional one: they are seen as transformations of the agent's perspective as a consequence of its interaction with the environment. In this analogy, the group corresponds to the action space of the agent. The assumption from Section 3.3 that the group is known a priori is natural and practical since it is related to the agent itself, independently of the environment or the interaction between the two. Since a representation is an instance of a perception module, equivariant representation learning can be regarded as addressing the problem of interactive perception. Equivariance not only enables to predict the effect of interactions in the latent space, but extracts the intrinsic geometry of the world. This can be exploited by the agent to address specified tasks via simple geometric tools.

Motivated by this, in **Paper F** we explore applications to robotics of our equivariant representation learning framework described in Section 3.3. In particular, we deploy geometric principles from control theory in conjunction with the representation inferred by EquIN in a robotic scenario. The overall goal is *stabilizing* a given pre-trained policy by attracting it to in-distribution states whenever the agent deviates to out-of-distribution ones. In order to illustrate the approach, suppose for simplicity that the group of symmetries consists of translations i.e., $G = \mathbb{R}^n$. For $n = 2, 3$, this is the case for a manipulator robot interacting with the environment via its end-effector. Consider a policy $\pi : \mathcal{X} \to \mathbb{R}^n$ associating such translations to the agent's observations. We assume that π has been trained on a dataset $P \subseteq \mathcal{X}$ to solve a given task, which is typically achieved via methods from offline reinforcement learning [90]. Therefore, π is expected to perform well on states close to P

(referred to as 'in-distribution'), but to behave unexpectedly and to potentially fail away from P (referred to as 'out-of-distribution'). In order to redirect the agent back to in-distribution states when the latter happens, we propose to introduce a *recovery policy* given by:

$$\widetilde{\pi}(x) = \nabla_z \rho(\varphi(x)). \tag{3.9}$$

Here, $\varphi : \mathcal{X} \to \mathbb{R}^n$ is a pre-trained EquIN model mapping observations to the corresponding orbit, while ρ is a density estimator over P. Intuitively, ρ represents an energy function, attracting the agent to the data P the original policy was trained on. Note that in order for the density estimation to be meaningful, φ has to be trained on task-independent data encompassing a broader range of states. Even though this potentially requires a more extensive data collection in practice, it makes the representation, together with the recovery policy $\widetilde{\pi}$, transferable across different tasks and policies π. In order to deal with multiple orbits, we condition the density estimator ρ to the orbit of x. The latter is conveyed by some contextual information, for example the initial state the agent observes when performing a rollout. When ρ is a Gaussian Mixture Model, the conditioning is conveniently implemented via a neural network, obtaining a model known as a Mixture Density Model [91]. Lastly, the recovery policy is combined with the original into a mixture policy given by:

$$\sigma(\rho(x))\ \pi(x) + (1 - \sigma(\rho(x)))\ \widetilde{\pi}(x), \tag{3.10}$$

where $\sigma(u) = (1 + e^{-u})^{-1}$ is a sigmoid function normalizing the density estimation in order to softly determine which one among the two policies to follow. Overall, our approach showcases how the geometry extracted via EquIN can be exploited to address control problems simply and effectively.

3.6 Harmonic Analysis, Invariant Networks and Symmetry Discovery

Symmetries and groups are closely related to harmonic analysis. Indeed, the Fourier transform – together with its core properties – can be abstracted to general groups. The Fourier transform has seen application at a fundamental level in several areas, both in its classical and group-theoretical version. These applications include signal processing – in particular computer vision and image compression – and computational algebra. Historically, the deployment of the Fourier transform and convolutional filters in computer vision has motivated the initial introduction of convolutions into deep learning, as discussed in Section 3.2. In what follows, we briefly overview the principles of group-theoretical harmonic analysis, focusing on finite groups. For a comprehensive treatment, see [60]. In particular, similarly to Section 3.2, we consider the complex vector space $\mathcal{X} = \mathbb{C}^G$ freely generated by elements of a finite group G. This space consists of (complex-valued) signals over G, and carries the structure necessary for the Fourier transform to be defined.

To begin with, consider an Abelian group G. In this case, harmonics are abstracted as homomorphisms $G \to \mathrm{U}(1)$ valued in unitary complex scalar, leading to the following definition.

Definition 3.6.1. The *dual* $G^\vee$ of G is the set of homomorphisms $\rho : G \to \mathrm{U}(1)$. It is itself a group when equipped with the point-wise composition $\rho\mu = \rho \odot \mu$ or, explicitly, $(\rho\mu)(g) = \rho(g)\mu(g)$ for $g \in G$.

When $\mathbb{C}^G$ is endowed with the canonical scalar product $\langle x, y\rangle = \sum_{g\in G} \overline{x_g} y_g$, $G^\vee \subseteq \mathbb{C}^G$ forms an orthogonal basis with all the norms equal to $|G|$. The linear base-change is, by definition, the Fourier transform over $\mathbb{C}^G$.

Definition 3.6.2. The *Fourier transform* is the map $\mathbb{C}^G \to \mathbb{C}^{G^\vee}$, $x \mapsto \hat{x}$, defined for $\rho \in G^\vee$ as:

$$\hat{x}_\rho = \langle \rho, x\rangle. \tag{3.11}$$

The Fourier transform is a linear isometry (i.e., a unitary operator) up to a multiplicative constant of $|G|$ and it exchanges the convolution product $\star$ over $\mathbb{C}^G$ (see Equation 3.1) with the Hadamard product $\odot$ over $\mathbb{C}^{G^\vee}$. Definition 3.6.2 generalizes the usual (discrete) Fourier transform in the following sense. If $G = \mathbb{Z}/N\mathbb{Z}$ is the group of modular integers, then the dual $G^\vee$ consists of homomorphisms of the form

$$\mathbb{Z}/N \ni g \mapsto e^{2\pi\sqrt{-1}gk/N} \tag{3.12}$$

for $k \in \{0, \cdots, N-1\}$. Equation 3.11 specializes then to the familiar definition of the Fourier transform.

All the construction above can be extended to non-Abelian groups, but require more advanced technical tools. In a nutshell, harmonics are generalized to 'matrix-valued' maps i.e., *unitary representations*. The latter are defined as homomorphisms $\rho_V : G \to \mathrm{U}(V)$, where V is finite-dimensional complex Hilbert space. Unitary representations that are irreducible w.r.t. direct sum decompositions form a finite set $\mathrm{Irr}(G)$, and the Fourier transform is a map $\mathbb{C}^G \to \bigoplus_{\rho_V \in \mathrm{Irr}(G)} \mathrm{End}(V)$ defined analogously to Equation 3.11. Here, $\mathrm{End}(V)$ denotes the space of linear operators over V.

Strikingly, harmonics have been observed to emerge in both biological and artificial neural networks. More specifically, the synaptic response of neurons known as *simple cells* in the primary visual cortex of the brain can be modelled via Gabor wavelets [92] – a localized version of planar harmonics. Similarly, the filters learned by convolutional neural networks on image classification tasks resemble Gabor wavelets and harmonics [93]. This suggests a deep connection between machine learning and harmonic analysis that is consistent throughout a variety of models and scenarios. One mathematical explanation to this phenomenon has been attributed to the statistics of data. Assuming that the data manifold (e.g., natural images) is

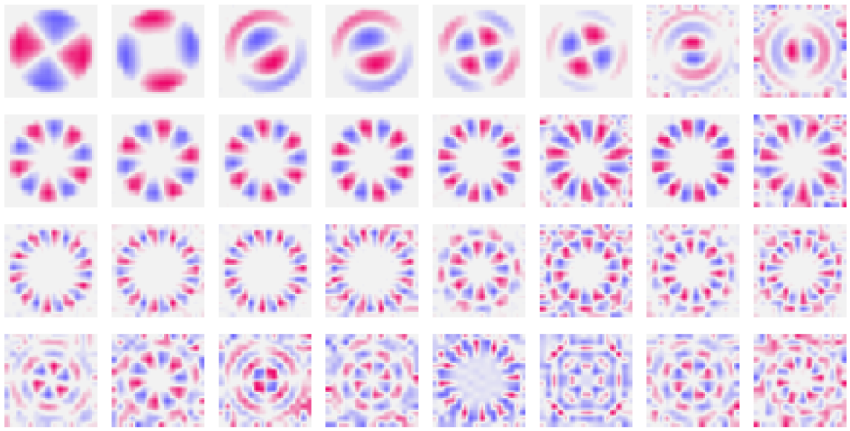


Figure 3.6: Weights learned by a rotation-invariant neural network resemble circular harmonics.

concentrated around a given vector basis (e.g., the Gabor/Fourier basis), a linear generative model can recover the latter unsupervisedly via a sparsity prior [94, 95]. This however not only does not explain the emergence of harmonics beyond the generative setting – such as image classification tasks – but in turn relies on an unexplained and specific assumption around the statistics of data. In **Paper G**, we offer an alternative explanation to the emergence of harmonics in the weights of neural networks based on an algebraic group-theoretical argument. Specifically, we prove that if a machine learning model of a certain kind is invariant to the action by G on $\mathbb{C}^G$, then its weights (i.e., the learner's parameters) coincide with the group-theoretical Fourier transform of G (up to appropriate adjustments). We therefore argue that the Fourier transform emerges in both biological and artificial learning systems due to invariance, which is a natural bias that can be induced implicitly from data or encouraged explicitly via contrastive learning (see Section 2.3).

We now outline the main result. To this end, let $V_1, \cdots, V_k$ be finite-dimensional Hilbert spaces and consider the space:

$$\Theta = \mathbb{C}^G \otimes \bigoplus_i \mathrm{End}(V_i). \tag{3.13}$$

The above represents the space of parameters of a complex-valued machine learning model. Specifically, $\mathbb{C}^G$ corresponds to the input space, while each $\mathrm{End}(V_i)$ represents a computational unit processing matrix-valued signals. The latter are commonly referred to as *capsules* [96] and will be necessary for the emergence of the Fourier transform in the non-Abelian case. The space Θ carries actions by G – affecting the left tensor factor – and by $\mathrm{U}(V_i)$ for each i – affecting the direct summands. In order to prove our result, we introduce the following abstract prop-

erty for learners. In order to prove our result, we introduce the following abstract property. Let $\mathcal{H}$ be a set.

Definition 3.6.3. We say that a map $\varphi\colon \Theta \to \mathcal{H}$ has *unitary symmetries* if for $\theta, \theta' \in \Theta$ of the same norm, $\varphi(\theta) = \varphi(\theta')$ implies that for each i there exists a unitary operator $U_i \in \mathrm{U}(V_i)$ such that $\theta_i = U_i \cdot \theta'_i$.

In the above, φ represents a machine learning model with parameter space Θ while $\mathcal{H}$ represents the space of functions parametrized by the learner – typically referred to as 'hypothesis space'. Intuitively, Definition 3.6.3 requires that only parameters related by unitary symmetries can correspond to the same hypothesis. We show that several common learners have unitary symmetries, including Spectral Networks [97], McCulloch-Pitts neurons and, to some extent, deep neural networks. The main result in **Paper G** is the following.

Theorem 3.6.1. *Suppose that $\varphi\colon \Theta \to \mathcal{H}$ has unitary symmetries and that for some $\theta \in \Theta$ the following holds:*

- *$\varphi(g \cdot \theta) = \varphi(\theta)$ for all $g \in G$.*
- *θ, seen as a linear map $\langle G \rangle \to \bigoplus_i \mathrm{End}(V_i)$, is surjective.*

Then for every i there exist $\theta'_i \in \mathrm{End}(V_i)$ and an irreducible unitary representation $\rho_i : G \to \mathrm{U}(V_i)$ such that for all $g \in G$,

$$\theta_i(g) = \theta'_i \, \rho_i(g)^\dagger. \tag{3.14}$$

When $\mathcal{H}$ is a space of functions with domain $\mathbb{C}^G$ and φ satisfies an adjunction property, the first assumption of Theorem 3.6.1 translates into invariance to G in the input space $\mathbb{C}^G$. With an additional orthogonality condition for θ, the full Fourier transform can be recovered. Overall, Theorem 3.6.1 shows how unitary representations, i.e. harmonics in a broader sense, arise from invariance to a given arbitrary finite group.

Since the Fourier transform encodes all the algebraic information around G, our results enable to recover the group structure (up to isomorphism) from the parameters of an invariant model with unitary symmetries. Even further, we show that this procedure is robust. Specifically, the group recovery guarantees hold even when the invariance assumption is relaxed according to certain explicit functional bounds. Extracting the group structure offers a tool for *symmetry discovery* – a problem consisting of inferring symmetries of data with minimal supervision [98,99]. Discovering symmetries is a form of structure extraction that enables to perform inference in cases when the group G is unknown a priori. The latter is a crucial assumption of all the symmetry-based approaches previously discussed in this thesis. The results in **Paper G** represent a first step to address these limitations, albeit within specific assumptions and context.

Chapter 4

Conclusions, Limitations and Future Work

I do not understand the definition of a Gorenstein ring.
–Daniel Gorenstein

4.1 Conclusions

In this thesis, we have addressed the problem of extracting geometry from data and deployed it for statistics and machine learning. We discussed methods leveraging on symmetries and metrics – two fundamental objects arising in both continuous and discrete geometry. From the metric side, we have discussed novel non-parametric methods based on the Voronoi tessellations and Delaunay triangulations. These include an active learning version of the nearest neighbor regressor as well as two Voronoi density estimators. We have formally shown how the adaptive geometry of Voronoi cells implies convergence guarantees for the proposed methods. On the symmetry side, we have focused on equivariant methods in representation learning. We have proposed a general equivariant representation learning framework suitable for data acted upon arbitrary (Lie) groups. The method is guaranteed to infer isomorphic representations, even in the challenging scenario when symmetries can stabilize data. We have additionally explored applications of the resulting representations to robotics. Lastly, we have analyzed theoretically the problem of symmetry discovery, showing that a class of learners, if invariant, extracts the Fourier transform together with the group structure of the underlying symmetries.

4.2 Limitations and Future Work

The material presented in this thesis is subject to a number of limitations and leaves questions open, tracing directions for future investigation.

Regarding metric-based approaches, **Papers A, B** and **C** focus on non-parametric methods. Although, as discussed in Section 2, this avoids the computational burden inherent in optimization, parametric methods can be advantageous because of their flexibility and expressivity. As a line of future investigation, we therefore envision extensions to the parametric setting. Since optimization is typically performed via gradient descent, a core challenge lies in designing differentiable versions of these methods and computing their derivatives. For Voronoi-based density estimators such as our CVDE and RVDE, this extension would be analogous to how Mixture Models generalize KDE by optimizing the parameters of the kernel – in particular its centers P. Note that differentiating w.r.t. P is particularly challenging since it involves deriving geometric quantities i.e., Voronoi tessellations. As suggested in [100], this can be addressed by appealing to the Reynolds transport theorem, but is affected by geometric singularities such as multiple points of P overlapping. On a similar note, designing density estimators which are differentiable w.r.t. the ambient point x is interesting and useful in situations requiring optimizing the density itself and/or performing its gradient. For example, this is crucial in the construction of the recovery policy from **Paper F** (see Equation 3.9). Note that even though RVDE is continuous w.r.t. x, it is inherently singular at the boundaries of Voronoi cells, which represents a major limitation from this perspective. Constructing a differentiable – and even smooth – Voronoi-based density estimator is a challenge which would require an approach alternative to those involved in defining CVDE and RVDE.

Another aspect of the metric-based methods discussed in this thesis which is worth exploring is their extension to more general metric spaces than the Euclidean one – in particular Riemannian manifolds. This is interesting since non-Euclidean manifolds often arise in statistics and machine learning. For example, data on spheres are the object of study of directional statistics [101], hyperbolic spaces are routinely deployed to represent hierarchical data [47], Lie groups are central for equivariant representation learning and pose estimation as discussed in Section 3.3, and even complex projective spaces arise as Kendall shape spaces in computer vision [102]. Even though Voronoi tessellations are defined and well-behaved on Riemannian manifolds, most of the techniques discussed in this thesis rely on Euclidean geometry, in particular the proofs of theoretical results such as Theorem 2.2.1. These would require a careful extension to the general Riemannian setting. Moreover, the ray-casting techniques involved in ANNR, CVDE and RVDE would need to be adapted by deploying the Riemannian *exponential map* and by taking the curvature of the manifold into account.

Regarding symmetry-based approaches, we highlight a limitation arising from the group-theoretical formulation. In order to illustrate this, consider the concrete setting from Section 3.5 where an agent interacts with an environment, with the group of symmetries corresponding to the agent's action space. Assume that the

environment is dynamic to some extent i.e., it can change (part of) its state as a consequence of interaction. For example, this happens when a robot collides with a physical object, displacing the latter. In this case, the transition map of (the agent's observations of) the environment is not a group action. Indeed, suppose that a mobile robot collides with an object while performing a path $g_1, \cdots, g_n \in G$ starting from $x \in \mathcal{X}$. If the robot can reach the same position while avoiding the collision by performing another path, say $h_1, \cdots, h_k \in G$, then it holds that $g_1 \cdots g_n = h_1 \cdots h_n$ but $(g_1 \cdots g_n) \cdot x \neq (h_1 \cdots h_k) \cdot x$. An algebraic interpretation of this phenomenon is that the associativity condition from Definition 3.1.1 does not hold since the paths $g_\bullet$, $h_\bullet$ do not coincide. This raises the need of generalizing the theory and methods from Section 3.3 in order to accommodate dynamic environments. A first step in this direction has been made in **Paper X-3** (not included in this thesis), where we consider generalized equivariant representations in the context of interactions with rigid physical objects and prove an analogue of Proposition 3.3.1 in this context. This approach is however limited to the specific dynamics considered. In order to address the problem in general, the above discussion suggests that a natural possibility is replacing the group G by the space of its paths, which acts on $\mathcal{X}$. This is however an intractable infinite-dimensional space whose action is non-free since any closed path that does not affect the environment is a stabilizer. On the high level, this scenario is reminiscent of the *monodromy action* of the fundamental groupoid (i.e., the path space up to homotopy) of a topological space over its coverings [103]. Monodromy might provide both inspiration and tools to address the problem, drawing connections between homotopy theory and interactive perception. This however falls beyond the scope of this thesis, and we leave it as a line for future investigation.

Chapter 5

Summary of Included Papers

This chapter contains the abstracts of the included papers and the contributions by the author of this thesis. The symbol * denotes equal contribution.

Paper A

Active Nearest Neighbor Regression Through Delaunay Refinement

A. Kravberg*, **G. L. Marchetti***, V. Polianskii*, A. Varava,
F. T. Pokorny, D. Kragic.
In *International Conference on Machine Learning* (ICML), 2022.

Abstract: We introduce an algorithm for active function approximation based on nearest neighbor regression. Our Active Nearest Neighbor Regressor (ANNR) relies on the Voronoi-Delaunay framework from computational geometry to subdivide the space into cells with constant estimated function value and select novel query points in a way that takes the geometry of the function graph into account. We consider the recent state-of-the-art active function approximator called DEFER, which is based on incremental rectangular partitioning of the space, as the main baseline. The ANNR addresses a number of limitations that arise from the space subdivision strategy used in DEFER. We provide a computationally efficient implementation of our method, as well as theoretical halting guarantees. Empirical results show that ANNR outperforms the baseline for both closed-form functions and real-world examples, such as gravitational wave parameter inference and exploration of the latent space of a generative model.

Contributions by the author: co-designed the method, provided the mathematical formulation and the theoretical analysis, designed and implemented the deep learning experiment, wrote the majority of the paper (excluding parts of the experimental section and of the introduction).

Paper B

Voronoi Density Estimator for High-Dimensional Data: Computation, Compactification and Convergence

V. Polianskii*, **G. L. Marchetti***, A. Kravberg, A. Varava,
F. T. Pokorny, D. Kragic.
In *Uncertainty in Artificial Intelligence* (UAI), 2022.

Abstract: The Voronoi Density Estimator (VDE) is an established density estimation technique that adapts to the local geometry of data. However, its applicability has been so far limited to problems in two and three dimensions. This is because Voronoi cells rapidly increase in complexity as dimensions grow, making the necessary explicit computations infeasible. We define a variant of the VDE deemed Compactified Voronoi Density Estimator (CVDE), suitable for higher dimensions. We propose computationally efficient algorithms for numerical approximation of the CVDE and formally prove convergence of the estimated density to the original one. We implement and empirically validate the CVDE through a comparison with the Kernel Density Estimator (KDE). Our results indicate that the CVDE and the KDE are comparable at their best performance and that the CVDE surpasses the KDE under arbitrary bandwidth selection.

Contributions by the author: co-designed the method, provided the mathematical formulation and the theoretical analysis, co-designed the experiments, wrote the majority of the paper (excluding parts of the experimental section and of the computational section).

Paper C

An Efficient and Continuous Voronoi Density Estimator

G. L. Marchetti, V. Polianskii, A. Varava, F. T. Pokorny, D. Kragic.
In *International Conference on Artificial Intelligence and Statistics* (AISTATS), 2023.

Abstract: We introduce a non-parametric density estimator deemed Radial Voronoi Density Estimator (RVDE). RVDE is grounded in the geometry of Voronoi tessellations and as such benefits from local geometric adaptiveness and broad convergence properties. Due to its radial definition RVDE is continuous and computable in linear time with respect to the dataset size. This amends for the main shortcomings of previously studied VDEs, which are highly discontinuous and computationally expensive. We provide a theoretical study of the modes of RVDE as well as an empirical investigation of its performance on high-dimensional data. Results show that RVDE outperforms other non-parametric density estimators, including recently introduced VDEs.

Contributions by the author: designed the method and its implementation, provided the mathematical formulation and the theoretical analysis, designed the experiments, wrote the entire paper.

Paper D

Equivariant Representation Learning via Class-Pose Decomposition

G. L. Marchetti*, G. Tegnér*, A. Varava, D. Kragic.
In *International Conference on Artificial Intelligence and Statistics* (AISTATS), 2023.

Abstract: We introduce a general method for learning representations that are equivariant to symmetries of data. Our central idea is to decompose the latent space into an invariant factor and the symmetry group itself. The components semantically correspond to intrinsic data classes and poses respectively. The learner is trained on a loss encouraging equivariance based on supervision from relative symmetry information. The approach is motivated by theoretical results from group theory and guarantees representations that are lossless, interpretable and disentangled. We provide an empirical investigation via experiments involving datasets with a variety of symmetries. Results show that our representations capture the geometry of data and outperform other equivariant representation learning frameworks.

Contributions by the author: designed the method and its implementation, provided the mathematical formulation and the theoretical analysis, co-designed the experiments, wrote the majority of the paper (excluding parts of the experimental section and of the introduction).

Paper E

Equivariant Representation Learning in the Presence of Stabilizers

L. A. P. Rey*, **G. L. Marchetti***, D. Kragic, D. Jarnikov, M. Holenderski.
In *European Conference on Machine Learning* (ECML-PKDD), 2023.

Abstract: We introduce Equivariant Isomorphic Networks (EquIN) – a method for learning representations that are equivariant with respect to general group actions over data. Differently from existing equivariant representation learners, EquIN is suitable for group actions that are not free, i.e., that stabilize data via nontrivial symmetries. EquIN is theoretically grounded in the orbit-stabilizer theorem from group theory. This guarantees that an ideal learner infers isomorphic representations while trained on equivariance alone and thus fully extracts the geometric structure of data. We provide an empirical investigation on image datasets with rotational symmetries and show that taking stabilizers into account improves the quality of the representations.

Contributions by the author: co-designed the method, provided the mathematical formulation, co-designed the experiments and implemented the model, wrote the majority of the paper (excluding parts of the experimental section).

Paper F

Back to the Manifold: Recovering from Out-of-Distribution States

A. Reichlin, **G. L. Marchetti**, H. Yin, A. Ghadirzadeh, D. Kragic.
In *International Conference on Intelligent Robots and Systems* (IROS), 2022.

Abstract: Learning from previously collected datasets of expert data offers the promise of acquiring robotic policies without unsafe and costly online explorations. However, a major challenge is a distributional shift between the states in the training dataset and the ones visited by the learned policy at the test time. While prior works mainly studied the distribution shift caused by the policy during the offline training, the problem of recovering from out-of-distribution states at the deployment time is not very well studied yet. We alleviate the distributional shift at the deployment time by introducing a recovery policy that brings the agent back to the training manifold whenever it steps out of the in-distribution states, e.g., due to an external perturbation. The recovery policy relies on an approximation of the training data density and a learned equivariant mapping that maps visual observations into a latent space in which translations correspond to the robot actions. We demonstrate the effectiveness of the proposed method through several manipulation experiments on a real robotic platform. Our results show that the recovery policy enables the agent to complete tasks while the behavioral cloning alone fails because of the distributional shift problem.

Contributions by the author: co-designed the method and its implementation, provided the mathematical formulation, wrote parts of the paper (mainly the methodological section).

Paper G

Harmonics of Learning: Universal Fourier Features Emerge in Invariant Networks

G. L. Marchetti, C. Hillar, D. Kragic, S. Sanborn.
Preprint. Available on ArXiv: https://arxiv.org/abs/2312.08550.

Abstract: In this work, we formally prove that, under certain conditions, if a neural network is invariant to a finite group then its weights recover the Fourier transform on that group. This provides a mathematical explanation for the emergence of Fourier features – a ubiquitous phenomenon in both biological and artificial learning systems. The results hold even for non-commutative groups, in which case the Fourier transform encodes all the irreducible unitary group representations. Our findings have consequences for the problem of symmetry discovery. Specifically, we demonstrate that the algebraic structure of an unknown group can be recovered from the weights of a network that is at least approximately invariant within certain bounds. Overall, this work contributes to a foundation for an algebraic learning theory of invariant neural network representations.

Contributions by the author: provided the mathematical formulation and the theoretical analysis, designed and implemented the experiments, wrote the majority of the paper (excluding parts of the introduction).

Bibliography

[1] D. Abramovich and A. Polishchuk, "Sheaves of t-structures and valuative criteria for stable complexes," 2006.

[2] A. N. Gorban and I. Y. Tyukin, "Blessing of dimensionality: mathematical foundations of the statistical physics of data," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2118, p. 20170237, 2018.

[3] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," *arXiv preprint arXiv:2104.13478*, 2021.

[4] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[5] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[6] H. Edelsbrunner and J. L. Harer, *Computational topology: an introduction*. American Mathematical Society, 2022.

[7] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.

[8] T. L. Heath *et al.*, *The thirteen books of Euclid's Elements*. Courier Corporation, 1956.

[9] F. Klein, "A comparative review of recent researches in geometry," *Bulletin of the American Mathematical Society*, vol. 2, no. 10, pp. 215–249, 1893.

[10] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *International conference on machine learning*, pp. 2990–2999, PMLR, 2016.

[11] A. Kravberg, G. L. Marchetti, V. Polianskii, A. Varava, F. T. Pokorny, and D. Kragic, "Active nearest neighbor regression through delaunay refinement," in *International Conference on Machine Learning*, pp. 11650–11664, PMLR, 2022.

[12] V. Polianskii, G. L. Marchetti, A. Kravberg, A. Varava, F. T. Pokorny, and D. Kragic, "Voronoi density estimator for high-dimensional data: Computation, compactification and convergence," in *Uncertainty in Artificial Intelligence*, pp. 1644–1653, PMLR, 2022.

[13] G. L. Marchetti, V. Polianskii, A. Varava, F. T. Pokorny, and D. Kragic, "An efficient and continuous voronoi density estimator," in *International Conference on Artificial Intelligence and Statistics*, pp. 4732–4744, PMLR, 2023.

[14] G. L. Marchetti, G. Tegnér, A. Varava, and D. Kragic, "Equivariant representation learning via class-pose decomposition," in *International Conference on Artificial Intelligence and Statistics*, pp. 4745–4756, PMLR, 2023.

[15] L. A. Pérez Rey, G. L. Marchetti, D. Kragic, D. Jarnikov, and M. Holenderski, "Equivariant representation learning in the presence of stabilizers," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 693–708, Springer, 2023.

[16] A. Reichlin, G. L. Marchetti, H. Yin, A. Ghadirzadeh, and D. Kragic, "Back to the manifold: Recovering from out-of-distribution states," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8660–8666, IEEE, 2022.

[17] G. L. Marchetti, C. Hillar, D. Kragic, and S. Sanborn, "Harmonics of learning: Universal fourier features emerge in invariant networks," *arXiv preprint arXiv:2312.08550*, 2023.

[18] J. Leray, "L'anneau spectral et l'anneau filtré d'homologie d'un espace localement compact et d'une application continue," *Cours professés au collège de France*, 1950.

[19] G. Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. premier mémoire. sur quelques propriétés des formes quadratiques positives parfaites.," *Journal für die reine und angewandte Mathematik (Crelles Journal)*, vol. 1908, no. 133, pp. 97–102, 1908.

[20] R. Descartes, *Principia philosophiae*. Apud Danielem Elsevirium, 1644.

[21] G. Lejeune Dirichlet, "Über die reduction der positiven quadratischen formen mit drei unbestimmten ganzen zahlen.," *Journal für die reine und angewandte Mathematik (Crelles Journal)*, vol. 1850, no. 40, pp. 209–227, 1850.

[22] E. K. Donald *et al.*, "The art of computer programming," *Sorting and searching*, vol. 3, pp. 426–458, 1999.

[23] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[24] B. Delaunay *et al.*, "Sur la sphere vide," *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, vol. 7, no. 793-800, pp. 1–2, 1934.

[25] S. M. Omohundro, *The Delaunay triangulation and function learning*. International Computer Science Institute, 1989.

[26] L. Chen and J.-c. Xu, "Optimal delaunay triangulations," *Journal of Computational Mathematics*, pp. 299–308, 2004.

[27] J. Ruppert, "A delaunay refinement algorithm for quality 2-dimensional mesh generation," *Journal of algorithms*, vol. 18, no. 3, pp. 548–585, 1995.

[28] L. P. Chew, "Guaranteed-quality mesh generation for curved surfaces," in *Proceedings of the ninth annual symposium on Computational geometry*, pp. 274–280, 1993.

[29] P. McMullen, "The maximum numbers of faces of a convex polytope," *Mathematika*, vol. 17, no. 2, pp. 179–184, 1970.

[30] H. Edelsbrunner and R. Seidel, "Voronoi diagrams and arrangements," in *Proceedings of the first annual symposium on Computational geometry*, pp. 251–262, 1985.

[31] C. E. Buckley, "A divide-and-conquer algorithm for computing 4-dimensional convex hulls," in *Workshop on Computational Geometry*, pp. 113–135, Springer, 1988.

[32] V. Polianskii and F. T. Pokorny, "Voronoi graph traversal in high dimensions with applications to topological data analysis and piecewise linear interpolation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2154–2164, 2020.

[33] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[34] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, pp. 109–165, Elsevier, 1989.

[35] D. M. Y. Sommerville, *An Introduction to the Geometry of n Dimensions*, vol. 512. Dover New York, 1958.

[36] K. Pearson, "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894.

[37] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The annals of mathematical statistics*, pp. 832–837, 1956.

[38] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[39] J. Marron, "A comparison of cross-validation techniques in density estimation," *The Annals of Statistics*, pp. 152–162, 1987.

[40] M. P. Wand, M. C. Jones, *et al.*, "Multivariate plug-in bandwidth selection," *Computational Statistics*, vol. 9, no. 2, pp. 97–116, 1994.

[41] D. Freedman and P. Diaconis, "On the histogram as a density estimator: L 2 theory," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 57, no. 4, pp. 453–476, 1981.

[42] J. K. Ord, "How many trees in a forest," *Mathematical Scientist*, vol. 3, pp. 23–33, 1978.

[43] M. E. Dyer and A. M. Frieze, "On the complexity of computing the volume of a polyhedron," *SIAM Journal on Computing*, vol. 17, no. 5, pp. 967–974, 1988.

[44] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on computers*, vol. 100, no. 5, pp. 401–409, 1969.

[45] K. Q. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *Advances in neural information processing systems*, vol. 18, 2005.

[46] M. Kaya and H. Ş. Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, no. 9, p. 1066, 2019.

[47] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," *Advances in neural information processing systems*, vol. 30, 2017.

[48] Y. Bi, B. Fan, and F. Wu, "Beyond mahalanobis metric: cayley-klein metric learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2339–2347, 2015.

[49] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[50] J. Teenbaum, D. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[51] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive representation learning: A framework and review," *Ieee Access*, vol. 8, pp. 193907–193934, 2020.

[52] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, 1993.

[53] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 2, pp. 1735–1742, IEEE, 2006.

[54] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[55] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, no. 0, 2006.

[56] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning*, pp. 9929–9939, PMLR, 2020.

[57] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21271–21284, 2020.

[58] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[59] S. Mac Lane, *Categories for the working mathematician*, vol. 5. Springer Science & Business Media, 2013.

[60] G. B. Folland, *A course in abstract harmonic analysis*, vol. 29. CRC press, 2016.

[61] T. S. Cohen, M. Geiger, and M. Weiler, "A general theory of equivariant cnns on homogeneous spaces," *Advances in neural information processing systems*, vol. 32, 2019.

[62] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in neural information processing systems*, vol. 2, 1989.

[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[64] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical cnns," *arXiv preprint arXiv:1801.10130*, 2018.

[65] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[67] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[68] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, pp. 44–51, Springer, 2011.

[69] J. E. Lenssen, M. Fey, and P. Libuschewski, "Group equivariant capsule networks," *Advances in neural information processing systems*, vol. 31, 2018.

[70] X. Guo, E. Zhu, X. Liu, and J. Yin, "Affine equivariant autoencoder.," in *IJCAI*, pp. 2413–2419, 2019.

[71] R. Quessard, T. Barrett, and W. Clements, "Learning disentangled representations and group structure of dynamical environments," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19727–19737, 2020.

[72] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Interpretable transformations with encoder-decoder networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5726–5735, 2017.

[73] E. Dupont, M. B. Martin, A. Colburn, A. Sankar, J. Susskind, and Q. Shan, "Equivariant neural rendering," in *International Conference on Machine Learning*, pp. 2761–2770, PMLR, 2020.

[74] D. Worrall and G. Brostow, "Cubenet: Equivariance to 3d rotation and translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 567–584, 2018.

[75] E. Van der Pol, T. Kipf, F. A. Oliehoek, and M. Welling, "Plannable approximations to mdp homomorphisms: Equivariance under actions," *arXiv preprint arXiv:2002.11963*, 2020.

[76] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," *Advances in neural information processing systems*, vol. 28, 2015.

[77] A. K. Mondal, V. Jain, K. Siddiqi, and S. Ravanbakhsh, "Eqr: Equivariant representations for data-efficient reinforcement learning," in *International Conference on Machine Learning*, pp. 15908–15926, PMLR, 2022.

[78] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *International conference on learning representations*, 2016.

[79] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *international conference on machine learning*, pp. 4114–4124, PMLR, 2019.

[80] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *arXiv preprint arXiv:1812.02230*, 2018.

[81] I. Higgins, S. Racanière, and D. Rezende, "Symmetry-based representations for artificial and biological general intelligence," *Frontiers in Computational Neuroscience*, vol. 16, p. 836498, 2022.

[82] L. Tonnaer, L. A. P. Rey, V. Menkovski, M. Holenderski, and J. W. Portegies, "Quantifying and learning linear symmetry-based disentanglement," *arXiv preprint arXiv:2011.06070*, 2020.

[83] H. Caselles-Dupré, M. Garcia Ortiz, and D. Filliat, "Symmetry-based disentangled representation learning requires interaction with environments," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[84] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[85] J. Pearl and D. Mackenzie, *The book of why: the new science of cause and effect*. Basic books, 2018.

[86] J. Brehmer, P. De Haan, P. Lippe, and T. S. Cohen, "Weakly supervised causal representation learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38319–38331, 2022.

[87] K. Ahuja, J. Hartford, and Y. Bengio, "Properties from mechanisms: an equivariance perspective on identifiable representation learning," *arXiv preprint arXiv:2110.15796*, 2021.

[88] M. Besserve and B. Schölkopf, "Learning soft interventions in complex equilibrium systems," in *Uncertainty in Artificial Intelligence*, pp. 170–180, PMLR, 2022.

[89] H. Moravec, *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.

[90] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review," *and Perspectives on Open Problems*, vol. 5, 2020.

[91] C. M. Bishop, "Mixture density networks," 1994.

[92] J. J. Kulikowski, S. Marčelja, and P. O. Bishop, "Theory of spatial position and spatial frequency relations in the receptive fields of simple cells in the visual cortex," *Biological cybernetics*, vol. 43, no. 3, pp. 187–198, 1982.

[93] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter, "An overview of early vision in inceptionv1," *Distill*, vol. 5, no. 4, pp. e00024–002, 2020.

[94] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annual review of neuroscience*, vol. 24, no. 1, pp. 1193–1216, 2001.

[95] C. J. Hillar and F. T. Sommer, "When can dictionary learning uniquely recover sparse data from subsamples?," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6290–6297, 2015.

[96] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[97] S. Sanborn, C. Shewmake, B. Olshausen, and C. Hillar, "Bispectral neural networks," *International Conference on Learning Representations (ICLR)*, 2023.

[98] R. Rao and D. Ruderman, "Learning lie groups for invariant visual perception," *Advances in neural information processing systems*, vol. 11, 1998.

[99] K. Desai, B. Nachman, and J. Thaler, "Symmetry discovery with deep learning," *Physical Review D*, vol. 105, no. 9, p. 096031, 2022.

[100] V. Nivoliers and B. Lévy, "Approximating functions on a mesh with restricted voronoi diagrams," in *Computer Graphics Forum*, vol. 32, pp. 83–92, Wiley Online Library, 2013.

[101] K. V. Mardia and P. E. Jupp, *Directional statistics*, vol. 494. John Wiley & Sons, 2009.

[102] C. P. Klingenberg, "Walking on kendall's shape space: understanding shape spaces and their coordinate systems," *Evolutionary Biology*, vol. 47, no. 4, pp. 334–352, 2020.

[103] R. Brown, "From groups to groupoids: a brief survey," *Bull. London Math. Soc*, vol. 19, no. 2, pp. 113–134, 1987.

Part II

Included Publications

Paper A

Active Nearest Neighbor Regression Through Delaunay Refinement

Alexander Kravberg*, Giovanni Luca Marchetti*, Vladislav Polianskii*,
Anastasiia Varava, Florian T. Pokorny, Danica Kragic

Abstract

We introduce an algorithm for active function approximation based on nearest neighbor regression. Our Active Nearest Neighbor Regressor (ANNR) relies on the Voronoi-Delaunay framework from computational geometry to subdivide the space into cells with constant estimated function value and select novel query points in a way that takes the geometry of the function graph into account. We consider the recent state-of-the-art active function approximator called DEFER, which is based on incremental rectangular partitioning of the space, as the main baseline. The ANNR addresses a number of limitations that arise from the space subdivision strategy used in DEFER. We provide a computationally efficient implementation of our method, as well as theoretical halting guarantees. Empirical results show that ANNR outperforms the baseline for both closed-form functions and real-world examples, such as gravitational wave parameter inference and exploration of the latent space of a generative model.

A.1 Introduction

The need to *actively* approximate a function by iteratively querying novel points from its domain appears in a variety of theoretical and experimental areas, such as modern physics and astronomy [1, 2], chemistry [3, 4] and density estimation [5–7].

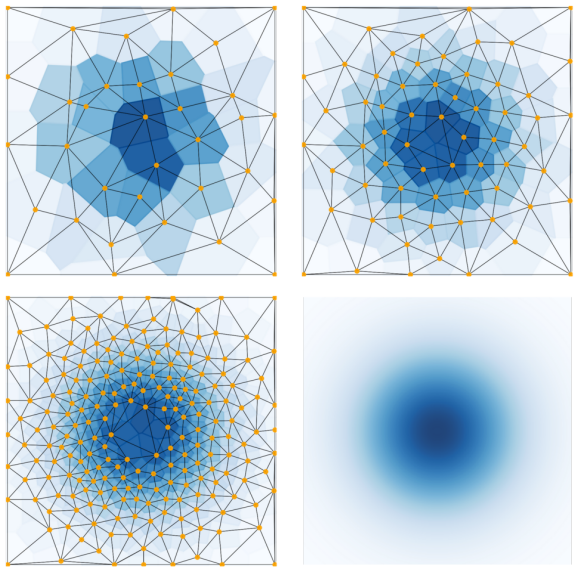


Figure A.1: The ANNR progressively approximating a Gaussian function. The approximation is depicted in blue-scale, with the ground-truth function plotted in the lower-right corner. The Delaunay triangulation is depicted in black as well.

Typically, the function to be approximated is not explicitly known, but can be evaluated at arbitrary points from its domain. Such formulation poses two intertwined problems: selecting points for evaluation (queries), and performing interpolation given a dataset with known function values. For the former, naive approaches such as sampling uniformly from the domain are computationally infeasible, especially for sparse and high-dimensional functions. Function evaluation is often expensive in real-life applications, as querying a single data point might require running a time- or resource-consuming experiment. It is thus crucial to design efficient strategies for selecting points to evaluate the function upon.

Recently, a scalable function approximator employing active querying has been proposed under the name DEFER [8]. It relies on a rectangular partitioning of the ambient space with the datapoints being the centers of the partitions, and approximates the (density) function piecewise constantly on each rectangle. DEFER outperforms state-of-the-art sampling methods in parameter inference tasks as well as in arbitrary function approximation. It is worth noting that despite [8] generally address density estimation problems, DEFER is effectively an active function approximator and differs from traditional density estimators [9] that are designed for static data. Using a rectangular partitioning, however, has a number of disadvantages. Rectangular approximations are not optimal for arbitrary shapes, especially

in high dimensions. Indeed, shape approximation via rectangles becomes progressively worse as dimensions grow, which can be illustrated in a well-known 'spiky rectangle phenomenon' [10].

We instead propose to upgrade the *Nearest Neighbour Regressor* (NNR) [9] to an active setting. The NNR is a function approximator which is locally constant on the Voronoi tessellation. The space is thus partitioned into Voronoi cells, which are arbitrary polytopes adaptive to the local geometry of data [11]. Such a space partitioning and the corresponding locally-constant approximation address the aforementioned disadvantages of DEFER, which we empirically show in the present work.

The core idea behind our active querying strategy is to look for points where the estimated function presents the largest variation. Such points are the most informative for the update of the approximator. This is done by considering the Delaunay triangulation [11], which is dual to the Voronoi tessellation. The triangulation allows to discretize the graph of the function and to look at its volume, which captures the function variation. Working with the graph of the function allows to balance between the exploration and exploitation strategies. This leads to a geometry-aware procedure deemed *Active Nearest Neighbour Regressor* (ANNR). Voronoi tessellations and their dual Delaunay triangulations enable to solve both the aforementioned problems of interpolation and querying within a single geometric framework.

To make the computation of the Delaunay triangulation feasible in high dimensions, we build upon on the approximate stochastic method described in [12]. Additionally, we prove via geometric arguments that the volumes of the Delaunay simplices over the graph of f get arbitrary small as the procedure progresses, obtaining halting guarantees for the ANNR. Our implementation of the ANNR is available at `https://github.com/vlpolyansky/annr`.

Our main contributions can be summarized as follows:

- A novel active querying procedure deemed ANNR exploiting the geometry of the graph of f through the Delaunay triangulation;
- An efficient high-dimensional implementation and a theoretical proof of halting for the ANNR;
- An empirical investigation of the ANNR through a series of experiments demonstrating improved performance and robustness over the recently introduced active function approximator DEFER.

A.2 Related Work

Delaunay Triangulations. The Delaunay triangulations were originally introduced in [13] as natural triangulations of point-clouds in arbitrary dimension. Because of their remarkable geometrical properties, they have seen extensive applications within computer graphics [14, 15] and topological data analysis [16]. The ANNR is in particular related to Delaunay-based techniques for mesh refinement [17], whose goal is to refine the Delaunay triangulation (mesh) of a coarse point-cloud by progressively adding novel points. Ruppert's algorithm [18] and Chew's second algorithm [19] are the most popular to this end. The idea underlying both algorithms is to insert the circumcenters of poor quality Delaunay triangles. Although our method follows a similar pattern, we are concerned with the task of function approximation rather than mesh refinement. We thus consider the known values of the ground-truth function and query points in order to refine the approximation, rather than aiming to optimally fill the ambient space. Moreover, mesh refinement algorithms and applications to computer graphics in general are concerned with a two- or three-dimensional ambient space. In contrast, we provide a general framework and an efficient implementation of the ANNR in high dimensions.

Deep Active Learning. Active querying strategies have been extensively studied in the context of deep learning. However, the vast majority of methods assume a predefined finite pool from which points can be queried ('pool-based active learning') or that datapoints can be sampled from the ground-truth distribution and then rejected for querying ('stream-based active learning') [20–22]. These methods typically deploy an 'acquisition function' representing some form of uncertainty that the desired query has to maximize [23–25]. Picking an arbitrary point in the ambient space –sometimes referred to as 'membership query synthesis'– is rarely considered in the literature and the corresponding subfield of active learning is relatively undeveloped. The reason is twofold: first, maximizing an acquisition function in the continuum for black-box models such as deep neural networks would require an additional expensive optimization procedure (gradient descent, for example) at each querying step. In contrast, simple but interpretable function approximators such as the piecewise linear ones considered in this work and in DEFER [8] allow to design querying strategies without the need of an acquisition function. Second, an arbitrary datapoint in the ambient space is likely to result in noise and thus to be uninformative to query, if even possible [26]. Due to recent advances in generative modeling, this can be nowadays amended by looking for points to query in the latent space of a generative model [27].

Active Sampling in Bayesian Inference. Acquisition functions based on uncertainty naturally occur in Bayesian inference, which is often used to approximate unknown distributions [5]. Classical Bayesian methods have a number of disadvantages, such as assuming a certain form of a distribution, requiring computable gra-

dients, and intractability. Recent developments in approximate Bayesian computation and Monte Carlo sampling methods address some of these issues [28–30], albeit suffering from impractical computational complexity [31]. When compared to DEFER, state-of-the-art Bayesian sampling methods showed comparable or worse performance, in particular, on multi-modal, sparse and discontinuous distributions [8]. In addition, these methods can only be applied to a particular class of likelihood functions, and are typically designed to perform sampling from the estimated distributions rather than to estimate a (likelihood) function value at an arbitrary point.

A.3 Method

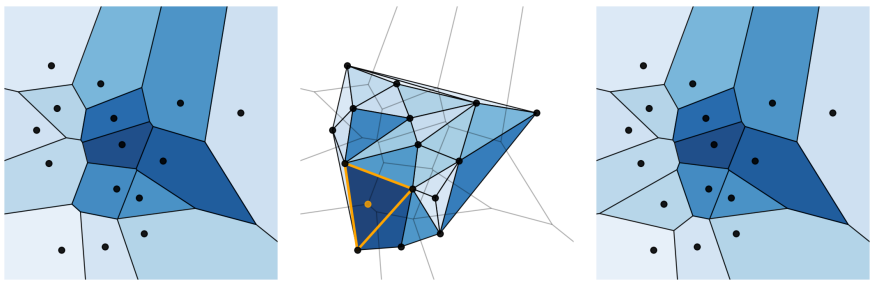


Figure A.2: A graphical depiction of the ANNR querying procedure. **Left**: the NNR (in blue-scale) of a two-dimensional dataset. **Center**: the corresponding Delaunay triangulation. The simplices are colored by the volume of their liftings, with the largest one highlighted in orange together with its circumcenter. **Right**: the updated NNR after the orange point has been added to the dataset.

Let $f : \mathcal{X} \to \mathbb{R}$ be a function defined on a connected metric space $(\mathcal{X}, d)$, where $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ denotes the distance. Given a finite set of points $P \subseteq \mathcal{X}$ (referred to as datapoints) on which the values of f are known, a natural way to extend f beyond P is to regress to the values at the nearest datapoint [9].

Definition A.3.1. Let x be a point such that all the distances $d(x, p)$, $p \in P$ are distinct. The value of the *Nearest Neighbor Regressor* (NNR) $\widetilde{f}$ at x coincides with the value of f at the closest datapoint i.e.,

$$\widetilde{f}(x) = f(\overline{p}), \qquad \overline{p} = \operatorname*{argmin}_{p \in P} d(x, p). \tag{A.1}$$

Recall that the *Voronoi cell* $C(p)$ of $p \in P$ contains the points in $\mathcal{X}$ that are closer to p than to any other datapoint i.e.,

$$C(p) = \{x \in \mathcal{X} \mid \forall q \in P \;\; d(x, q) \geq d(x, p)\}. \tag{A.2}$$

The Voronoi cells are closed, cover the ambient space $\mathcal{X}$ and intersect at their boundary. When $\mathcal{X} = \mathbb{R}^m$, they are arbitrary m-dimensional convex polytopes [11]. From the point of view of the Voronoi tessellation, the NNR is defined on the interior of Voronoi cells and is constant locally therein.

A.3.1 Active Nearest Neighbor Regression

In this work we upgrade the NNR to an active setting. A general active procedure consists in updating a dataset inductively by querying for the value of f at a new datapoint based on the current dataset, on which f is assumed to be known. Starting from an initial dataset P_0, this produces a sequence $\{P_t\}_{t\in\mathbb{N}}$ obtained by adding a datapoint at each step: $P_{t+1} = P_t \cup \{p_t\}$. We are now going to describe our proposed querying strategy.

To update the dataset, we first consider the triangulation which is dual to the Voronoi tessellation. From now on, we focus on the m-dimensional Euclidean space $\mathcal{X} = \mathbb{R}^m$. In the following we denote by $\langle\cdot\rangle$ the convex hull of a set.

Definition A.3.2. The *Delaunay triangulation* Del_P generated by P is the simplicial complex with vertices in P that contains a k-dimensional simplex $\sigma = \langle v_0, \cdots, v_k\rangle$, $v_i \in P$, if and only if

$$\bigcap_{0\leq i\leq k} C(v_i) \neq \emptyset. \tag{A.3}$$

If P is in general position then the simplices in Del_P are non-degenerate. In that case, a remarkable property of the Delaunay triangulation is that no point in P lies inside the circumsphere of an m-dimensional simplex $\sigma \in \mathrm{Del}_P$ [11].

Our querying strategy intuitively relies on the variation of f over the m-dimensional simplices of the Delaunay triangulation. Concretely, we compute the volume of the 'lifting' of such simplices to the graph $\Gamma_{\lambda f} = \{(x, \lambda f(x)) \mid x \in \mathcal{X}\}$ of λf, where $\lambda \in \mathbb{R}_{\geq 0}$ is a hyperparameter.

Definition A.3.3. The *lifting* via λf of an m-dimensional simplex $\sigma = \langle v_0, \cdots, v_m\rangle$ in $\mathbb{R}^m$ is the simplex in $\mathbb{R}^{m+1}$

$$\hat{\sigma} = \langle (v_0, \lambda f(v_0)), \cdots, (v_m, \lambda f(v_m))\rangle. \tag{A.4}$$

Our algorithm looks for the m-dimensional simplex $\sigma \in \mathrm{Del}_P$ that maximizes $\mathrm{Vol}(\hat{\sigma})$. To this end, the k-dimensional volume of a simplex $\sigma = \langle v_0, \cdots, v_k\rangle$ can be efficiently computed even for high dimensions via the *Cayley-Menger determinant* [32]:

$$\mathrm{Vol}(\sigma) = \sqrt{\frac{(-1)^{k+1}}{2^k (k!)^2}\, \det M}. \tag{A.5}$$

Here, M is the $(k+2) \times (k+2)$ matrix obtained by padding by a top row and a left column equal to $(0, 1, \cdots, 1)$ the matrix of mutual distances $d(v_i, v_j)^2$.

The role of the hyperparameter λ is to 'sharpen' the lifted simplices. Since λ controls the increment of λf, $\mathrm{Vol}(\hat{\sigma})$ depends monotonically on λ. As $\mathrm{Vol}(\sigma)$ is constant, $\lambda \gg 0$ encourages simplices with high variation even if the base simplex σ is small. On the other hand, $\mathrm{Vol}(\sigma) \sim \mathrm{Vol}(\hat{\sigma})$ for $\lambda \sim 0$, in which case the ANNR regularly explores the domain by looking for the largest simplices, disregarding f and its variation completely. In summary, λ can be interpreted as governing a trade-off between exploitation of the estimated variation and domain exploration – a classical compromise in active learning [33]. We make the latter statement formal in Section A.4.1 and refer to Section A.5.1 for a further discussion around the practical choice of λ.

Given the simplex $\overline{\sigma}$ that maximizes $\mathrm{Vol}(\hat{\sigma})$, the point we query is the dual of $\overline{\sigma}$ in the Voronoi tessellation. In other words, the novel query is the value of f at the intersection between the $m+1$ Voronoi cells of the vertices of $\overline{\sigma}$. It is thus a point where the NNR (Equation A.1) is maximally discontinuous, which motivates the need to gain information on f around it. Geometrically, it coincides with the *circumcenter* of $\overline{\sigma}$ (i.e., the point in $\mathbb{R}^m$ equidistant from the vertices of $\overline{\sigma}$) and we consequently denote it by $\mathrm{Circ}(\overline{\sigma})$. Our querying strategy is graphical depicted in Figure A.2 and can be formally summarized as follows:

$$p_{t+1} = \mathrm{Circ}(\overline{\sigma}), \qquad \overline{\sigma} = \underset{\sigma \in \mathrm{Del}_{P_t}}{\mathrm{argmax}}\ \mathrm{Vol}(\hat{\sigma}). \tag{A.6}$$

We stop the iteration when the maximum volume of a lifting reaches a given threshold $\varepsilon > 0$. If care is taken in order to constrain the dataset in a compact region (which is discussed in Section A.3.2), the algorithm is guaranteed to halt with mild assumptions on f (see Section A.4.2). The overall procedure is summarized in the pseudocode Algorithm A.1.

A.3.2 Bounding the Query

The circumcenter of a simplex can possibly lie outside of the simplex itself. When a m-simplex is close to being degenerate (i.e., contained in a $(m-1)$-dimensional affine subspace), the circumcenter tends to escape in an infinite direction. It is thus necessary to limit the expansion of the dataset within a compact region.

We propose to fix a convex compact set $A \subseteq \mathbb{R}^m$ containing the initialization P_0 and to constrain P_t to be contained in A in the following way. Suppose that $\overline{\sigma}$ is the simplex in Del_{P_t} whose lifting has the largest volume and that $\mathrm{Circ}(\overline{\sigma}) \notin A$. Instead of setting $p_{t+1} = \mathrm{Circ}(\overline{\sigma})$ we set p_{t+1} as the intersection between the boundary ∂A of A and the segment connecting the barycenter of $\overline{\sigma}$ (which is contained in $\overline{\sigma}$) and $\mathrm{Circ}(\overline{\sigma})$. Such segment is known as 'Euler line' of $\overline{\sigma}$ [34] and the intersection

Algorithm A.1 Active Nearest Neighbor Regression (ANNR)

```
Initialize P
maxVol = ε
while maxVol ≥ ε do
  Compute the Delaunay triangulation Del_P
  maxVol = 0
  for σ ∈ Del_P of dimension m do
    Compute the volume Vol(σ̂) via Equation A.5
    if Vol(σ̂) > maxVol then
      maxVol = Vol(σ̂)
      maxSimplex = σ̂
    end if
  end for
  Add the circumcenter of maxSimplex to P
end while
```

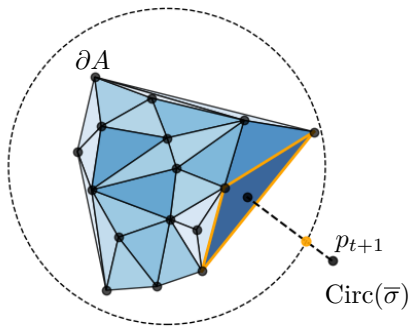


Figure A.3: Depiction of the query in the case the circumcenter falls outside of the bounding region A.

between it and ∂A is unique due to the convexity of A. The intuition behind this choice for p_{t+1} is that it is the furthest point in A from (the barycenter of) $\overline{\sigma}$ in the direction of the circumcenter of the latter. Moreover, the theoretical halting guarantees discussed in Section A.4.2 crucially rely on the procedure described here. A graphical depiction is provided in Figure A.3.

The bounding set A additionally enables to initialize the dataset P_0 by uniformly sampling from A. Since the Delaunay triangulation covers the convex hull of its vertices, one can additionally enlarge P_0 so that $\langle P_0 \rangle = A$ and thus $\langle P_0 \rangle = A$ for every t. This is always possible if A is polytopal. When A is a hypercube (which is the case in all of our experiments) this can be done by adding its 2^m vertices to

P_0. We stick to this form of initialization when implementing Algorithm A.1.

A.3.3 Computing the Delaunay Triangulation

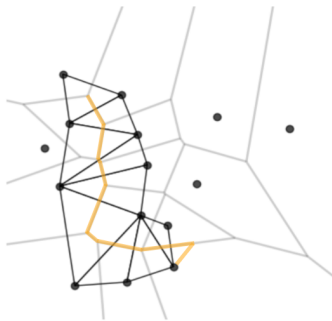


Figure A.4: A depiction of a random walk along the boundary of the Voronoi cells (orange) together with the Delaunay simplices found along the way.

A naive implementation of ANNR involving standard exact methods of constructing a Delaunay triangulation would present a significant computational challenge. Namely, the Delaunay triangulation is prohibitively expensive to compute in high dimensions as the number of simplices grows exponentially with respect to the dimension m in a general scenario [35].

We propose to approximately infer Del_P via an adaptation of the stochastic ray-casting technique first considered in [36] and later expanded in [12]. The central idea lies in performing a random walk along the boundary of the Voronoi cells in order to look for their vertices, which in turn correspond by duality to the desired Delaunay simplices. More precisely, a Markov Chain (MC) is constructed in the following way. First, a starting datapoint x_0 is picked uniformly from P. Then a random direction $\theta_0 \in \mathbb{S}^{m-1}$ is chosen and the intersection x_1 between the ray $\{x_0+t\theta_0\}_{t\in\mathbb{R}_{\geq 0}}$ cast from x_0 and a $(m-1)$-dimensional face of the polytopal Voronoi cell $C(x_0)$ is found by an explicit analytic expression. The procedure is repeated by additionally constraining the subsequent ray to lie on the aforementioned face, obtaining a point x_2 on a $(m-2)$-dimensional face of $C(x_0)$, and so on. After m rounds of iteration, a vertex x_m of the Voronoi cell (i.e., a 0-dimensional face) is obtained. By keeping track of the other Voronoi cells to which the encountered faces belong to, the (vertices of the) Delaunay simplex corresponding to x_m are automatically available. After that, the random walk continues on the 1-skeleton of the Voronoi tessellation in an analogous manner, possibly departing from $C(x_0)$ and finding other Delaunay simplices. The result is a subset of the Delaunay triangulation which approximates the whole Del_P. By deploying a nearest-neighbor lookup structure such as a k-d tree [37], every ray intersection described above

can be performed with a complexity that depends *logarithmically* on the number of datapoints. We refer to [12] for further details. A graphical depiction of the described procedure is presented in Figure A.4.

The vertices of the Voronoi cells corresponding to the found Delaunay simplices (i.e., the circumcenters of the latter) are obtained as a byproduct, thus there is no need for a further computation of the point to query. Because of the iterative nature of the search for Delaunay simplices, we integrate the latter with the main loop in Algorithm A.1 for a further computational improvement. We further heuristically adjust the initialization of the random walks by picking datapoints close to barycenters of the simplices with the highest lifted volume at the previous step. As suggested in [12], this can be accomplished via a 'visibility walk' [38] in the new 1-skeleton towards such barycenters before initiating the random walk.

A.3.4 Complexity Analysis

In this section we provide a comparison of the computational complexity of the ANNR and DEFER for both data querying and evaluation of the approximated function.

Since each random walk has logarithmic complexity (Section A.3.3), querying a new datapoint has complexity $\mathcal{O}(L \log |P|)$ for the ANNR, where L is the number of MC steps performed. The latter is a hyperparameter typical for MC methods and its effect can be mitigated by running multiple walks in parallel. For DEFER, querying has complexity $\mathcal{O}(\log |P|)$. The difference in complexity due to L indeed reflects the price of approximating the Delaunay triangulation, as opposed to the exact geometry of rectangular partitions in DEFER. Additionally, the ANNR computes volumes of simplices, which has complexity $\mathcal{O}(m^3)$ w.r.t. the dimension m due to the Cayley-Menger determinant (Equation A.5), in contrast to linear complexity for volumes of rectangles in DEFER. Overall, querying complexity of the ANNR is $\mathcal{O}(L(m^3 + \log |P|))$.

Evaluating the approximated function has identical complexity $\mathcal{O}(\log |P|)$ with respect to the current dataset size $|P|$ in both methods. This is due to data structures such as k-d trees underlying both the nearest neighbor lookup for Voronoi cells in the ANNR and the rectangle lookup in DEFER.

A.4 Theoretical Results

A.4.1 Geometric Interpretation

In this section we present a geometric interpretation of the ANNR based on an inequality for volumes of graphs of functions. To this end, suppose that $\Omega \subseteq \mathbb{R}^m$ is an m-dimensional connected and compact submanifold (with boundary). For a

smooth function $f : \Omega \to \mathbb{R}$ denote by Γ_f its graph, which is an m-dimensional manifold. Additionally, denote by $f_\Omega = \frac{1}{\mathrm{Vol}(\Omega)} \int_\Omega f$ the average of f over Ω and by g_f the matrix $\nabla_f \otimes \nabla_f$ i.e., $(g_f)_{i,j} = \partial_{x_i} f \, \partial_{x_j} f$.

Proposition A.4.1. *There exists a constant $C > 0$ such that for every smooth function $f : \Omega \to \mathbb{R}$ the following inequality holds:*

$$\log \mathrm{Vol}(\Gamma_f) \geq C\|f - f_\Omega\|_2^2 + \log \mathrm{Vol}(\Omega) + o(\|g_f\|^2). \tag{A.7}$$

We refer to the Appendix for a proof. If Ω is a Voronoi cell then the NNR takes the value of f at the only datapoint contained in the cell, which is a one-sample Monte Carlo estimate of f_Ω. In other words, $f_\Omega \sim \widetilde{f}$ on Ω. In light of Proposition A.4.1, the (logarithm of the) volume of the graph of f is approximately bounding the error between f and its NNR estimation plus a term $\log \mathrm{Vol}(\Omega)$ penalizing domains with large volume. The latter can be thought as a form of regularization. The ANNR exploits this as it computes and approximation to the volume of the graph of (a multiple of) f. Such volume is not directly approximable when Ω is a Voronoi cell because of lack of data in Ω. We thus compute it over the discretization of the manifold Γ_f given by the lifted Delaunay triangulation $\{\hat{\sigma}\}_{\sigma \in \mathrm{Del}_P}$. In other words, the volume is computed for the simplices formed by the datapoints in neighboring intersecting cells.

Note that if we replace f by λf for some $\lambda \in \mathbb{R}_{\geq 0}$ then the right hand-side of Equation A.7 becomes $C\lambda^2\|f - f_\Omega\|_2^2 + \log \mathrm{Vol}(\Omega) + \lambda^4 o(\|g_f\|^2)$. A natural interpretation is that λ controls the balance between error minimization (corresponding to the term $\|f - f_\Omega\|_2^2$) and exploration of the domain (corresponding to the regularization term $\log \mathrm{Vol}(\Omega)$). This motivates the insertion of the hyperparameter λ in the ANNR (see Equation A.4).

A.4.2 Halting Guarantees

In this section we establish formal halting guarantees for the ANNR given a continuity assumption on the ground-truth function f. Since our algorithm stops when a fixed threshold ε is met, halting can be equivalently reformulated as the vanishing of the corresponding lower limit. The halting guarantee is then given by the following.

Proposition A.4.2. *Assume that the ground-truth function f is Lipschitz and let $s_t = \max_{\sigma \in \mathrm{Del}_{P_t}} \mathrm{Vol}(\hat{\sigma})$. Then Algorithm A.1 always halts for any ε or, in other words, $\underline{\lim}_{t \to \infty} s_t = 0$.*

We refer to the Appendix for a proof. Although the Lipschitz assumption is necessary for the theoretical proof, we empirically show in the experimental section that the ANNR is well-behaved even for highly discontinuous functions such as characteristic functions of geometrically articulated domains (see Section A.5.2). The

proof of Proposition A.4.2 remains valid when s_t and the corresponding maximum is computed w.r.t. a partial Delaunay triangulation coming from the approximation discussed in Section A.3.3. That is, the ANNR is guaranteed to halt even with the approximation procedure.

A.5 Experiments

We select DEFER as our primary baseline as it is the state-of-the-art method for active function approximation which is suitable for arbitrary functions [8]. As an ablation, we additionally compare with the non-active version of the NNR which samples datapoints *uniformly* from A, denoted by nANNR.

In all the experiments, the number of queries is referred to as N. For numerical validation, we rely on the standard score Mean Average Error

$$\text{MAE} = \frac{1}{|P_{\text{test}}|} \sum_{p \in P_{\text{test}}} \left| \widetilde{f}(p) - f(p) \right|, \tag{A.8}$$

where P_{test} is a test set. We generate P_{test} as an equally-spaced grid in $m = 2$ dimensions and by uniformly sampling from A when $m > 2$ since exhaustive grid sampling is computationally infeasible.

A.5.1 Hyperparameter λ

As discussed in Section A.3.1 and A.4.1, the ANNR has a single hyperparameter – the lifting coefficient λ – which governs a natural exploration-exploitation trade-off. This is graphically demonstrated in Figure A.5, where a higher λ encourages querying in areas where f varies the most. In practice, we suggest the following heuristic choice for λ, which we implement in our experiments. We select λ proportional to the size of the domain and inversely proportional to the scale of the function, effectively bringing domain and codomain to the same scale to balance exploration and exploitation of the function: $\lambda = \frac{\text{Vol}(A)}{\max f - \min f}$. Here, max and min can be estimated from prior knowledge or, more concretely, directly evaluated from the initial dataset P_0.

A.5.2 Geometric Advantages of the ANNR

In this experiment, we consider characteristic functions with various supports and demonstrate that the ANNR is well suited for approximating the support shape. The ANNR leverages the variable geometry of Voronoi cells, which results in a fine-grained approximation with a small number of queries. In contrast, DEFER relies on a rectangular partitioning of the space, which, as we show, can result in sub-optimal approximations. This feature also allows ANNR to approximate functions with arbitrary compact domains, in particular those with a small volume compared

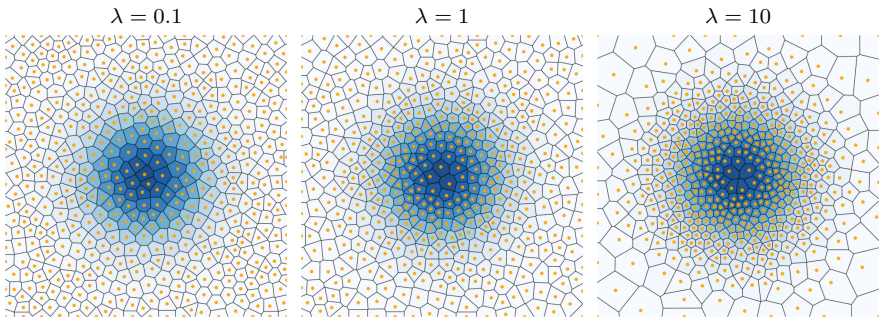


Figure A.5: ANNR approximation of a normalized Gaussian with $\sigma^2 = 0.1$ for various values of λ ($N = 500$).

to A (see the Appendix for an example).

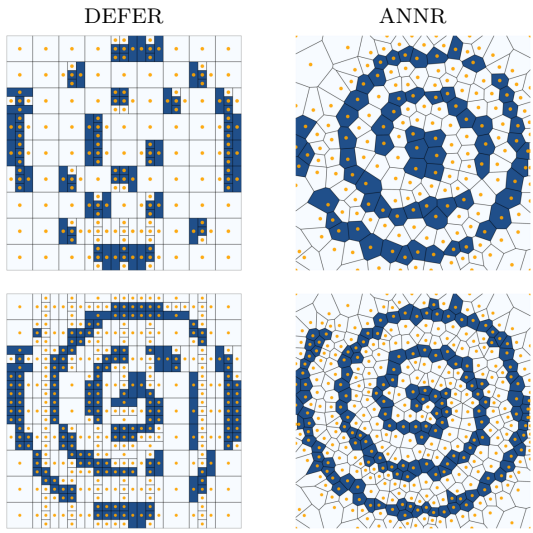


Figure A.6: Approximation of a spiral characteristic function (see the Appendix for details). $N = 200$ (top), $N = 400$ (bottom).

We first consider the characteristic function of a spiral (Figure A.6). The ANNR displays a visibly better approximation and has a connected support already after $N = 400$ queries. The support of DEFER is instead highly disconnected as it is unable to capture regions where the spiral is misaligned with the Cartesian axes due to its rectangular bias (see the Appendix for details).

Rotational Equivariance. DEFER is sensitive to rotations of the domain due to its inherent bias towards Cartesian axes. In contrast, the ANNR allows to approximate shapes without any orientation bias since Voronoi cells are arbitrary polytopes. Figure A.7 demonstrates improved stability of the ANNR approximation with respect to rotations of $\mathbb{R}^2$ compared to DEFER (see the Appendix for details).

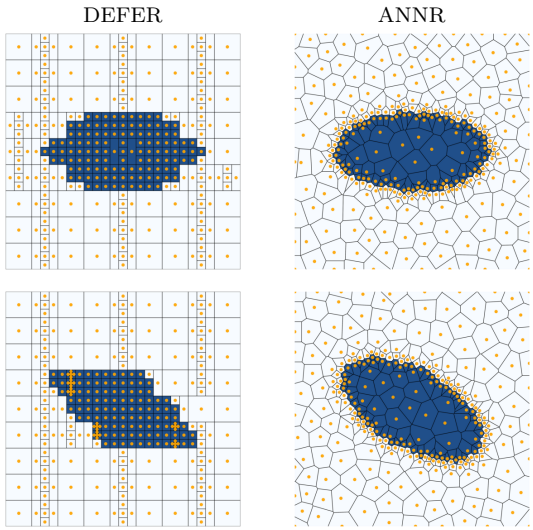
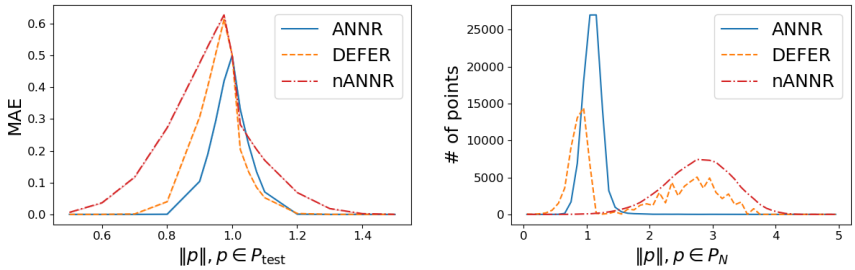


Figure A.7: Approximation of an ellipse characteristic function $\mathbf{1}_{x^2+4y^2\leq 1}$ with 0° (top) and 30° (bottom) rotations of the domain ($N = 300$).

Curse of Dimensionality. The bias of DEFER towards rectangular geometry can potentially affect the quality of approximation in high-dimensions due to the increasing *spikiness* of rectangles [10, 39]. To demonstrate this, we consider a characteristic function $\mathbf{1}_{\|x\|\leq 1}$ of a unit ball in $\mathbb{R}^6$ (with $A = [-2, 2]^6$). Figure A.8 reports the score (Figure A.8a) and query point distribution (Figure A.8b) with respect to the norm of test and query points respectively.

Figure A.8a shows that the MAE scores decrease when the test points are sampled away from the discontinuity of f i.e., the boundary of the ball. ANNR's score is significantly lower inside the ball ($\|p\| < 1$, $p \in P_{\text{test}}$). At the same time, Figure A.8b shows that all the queries of the ANNR are queried in the proximity of the boundary, while DEFER's query norm distribution is shifted towards the uniform one (nANNR).



(a) MAE scores for test sets at different norms.

(b) Norm distribution of the queried points.

Figure A.8: Analysis of the 6-dimensional unit ball approximations based on the norm of data ($N = 10^5$ and $|P_{\text{test}}| = 10^7$ per norm).

A.5.3 Performance Comparison

In this section we compare the performance of the ANNR with DEFER and the non-active sampler nANNR for articulated functions including parameter estimation of gravitational waves and exploration of the latent space of a deep generative model. The scores are presented in Table A.1. For the ANNR and the nANNR, we report averages and standard deviations over 10 runs with different random initializations (in contrast, DEFER has no stochasticity). We set $N = 1000$ for the two-dimensional experiment (latent manifold exploration) and $N = 10^5$ when $m > 2$. Our method outperforms the baseline in terms of MAE by a significant margin.

Table A.1: Performance Comparison (MAE).

	ANNR	DEFER	nANNR
Gravitational waves parameter estimation	0.4710 ±0.0232	0.5309	0.5317 ±0.2175
Latent space volume density	47.0509 ±0.4975	50.5073	54.6290 ±0.8010

Gravitational Waves. A large area of application for active function approximation is astrophysics and, in particular, gravitational wave parameter inference problems [1, 40]. Observation of gravitational waves are rare, and the magnitude of the effect is low, implying scarcity of collected data. With limited observations and dozens of parameters describing the gravitational event, the problem of parameter estimation is commonly solved by Bayesian inference methods approximating

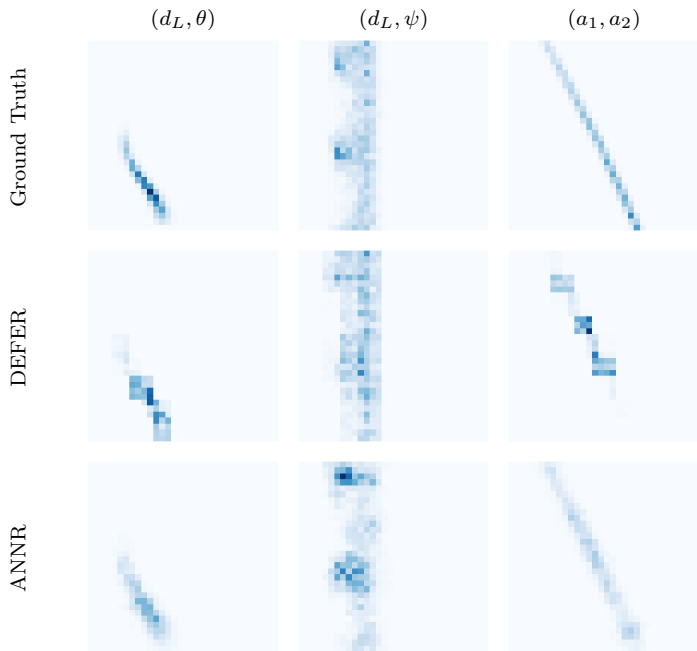


Figure A.9: Marginal distributions for gravitational waves over selected two-dimensional slices. Each slice is identified by a pair of parameters named as in [8].

a simulated log-likelihood function on the parameters [1, 2].

We use the same 6-dimensional formulation of parameter inference as described in [8], and refer the reader to the Appendix for a detailed description of parameters. In addition to the original setup, we rescale the function domain to a unit hypercube $A = [0, 1]^6$ for convenience and multiply f by a factor of e^{8000} for better numerical stability. Lastly, in order to deal with practical unboundedness of the density function, we perform an adaptive clipping of extensively sharp volumes. The details of the clipping are available in the Appendix.

Apart from the MAE comparison in Table A.1, we present a visualization of the function approximations in Figure A.9 of a single run. The figure displays histograms over marginal distributions over a set of two-dimensional slices of the domain. Each marginalization was performed via Monte-Carlo sampling with 10^5 samples per bin. The figure shows the visual closeness of ANNR marginalizations to the original distribution compared to the baseline. Marginals over all 15 pairs of parameters are available in the Appendix.

Latent Manifold Exploration. Our last experiment addresses a manifold explo-

ration task on the latent space of a deep generative model. The generative process enables to query an arbitrary point in the latent space and thus suits the framework of the ANNR and DEFER. Moreover, decoding the queried points allows to interpret and semantically evaluate the exploration procedure. To this end, we deploy a generative model $\varphi : \mathcal{Z} \to \mathcal{X}$ that maps a prior distribution on the latent space $\mathcal{Z} = \mathbb{R}^m$ to the data distribution on $\mathcal{X} = \mathbb{R}^n$ and regard it as (the probabilistic analogue of) a parametrization of the data manifold. The 'mass' distribution of data on $\mathcal{Z}$ is represented by the induced *volume density* f i.e., the density of the volume form of the Riemannian metric induced by φ on $\mathcal{Z}$ via pull-back [41, 42]. Such volume density is concretely expressed as $f(z) = \sqrt{\det\left(J_\varphi(z)^T J_\varphi(z)\right)}$ where $J_\varphi(z)$ is the Jacobian matrix of φ at $z \in \mathcal{Z}$. Intuitively, f can be seen as a 'fuzzy' characteristic function of the latent manifold and an active function approximator for f can be interpreted as progressively exploring such manifold.

We train φ as (the decoder of) a Variational Autoencoder (VAE, [43]) on the MNIST dataset [44] of gray-scale images of hand-written digits ($n = 784$). We deploy a two-dimensional latent space ($m = 2$) with a standard Gaussian prior. Despite its low dimensionality, such a latent space is sufficient for generation of quality MNIST images [43] and enables direct visualization. For better parametrization quality, the latent prior is additionally encouraged by a hyperparameter $\beta = 2$ multiplying the corresponding ELBO loss term [45].

Figure A.10 displays the progressive approximation of f as well as images decoded from queried points. As can be seen from the latter, around $t = 500$ the class of p_t (digit) is different at each step t and the ANNR is thus covering a wide semantic range of data. In contrast, around $t = 1000$ the decoded images are similar, indicating a phase close to convergence in which the approximation is refined locally.

Runtime Comparison. The overall flexibility of the geometric framework utilized by the ANNR comes with a certain computational cost, which highlights a tradeoff between effectiveness and efficiency of the methods. The computational cost of the ANNR is largely mitigated by the Delaunay approximation the method relies on. In addition, the runtime of the ANNR can be heavily controlled by adjusting the desired Markov Chain sampling precision and increasing the number of available parallel threads.

We provide a runtime comparison between the methods on experiments discussed earlier in this section in Table A.2. Each number represents the average runtime for a single experimental run. All experiments are performed on CPU Ryzen 9 5950X 16-Core. We note that in case of the ANNR single queries still take milliseconds to compute, which is reasonable for real-life applications with high function evaluation cost.

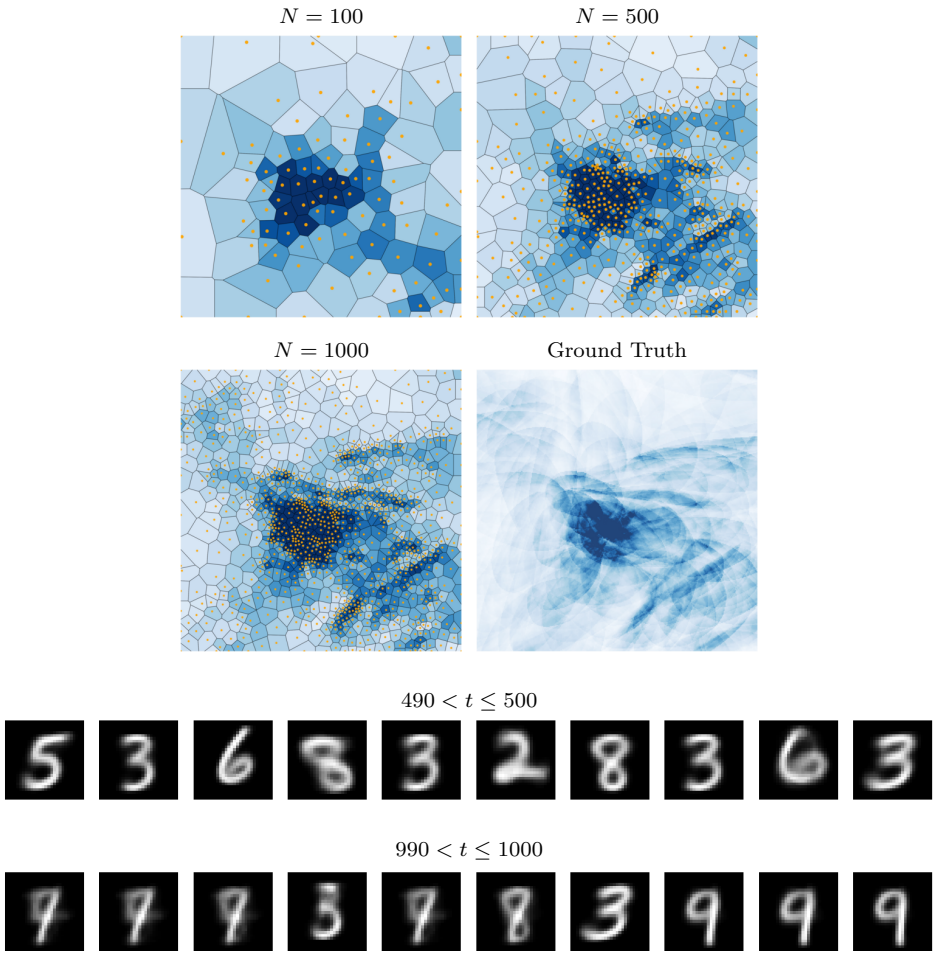


Figure A.10: Top: approximation of the latent space volume density of a deep generative model. **Bottom**: images decoded from the queried latent points p_t for two intervals of t.

A.6 Conclusion and Future Work

In this work we introduced the ANNR – an adaptation of nearest neighbor regression to an active setting. We provided a computationally efficient implementation of the ANNR as well as theoretical halting guarantees. Our empirical investigations have shown that the ANNR outperforms the state-of-the-art active function approximator called DEFER.

An interesting line for future investigation lies in designing active querying

Table A.2: Runtime Comparison.

	ANNR	DEFER
Gravitational waves parameter estimation	1347 s	186 s
Latent space volume density	220 ms	90 ms

strategies for other higher-dimensional function approximators from the existing literature. Examples include the piecewise linear Delaunay interpolator [46] or the tensor product of cubic splines [47].

A.7 Ackowledgements

This work was supported by the Swedish Research Council, the Knut and Alice Wallenberg Foundation and the European Research Council (ERC-BIRD-884807).

A.8 Appendix

A.8.1 Proofs of Theoretical Results

In this section we provide proofs for the theoretical results presented in the main body of the paper. We start by proving the result from Section A.4.1.

Proposition A.8.1. *There exists a constant $C > 0$ such that for every smooth function $f : \Omega \to \mathbb{R}$ the following inequality holds:*

$$\log \mathrm{Vol}(\Gamma_f) \geq C\|f - f_\Omega\|_2^2 + \log \mathrm{Vol}(\Omega) + o(\|g_f\|^2). \tag{A.9}$$

Proof. The volume of Γ_f can be expressed as

$$\mathrm{Vol}(\Gamma_f) = \int_\Omega \sqrt{\det(1 + g_f)} \tag{A.10}$$

and thus by Jensen inequality we get:

$$\log \frac{\mathrm{Vol}(\Gamma_f)}{\mathrm{Vol}(\Omega)} \geq \frac{1}{2\mathrm{Vol}(\Omega)} \int_\Omega \log\det(1 + g_f). \tag{A.11}$$

Since by general properties of matrices $\log\det(g_f) = \mathrm{tr}(\log g_f)$ and $\log(1 + g_f) = g_f + o(\|g_f\|^2)$ where $\|\cdot\|$ denotes the standard matrix norm, the right hand side of Equation A.11 reduces to

$$C_1 \int_\Omega \operatorname{tr}(g_f) + o(\|g_f\|^2) = C_1 \|\nabla_f\|_2^2 + o(\|g_f\|^2). \tag{A.12}$$

Finally, by the Poinaré-Wirtinger inequality [48] $\|\nabla_f\|_2^2 \geq C_2 \|f - f_\Omega\|_2^2$ for some constant $C_2 > 0$, which concludes the proof. □

We now prove the halting guarantee from Section A.4.2. First, we provide the following result concerning high-dimensional Euclidean geometry which will be necessary for the main proof. Recall that $A \subseteq \mathbb{R}^m$ is a convex compact set.

Lemma A.8.2. *For each $\delta > 0$ there exists an $\eta > 0$ such that for each n-dimensional simplex $\sigma \subseteq A$ if $\operatorname{Vol}(\sigma) > \delta$ then $d(v, x) > \eta$ for each vertex v of σ and each point x of the segment connecting the baricenter and the circumcenter of σ.*

Proof. By comparing the volume of a simplex with the volume of its circumsphere, there exists $\eta_1 > 0$ such that for any simplex σ if $\operatorname{Vol}(\sigma) > \delta$ then $R_\sigma > \eta_1$, where R_σ denotes the radius of the circumsphere of σ. There also exists a $\eta_2 > 0$ such that for any simplex σ if $\operatorname{Vol}(\sigma) > \delta$ then all the heights (i.e., the segments passing through the vertices and orthogonal to the opposite faces) of σ are greater than η_2. To see this, note that otherwise there would exist triangles of arbitrarily small heights and thus, by the volume constraint, with faces of arbitrarily large volume, contradicting the compactness of A. Now, given a simplex σ and a vertex v of σ, the segment connecting v to its opposite face and containing the barycenter b of σ is shorter than the height passing through v because of orthogonality of the latter. The barycenter separates that segment into a portion of $\frac{m-1}{m}$ of it containing v. Thus, we have $d(v, b) > \frac{m-1}{m}\eta_2$.

Consider $\eta_3 = \min\{\eta_1, \frac{m-1}{m}\eta_2\}$. For any simplex σ with circumenter c and barycenter b we then have $d(v, b) > \eta_3$ and $d(v, c) > \eta_3$ for every vertex v of σ. Since $(b-v)\cdot(c-v) > 0$, we have that $d(v, x) > \frac{\sqrt{2}}{2}\eta_3 = \eta$ for each x in the segment connecting b and c, as desired. □

We now prove the main result from Section A.4.2.

Proposition A.8.3. *Assume that the ground-truth function f is Lipschitz and let $s_t = \max_{\sigma \in \operatorname{Del}_{P_t}} \operatorname{Vol}(\hat{\sigma})$. Then Algorithm A.1 always halts for any ε or, in other words, $\underline{\lim}_{t\to\infty} s_t = 0$.*

Proof. Suppose by contradiction that there exists $\varepsilon > 0$ such that $s_t > \varepsilon$ for all sufficiently large t. Since f (and thus λf) is Lipschitz, there exists $\delta > 0$ such that for any simplex σ if $\operatorname{Vol}(\hat{\sigma}) > \varepsilon$ then $\operatorname{Vol}(\sigma) > \delta$.

Let us prove that there can not be infinite queries in the interior of the bounding region A. By comparing the volume of a simplex with the volume of its circumsphere, there exists $\gamma > 0$ such that for any simplex σ if $\text{Vol}(\sigma) > \delta$ then $R_\sigma > \gamma$, where R_σ denotes the radius of the circumsphere of σ. When a point $p_{t+1} = \text{Circ}(\overline{\sigma})$ is queried, since by hypothesis $s_t = \text{Vol}(\widehat{\overline{\sigma}}) > \varepsilon$ we have $R_{\overline{\sigma}} > \gamma$. But then $d(p_{t+1}, q) > \gamma$ for all $q \in P_t$ since no points in P_t are contained inside the circumsphere of the Delaunay simplex $\overline{\sigma}$. If infinite points in the interior of A are queried, one gets an infinite sequence in A whose pairwise distances are all greater than γ. This is impossible since A is compact.

As a consequence, the queried points p_{t+1} belong to the boundary of A for $t \gg 0$ and are obtained via the method described in Section A.3.2. By Lemma A.8.2 there exists $\eta > 0$ such that $d(v, p_{t+1}) > \eta$ for $t \gg 0$ where v is any vertex of the simplex $\overline{\sigma}$ whose lifting has the largest volume at step t. Since $\overline{\sigma}$ is a Delaunay simplex it holds that $d(q, p_{t+1}) > \eta$ for every $q \in P_t$ and $t \gg 0$, which again contradicts the compactness of A. □

A.8.2 Extended Experiments

Spiral

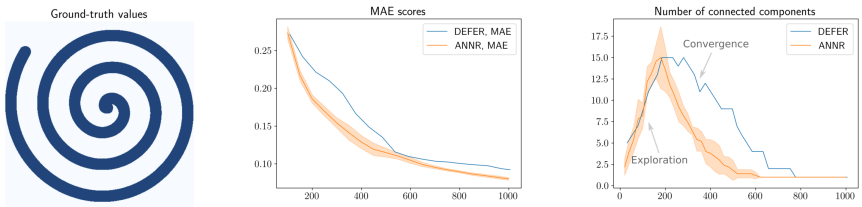


Figure A.11: Ground-truth plot of a spiral characteristic function and performance comparison between the ANNR and DEFER as the number of queries increases. Scores are averaged over 5 runs.

Arbitrary Domain

In many cases the actual domain of a function is not the entire $\mathbb{R}^m$, but rather some compact subspace. In many optimization algorithms, the function domain is artificially extended to a bounding volume (usually a bounding box) containing it, assuming some zero value outside of the original function domain. In higher dimensions, this significantly increases the volume to explore [10]. The ANNR takes advantage of the flexibility of Delaunay partitioning to restrict new queries to the (boundary of the) function domain. Figure A.12 illustrates an approximation of $f(x) = \|x\|$ restricted to the intersection of two circles (the volume of the domain is

only $\sim 1/30$ of the bounding box volume), which otherwise can be very inefficiently approximated by its bounding box.

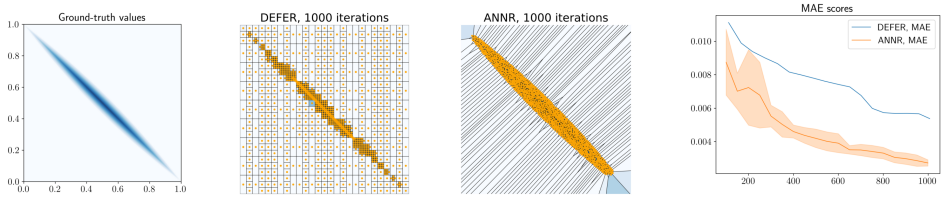


Figure A.12: Approximation of a distance function defined on an intersection of two circles with radius 5 and centered at (-3, -3) and (4, 4).

Rotational Equivariance

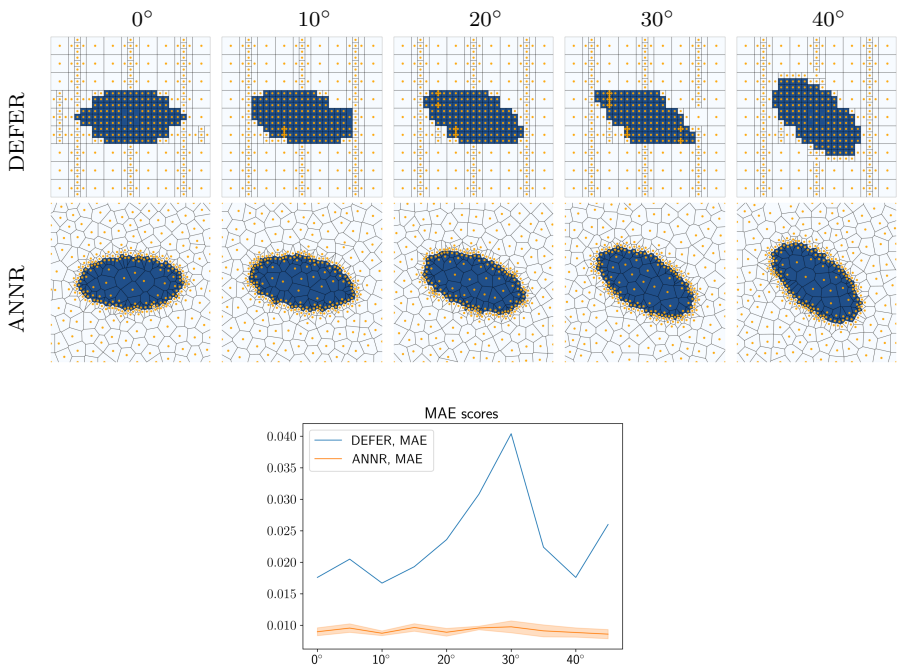


Figure A.13: Top: approximation of an ellipse characteristic function $\mathbf{1}_{x^2+4y^2\leq 1}$ by DEFER and the ANNR with various rotations of the domain ($N = 300$). **Bottom**: performance comparison as the angle varies.

Gravitational Waves

Data preprocessing. We follow the initial setup of the problem described in [8]. Namely, the function is constructed from a gravitational wave example provided tutorial. The six parameters of a gravitational wave generated by a binary black hole that are inferred by the model are: the luminosity distance d_L, the inclination angle ι, the polarization angle ψ, the phase ϕ_c relative to a reference time and two spin magnitudes a_1 and a_2. For more details about the nature of the data we refer to [1].

Evaluation of the logarithm of the approximated likelihood on several uniformly selected points in the domain yields values distributed around -8000 (the simulator provides log-likelihood). Since our aim is to approximate the original likelihood function, we add 8000 to all log-density evaluations and then exponentiate the values, effectively multiplying the likelihood function by e^{8000}. This operation brings the majority of output values of the underlying function close to single digits, stabilizing the computations. Such scaling does not affect the baseline as DEFER is invariant to such transformations, and multiplicatively affects our choice of the lifting parameter.

Volume clipping. While the approximated density function may appear to take relatively low values over the large part of the domain, some of the concentrated areas may produce values many orders of magnitude higher than that. Without any Lipschitz guarantees for the underlying function, such areas could create attractors for the ANNR, forcing the method's exploration to 'sink' in such singularities and over-exploit the area. In order to mitigate that, we propose to truncate the scores of simplices in accordance to a pre-selected sensible Lipschitz constant.

Consider a simplex $\sigma \in \mathrm{Del}_P$ and its lifting $\hat{\sigma}$ and note that $\mathrm{Vol}(\hat{\sigma}) = \frac{1}{\cos\alpha}\mathrm{Vol}(\sigma)$, where α is the *dihedral angle* between σ and $\hat{\sigma}$ both naturally embedded in $\mathbb{R}^{m+1}$. Limiting the Lipschitz constant is equivalent to limiting the maximal dihedral angle to some α_0. As the result of the clipping, $\mathrm{Vol}(\hat{\sigma})$ in Algorithm A.1 gets transformed into $\min\{\mathrm{Vol}(\hat{\sigma}), \frac{1}{\cos\alpha_0}\mathrm{Vol}(\sigma)\}$. In our experiments with gravitational waves, we use $\alpha_0 = 89°$.

Marginals. FigureA.14 presents the marginalizations of the approximated function over all possible two-parameter slices.

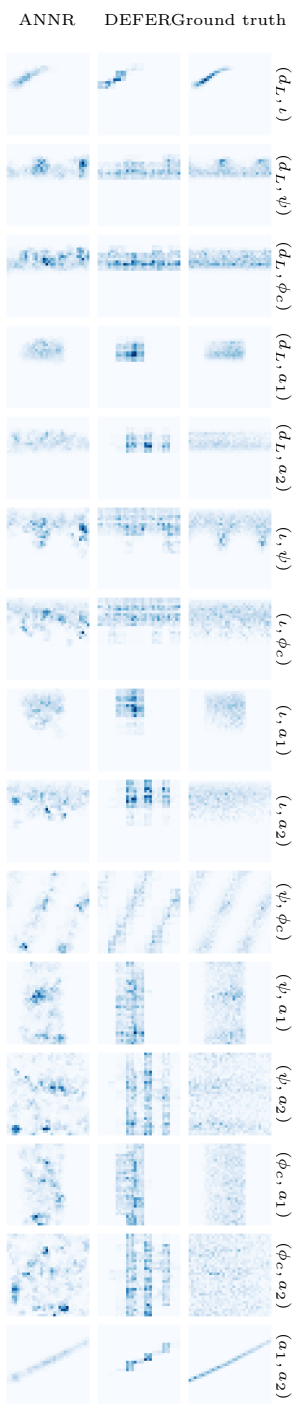


Figure A.14: Marginal distribution for gravitational waves over all two-dimensional slices of parameters. Each slice is identified by a pair of parameters named as in [8].

References

[1] G. Ashton, M. Hübner, P. D. Lasky, C. Talbot, K. Ackley, S. Biscoveanu, Q. Chu, A. Divakarla, P. J. Easter, B. Goncharov, F. H. Vivanco, J. Harms, M. E. Lower, G. D. Meadors, D. Melchor, E. Payne, M. D. Pitkin, J. Powell, N. Sarin, R. J. E. Smith, and E. Thrane, "Bilby: A user-friendly bayesian inference library for gravitational-wave astronomy," *The Astrophysical Journal Supplement Series*, vol. 241, no. 2, p. 27, 2019.

[2] E. Thrane and C. Talbot, "An introduction to bayesian inference in gravitational-wave astronomy: Parameter estimation, model selection, and hierarchical models," *Publications of the Astronomical Society of Australia*, vol. 36, p. e010, 2019.

[3] K. Gubaev, E. V. Podryabinkin, and A. V. Shapeev, "Machine learning of molecular properties: Locality and active learning," *The Journal of chemical physics*, vol. 148, no. 24, p. 241727, 2018.

[4] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Big data meets quantum chemistry approximations: the δ-machine learning approach," *Journal of chemical theory and computation*, vol. 11, no. 5, pp. 2087–2096, 2015.

[5] K. Sung and P. Niyogi, "Active learning for function approximation," in *Advances in Neural Information Processing Systems* (G. Tesauro, D. Touretzky, and T. Leen, eds.), vol. 7, pp. 593–600, MIT Press, 1995.

[6] M. Järvenpää, M. Gutmann, A. Vehtari, and P. Marttinen, "Efficient acquisition rules for model-based approximate bayesian computation," *Bayesian Analysis*, vol. 14, 06 2019.

[7] K. Kandasamy, J. Schneider, and B. Póczos, "Bayesian active learning for posterior estimation," in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, p. 3605–3611, AAAI Press, 2015.

[8] E. Bodin, Z. Dai, N. Campbell, and C. H. Ek, "Black-box density function estimation using recursive partitioning," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of

Proceedings of Machine Learning Research, pp. 1015–1025, PMLR, 18–24 Jul 2021.

[9] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[10] G. Devi, C. Chauhan, and S. Chakraborti, "Conceptualizing curse of dimensionality with parallel coordinates," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[11] S. Fortune, "Voronoi diagrams and delaunay triangulations," *Computing in Euclidean geometry*, pp. 225–265, 1995.

[12] V. Polianskii and F. T. Pokorny, "Voronoi graph traversal in high dimensions with applications to topological data analysis and piecewise linear interpolation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2154–2164, 2020.

[13] B. Delaunay *et al.*, "Sur la sphere vide," *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskih i Estestvennykh Nauk*, vol. 7, no. 793-800, pp. 1–2, 1934.

[14] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation," *Computing in Euclidean geometry*, vol. 1, pp. 23–90, 1992.

[15] L. Chen, "Mesh smoothing schemes based on optimal delaunay triangulations.," in *IMR*, pp. 109–120, Citeseer, 2004.

[16] H. Edelsbrunner and J. Harer, *Computational topology: an introduction*. American Mathematical Soc., 2010.

[17] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation," *Computational geometry*, vol. 22, no. 1-3, pp. 21–74, 2002.

[18] J. Ruppert, "A delaunay refinement algorithm for quality 2-dimensional mesh generation," *Journal of algorithms*, vol. 18, no. 3, pp. 548–585, 1995.

[19] L. P. Chew, "Guaranteed-quality mesh generation for curved surfaces," in *Proceedings of the ninth annual symposium on Computational geometry*, pp. 274–280, 1993.

[20] A. Kontorovich, S. Sabato, and R. Urner, "Active nearest-neighbor learning in metric spaces," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, (Red Hook, NY, USA), p. 856–864, Curran Associates Inc., 2016.

[21] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[22] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[23] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR'94*, pp. 3–12, Springer, 1994.

[24] D. J. C. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.

[25] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, no. 2, pp. 201–221, 1994.

[26] E. B. Baum and K. Lang, "Query learning can work poorly when a human oracle is used," in *International joint conference on neural networks*, vol. 8, p. 8, 1992.

[27] J.-J. Zhu and J. Bento, "Generative adversarial active learning," *arXiv preprint arXiv:1702.07956*, 2017.

[28] M. Järvenpää, A. Vehtari, and P. Marttinen, "Batch simulations and uncertainty quantification in gaussian process surrogate approximate bayesian computation," in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)* (J. Peters and D. Sontag, eds.), vol. 124 of *Proceedings of Machine Learning Research*, pp. 779–788, PMLR, 03–06 Aug 2020.

[29] L. Acerbi, "Variational bayesian monte carlo," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, (Red Hook, NY, USA), p. 8223–8233, Curran Associates Inc., 2018.

[30] E. Higson, W. Handley, M. P. Hobson, and A. N. Lasenby, "Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation," *Statistics and Computing*, pp. 1–23, 2019.

[31] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable gps," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4405–4423, 2020.

[32] D. M. Y. Sommerville, *An Introduction to the Geometry of n Dimensions*, vol. 512. Dover New York, 1958.

[33] A. Bondu, V. Lemaire, and M. Boullé, "Exploration vs. exploitation in active learning : A bayesian approach," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2010.

[34] C. Kimberling, *Triangle centers and central triangles*. Utilitas Mathematica Publishing Incorporated, 1998.

[35] V. Klee, "On the complexity of d-dimensional voronoi diagrams.," tech. rep., Washington Univ. Seattle Dept. of Mathematics, 1979.

[36] A. Rushdi, L. P. Swiler, E. T. Phipps, M. D'Elia, and M. S. Ebeida, "Vps: Voronoi piecewise surrogate models for high-dimensional data fitting," *International Journal for Uncertainty Quantification*, vol. 7, no. 1, 2017.

[37] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[38] O. Devillers, S. Pion, and M. Teillaud, "Walking in a triangulation," in *Proceedings of the seventeenth annual symposium on Computational geometry*, pp. 106–114, 2001.

[39] A. Blum, J. Hopcroft, and R. Kannan, *Foundations of data science*. Cambridge University Press, 2020.

[40] J. S. Speagle, "dynesty: a dynamic nested sampling package for estimating Bayesian posteriors and evidences," *Monthly Notices of the Royal Astronomical Society*, vol. 493, pp. 3132–3158, 02 2020.

[41] G. Arvanitidis, L. K. Hansen, and S. Hauberg, "Latent space oddity: on the curvature of deep generative models," in *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings*, 2018.

[42] G. Arvanitidis, S. Hauberg, and B. Schölkopf, "Geometrically enriched latent spaces," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics* (A. Banerjee and K. Fukumizu, eds.), vol. 130 of *Proceedings of Machine Learning Research*, pp. 631–639, PMLR, 13–15 Apr 2021.

[43] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[44] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[45] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[46] S. M. Omohundro, *The Delaunay triangulation and function learning*. International Computer Science Institute, 1989.

[47] C. C. Lalescu, "Two hierarchies of spline interpolations. practical algorithms for multivariate higher order splines," *CoRR*, vol. abs/0905.3564, 2009.

[48] H. Brezis, *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

Paper B

Voronoi Density Estimator for High-Dimensional Data: Computation, Compactification and Convergence

Vladislav Polianskii*, Giovanni Luca Marchetti*, Alexander Kravberg, Anastasiia Varava, Florian T. Pokorny, Danica Kragic

Abstract

The Voronoi Density Estimator (VDE) is an established density estimation technique that adapts to the local geometry of data. However, its applicability has been so far limited to problems in two and three dimensions. This is because Voronoi cells rapidly increase in complexity as dimensions grow, making the necessary explicit computations infeasible. We define a variant of the VDE deemed Compactified Voronoi Density Estimator (CVDE), suitable for higher dimensions. We propose computationally efficient algorithms for numerical approximation of the CVDE and formally prove convergence of the estimated density to the original one. We implement and empirically validate the CVDE through a comparison with the Kernel Density Estimator (KDE). Our results indicate that the CVDE and the KDE are comparable at their best performance and that the CVDE surpasses the KDE under arbitrary bandwidth selection.

B.1 Introduction

Given a discrete set of data sampled from an unknown probability distribution, the aim of density estimation is to recover the underlying Probability Density Function

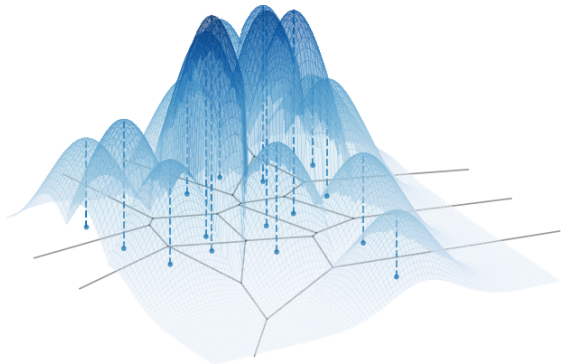


Figure B.1: Graph of a density estimated by the CVDE, with the Voronoi tessellation underneath.

(PDF) [1,2]. Non-parametric methods achieve this by directly computing the PDF through a closed formula, avoiding the potentially expensive need of searching for optimal parameters.

One of the most common non-parametric density estimation techniques is the Kernel Density Estimator (KDE; [3]). The resulting PDF is a convolution between a fixed kernel and the discrete distribution of samples. In case of the Gaussian kernel, this corresponds to a mixture density with a Gaussian distribution centered at each sample. Another popular density estimator, more commonly used for visualization purposes is given by histograms [4], which depend on a prior tessellation of the ambient space (typically, a grid). The estimation is piece-wise constant and is obtained by the number of samples falling in each cell normalised by its volume.

A common limitation of the aforementioned methods is a bias towards a fixed local geometry. Namely, estimates through KDE near a sample are governed by the level sets of the chosen kernel. In the Gaussian case, such level sets are ellipsoids of high estimated probability. Histograms suffer from an analogous bias towards the geometry of the cells of the tessellation (i.e., the bins of the histograms), on which the estimated PDF is constant. The issue of geometrical bias severely manifests when considering real-world high-dimensional data. Indeed, one cannot expect to approximate the rich local geometries of complex data with a simple fixed one. Both the estimators come with hyperparameters controlling the scale of the local geometries which require tuning. This amounts to the bandwidth for KDE and the diameter of the cells for histograms.

The *Voronoi Density Estimator* (VDE) has been suggested to tackle the challenges discussed above [5]. By considering the Voronoi tessellation generated by

data [6], the estimated PDF is piece-wise constant on the cells and proportional to their inverse volume. The Voronoi tessellation adapts local polytopes so that each datapoint is equally likely to be the closest when sampling from the resulting PDF. This has enabled successful application of the VDE to geometrically articulated real-world distributions in lower dimensions [7–9].

The goal of the present work is to enable the VDE for high-dimensional scenarios. Although the VDE constitutes a promising candidate due to its local adaptivity, the following aspects have to be addressed:

Computation. The Voronoi cells are arbitrary convex polytopes and their volume is thus challenging to compute explicitly, which yields the necessity for fast approximate computations.

Compactification. Data is often concentrated around low-dimensional submanifolds, which makes most of the ambient space empty and several Voronoi cells unbounded, i.e. of infinite volume (see Figure B.3). One still needs to produce a finite estimate on those cells, a process we refer to as 'compactification'.

We propose solutions to the problems above. First, we present efficient algorithmic procedures for volume computation and sampling from the estimated density. We formulate the cell volumes as integrals over a sphere, which can then be approximated by Monte Carlo methods. Furthermore, we propose a sampling procedure for the distribution estimated by the VDE. This consists in randomly traversing the Voronoi cells via a 'hit-and-run' Markov chain [10]. The proposed algorithms are highly parallelizable, allowing efficient computations on the GPU.

In order to compactify the cells, we place a finite measure on each of them by means of a fixed kernel (typically, a Gaussian one), leading to an altered version of the VDE which we refer to as *Compactified Voronoi Density Estimator* (CVDE). Figure B.1 shows an example of an estimate by the CVDE on a simple two-dimensional dataset. All the computational and sampling procedures naturally extend to the CVDE.

A further contribution of the present work is a theoretical proof of **convergence** for the CVDE. Assuming the original density has support in the whole ambient space, we show that the PDF estimated by the CVDE converges (with respect to an appropriate notion for random measures) to the ground-truth one as the number of datapoints increases. The convergence holds without any continuity assumptions on the ground-truth PDF nor on the kernel and does not require the kernel bandwidth to vanish asymptotically. This is in contrast with the convergence properties of the KDE. Due to the aforementioned local geometric bias of the KDE, the bandwidth has to decrease at an appropriate rate in order to amend for the local influence of the kernel and guarantee convergence to the underlying

distribution [11, 12].

Finally, we implement the CVDE in *C++* and parallelize computations via the OpenCL framework. Our code, with a provided Python interface, is publicly available at `https://github.com/vlpolyansky/cvde`.

B.2 Compactified Voronoi Density Estimator

This section presents Voronoi cell compactification and Compactified Voronoi Density Estimator, CVDE. We begin by defining the Voronoi tessellations in a general setting (see [6] for a comprehensive treatment). Suppose that (X, d) is a connected metric space and $P \subseteq X$ is a finite collection of distinct points referred to as *generators*.

Definition B.2.1. The *Voronoi cell*[1] of $p \in P$ is defined as

$$C(p) = \{x \in X \mid \forall q \in P \; d(x, q) \geq d(x, p)\}. \tag{B.1}$$

The Voronoi cells intersect at the boundary and cover the ambient space X. The collection $\{C(p)\}_{p \in P}$ is called *Voronoi tessellation* generated by P. For a point $x \in X$ not on the boundary of any cell, we write $C(x)$ for the unique cell containing it. When $X = \mathbb{R}^n$ with Euclidean distance, the Voronoi cells are convex n-dimensional polytopes which are possibly unbounded.

Assume now that X is equipped with a finite Borel measure denoted by Vol. An additional technical condition is that the boundaries of the Voronoi cells have vanishing measure.

Definition B.2.2. The *Voronoi Density Estimator* (VDE) at a point $x \in X$ is defined almost everywhere as

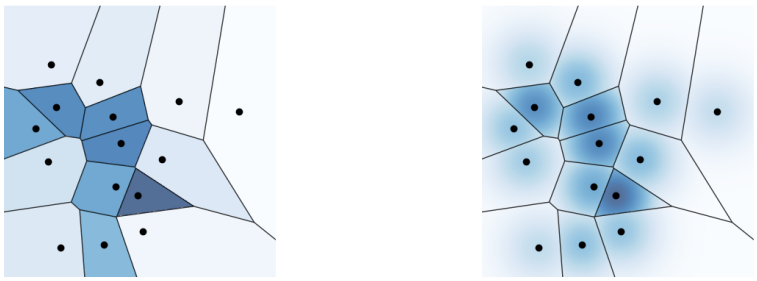
$$\widetilde{f}(x) = \frac{1}{|P| \operatorname{Vol}(C(x))} \tag{B.2}$$

where $|\cdot|$ denotes cardinality.

The function $\widetilde{f}$ defines a locally constant PDF on X and thus a probability measure $\widetilde{f}\operatorname{Vol}$. With respect to this distribution the cells are equally likely, and the restriction to each cell coincides with the normalization of Vol.

We focus on the case where $X = \mathbb{R}^n$ equipped with Euclidean distance. One major issue for the choice of Vol is that the standard Lebesgue measure does not

[1] Sometimes referred to as *Dirichlet cell*.



VDE with bounding square A CVDE with Gaussian kernel

Figure B.2: Comparison between VDE and CVDE for generators in the plane. A darker color represents higher estimated density.

satisfy the finiteness requirement. A common solution in the literature is to restrict the measure to a fixed bounded region $A \subseteq \mathbb{R}^n$ containing P [13,14], which is equivalent to setting $X = A$ as the ambient space. However, this results in an often unsuitable solution for high-dimensional data. Under the manifold hypothesis [15], data are concentrated around a submanifold with high codimension which implies that most of $\mathbb{R}^n$ falls outside the support. Moreover, the cells of the points lying at the boundary of the convex hull of data, which constitute the majority of cells for such submanifolds, are unbounded (see Figure B.3). Estimating the density as uniform, after eventually intersecting with the bounded region A, becomes thus unreasonable and heavily relies on the a priori choice of A.

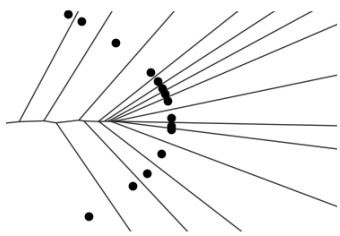


Figure B.3: Voronoi tessellation for generators distributed on a submanifold (a parabola). In this case, all the Voronoi cells are unbounded and the VDE is strongly biased by the choice of the bounding region A.

We instead take a different route. The idea is to make the measure of each cell finite ('compactify') by considering a *local* distribution with mode at the corresponding generator in P. In general terms, we fix a positive kernel $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$

which is at least integrable in the second variable and define the following:

Definition B.2.3. The *Compactified Voronoi Density Estimator* (CVDE) at a point $x \in \mathbb{R}^n$ is defined almost everywhere as

$$f(x) = \frac{K(p,x)}{|P| \operatorname{Vol}_p(C(x))} \tag{B.3}$$

where $\operatorname{Vol}_p(C(x)) = \int_{C(x)} K(p,y)\, \mathrm{d}y$ and p is the generator of $C(x)$ i.e., the generator $p \in P$ closest to x.

In practice, a commonly considered kernel is the Gaussian one

$$K(p,x) = e^{-\frac{\|p-x\|^2}{2h^2}} \tag{B.4}$$

where $h \in \mathbb{R}_{>0}$ is a hyperparameter referred to as 'bandwidth'. More generally, with abuse of notation a kernel can be constructed from an arbitrary integrable map $K \in L^1(\mathbb{R}^n)$:

$$K(p,x) = K\left(\frac{p-x}{h}\right). \tag{B.5}$$

Note that the VDE with a bounding region A corresponds to the particular case of the CVDE with the characteristic function of A as kernel i.e., $K(p,x) = \chi_A(x)$. Figure B.2 shows a comparison between the VDE and the Gaussian CVDE on a simple two-dimensional dataset.

It is worth to briefly compare the CVDE to the Kernel Density Estimator (KDE). Recall that the KDE with kernel K (which is assumed to integrate to 1 in the second variable) is given by $\frac{1}{|P|}\sum_p K(p,x)$. The kernel is aggregated over all the generators, which can possibly oversmooth the estimation. In contrast, the CVDE $f(x)$ involves K evaluated at the closest generator alone. Furthermore, assume that all the cells have the same local volume i.e, $\operatorname{Vol}_p(C(p)) = 1$ for all $p \in P$, and that K monotonically decreases with respect to the distance i.e., $K(p,x) \leq K(p',x)$ when $d(p,x) \geq d(p',x)$. Then the CVDE reduces to

$$f(x) = \frac{1}{|P|} \max_{p \in P} K(p,x) \tag{B.6}$$

which is a variant of the KDE where the sum gets replaced by a maximum. Such distributions are sometimes referred to as 'max-mixtures' [16]. An empirical comparison with KDE is presented in our experimental section (Section B.6.4).

B.3 Algorithmic Procedures

The CVDE presents a number of computational challenges in high dimensions ($n \gg 3$) due to the increasing geometric complexity of Voronoi tessellations. We

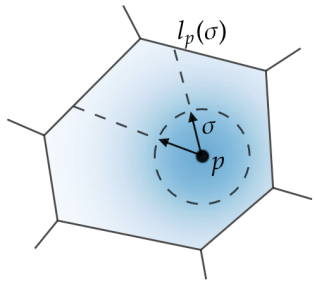


Figure B.4: An illustration of the directional radius involved in volume estimation and sampling.

propose to deploy raycasting methods on polytopes which reduce the problem to one-dimensional subspaces. In the context of Voronoi tessellations raycasting has been considered to explore the boundaries of the cells in [17], which has led to a US Patent [18], as well as in [19]. We utilize these techniques for volume computation and point sampling, and improve the time complexity through pre-computations and parallelization.

We first introduce an algebraic quantity necessary for the subsequent methods. Consider an arbitrary versor σ and a point $z \in \mathbb{R}^n$. Define $l_z(\sigma)$ as the maximum t such that $z + t\sigma$ is contained in $C(z)$, and $l_z(\sigma) = \infty$ if such t does not exist. We refer to this value as a *directional radius*, originating at z in the direction σ (see Figure B.4). The directional radius can be expressed via a closed and computable formula. Denote by p the generator closest to z and for $q \in P \setminus \{p\}$, set

$$l_z^q(\sigma) = \frac{\|q - z\|^2 - \|p - z\|^2}{2\langle \sigma, q - p \rangle}. \tag{B.7}$$

As shown in ([20]), the directional radius is given by

$$l_z(\sigma) = \min_{q \neq p,\ l_z^q(\sigma) \geq 0} l_z^q(\sigma) \tag{B.8}$$

with $l_z(\sigma) = \infty$ if $l_z^q(\sigma)$ is negative for all q.

B.3.1 Volume Estimation and Sampling

We now present a way to efficiently compute the (local) volumes Vol_p via spherical integration. Such an approach to integration over high-dimensional Voronoi tessellations has been explored in the past by [21] and [20].

Assume that the kernel is as in Equation B.5 for a continuous K. By a change of variables into spherical coordinates centered at p and due to convexity of $C(p)$,

the volumes can be rewritten as an integral over the unit sphere $\mathbb{S}^{n-1} \subseteq \mathbb{R}^n$:

$$\mathrm{Vol}_p = \int_{\mathbb{S}^{n-1}} \int_{[0, l_p(\sigma)]} K(t\sigma) t^{n-1} \mathrm{d}t \mathrm{d}\sigma \tag{B.9}$$

where $l_p(\sigma)$ is the directional radius of the cell originating from its generator ($z = p$). The spherical integral can be computed via Monte Carlo approximation by sampling a finite set of versors $\Sigma_p \subseteq \mathbb{S}^{n-1}$ uniformly and estimating the empirical average

$$\frac{2\pi^{\frac{n}{2}}}{|\Sigma_p|\Gamma(\frac{n}{2})} \sum_{\sigma \in \Sigma_p} \int_{[0, l_p(\sigma)]} K(t\sigma) t^{n-1} \mathrm{d}t \tag{B.10}$$

where Γ denotes Euler's Gamma function. In the case of Gaussian kernel (Equation B.4), by bringing the constant $\mathrm{Vol}(\mathbb{S}^{n-1}) = \frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})}$ under the summation the summand simplifies to $(2\pi h^2)^{\frac{n}{2}} \overline{\gamma}\left(\frac{n}{2},\ l_p(\sigma)\right)$, where $\overline{\gamma}$ denotes the regularized lower incomplete Gamma function $\overline{\gamma}(a, z) = \frac{1}{\Gamma(a)} \int_0^z t^{a-1} e^{-t} \mathrm{d}t$.

Next, we propose a sampling procedure for the CVDE which is a version of the *hit-and-run* sampling for distributions on higher-dimensional polytopes [10]. It consists in first choosing a generator $p = z^{(0)} \in P$ uniformly. Then, one traverses the cell $C(p)$ by constructing a Markov chain $\{z^{(i)}\}$ in the following way. A random versor $\sigma^{(i+1)} \in \mathbb{S}^{n-1}$ is sampled uniformly and the next point $z^{(i+1)}$ is sampled from $\frac{1}{\mathrm{Vol}_p} K(p, \cdot)$ restricted to the segment $\{z^{(i)} + t\sigma^{(i+1)} \mid t \in [-l_{z^{(i)}}(-\sigma^{(i+1)}),\ l_{z^{(i)}}(\sigma^{(i+1)})]\}$. As shown by [10], the Markov chain converges w.r.t. total variation distance to the underlying distribution $\frac{1}{\mathrm{Vol}_p} K(p, \cdot)$ over $C(p)$. In practice, one terminates the sampling process after a number I of steps returning the last point $z^{(I)}$. Figure B.5 shows an instance of hit-and-run on a simple two-dimensional dataset.

B.3.2 Computational Complexity

The computational optimizations deserve a separate discussion. As seen from Equations B.8 and B.7, the natural way of estimating the directional radius $l_z(\sigma)$ for given $z \in \mathbb{R}^n$ and $\sigma \in \mathbb{S}^{n-1}$ would require $O(n|P|)$ numerical operations. This would bring the overall computational cost to $O(n \max_p |\Sigma_p||P|^2)$ for the spherical integrals and to $O(n|P|I)$ for a sampling run with I hit-and-run steps.

In order to optimize the algorithms, we first rewrite Equation B.7 as

$$l_z^q(\sigma) = \frac{\langle q, q \rangle - \langle p, p \rangle - 2\langle z, q \rangle + 2\langle z, p \rangle}{2\langle \sigma, q \rangle - 2\langle \sigma, p \rangle}. \tag{B.11}$$

In spherical integration, we deploy the same set of versors $\Sigma = \Sigma_p \subset \mathbb{S}^{n-1}$ for all the generators. This allows to pre-compute $\langle q, p \rangle$ and $\langle \sigma, p \rangle$ for all $p, q \in P, \sigma \in \Sigma$,

Algorithm B.1 Vol_p computation with Gaussian kernel

Input: $P \subset \mathbb{R}^n$ set of generators
$\Sigma \subset \mathbb{S}^{n-1}$ set of versors
Output: Vol_p for all $p \in P$
Compute $\langle q, p\rangle$ **for all** $(q, p) \in P \times P$
Compute $\langle \sigma, p\rangle$ **for all** $(\sigma, p) \in \Sigma \times P$
for all $p \in P$ **do**
Initialize $\mathrm{Vol}_p \leftarrow 0$
for all $\sigma \in \Sigma$ **do**
Initialize $l_p(\sigma) \leftarrow \infty$
for all $q \in P \setminus \{p\}$ **do**
$l_p^q(\sigma) \leftarrow \frac{\langle q,q\rangle - 2\langle q,p\rangle + \langle p,p\rangle}{2\langle \sigma,q\rangle - 2\langle \sigma,p\rangle}$
if $l_p^q(\sigma) > 0$ **then**
$l_p(\sigma) \leftarrow \min\{l_p(\sigma), l_p^q(\sigma)\}$
end if
end for
$\mathrm{Vol}_p \leftarrow \mathrm{Vol}_p + |\Sigma|^{-1}\left(2\pi h^2\right)^{\frac{n}{2}}\ \overline{\gamma}\left(\frac{n}{2},\ l_p(\sigma)\right)$
end for
end for

achieving a total computational complexity of $O(n|P|^2 + n|\Sigma||P| + |\Sigma||P|^2)$.

For the sampling procedure, we similarly fix a prior finite set Σ of all available versors. This does not affect the convergence property of the hit-and-run Markov chain assuming Σ linearly spans $\mathbb{R}^n$ [22]. While $\langle \sigma, p\rangle$ and $\langle q, p\rangle$ can be precomputed in $O(n|P|^2 + n|\Sigma||P|)$ time, the terms involving z in Equation B.11 require more care. To that end, the i-th step of the hit-and-run Markov chain is given by $z^{(i)} = z^{(i-1)} + t^{(i-1)}\sigma^{(i-1)}$ for appropriately sampled $t^{(i-1)}, \sigma^{(i-1)}$. The term $\langle z, p\rangle$ can then be updated inductively in $O(1)$ as $\left\langle z^{(i)}, p\right\rangle = \left\langle z^{(i-1)}, p\right\rangle + t^{(i-1)}\left\langle \sigma^{(i-1)}, p\right\rangle$. Summing up, the cost of a hit-and-run Markov chain run reduces to $O((|\Sigma| + |P|)I)$, which does not depend on the space dimensionality n multiplicatively.

Algorithms B.1 and B.2 provide a more detailed description of volume computation and point sampling via the hit-and-run procedure respectively, including the discussed optimizations. Note that the loops in both algorithms are independent and involve elementary algebraic operations. This allows to utilize GPU capabilities, which also significantly boosts the computation performance.

B.4 Theoretical Properties

B.4.1 Convergence

We now discuss the convergence of the CVDE when the set P of generators is sampled from an underlying distribution. Suppose thus that there is an absolutely continuous probability measure $\mathbb{P} = \rho \mathrm{d}x$ on $\mathbb{R}^n$ defined by a density $\rho \in L^1(\mathbb{R}^n)$.

Algorithm B.2 CVDE sampling

Input: $P \subset \mathbb{R}^n$ set of generators
$\Sigma \subset \mathbb{S}^{n-1}$ set of versors
m desired number of samples
I number of hit-and-run steps
Output: $Z = Z^{(I)} \subset \mathbb{R}^n$ samples from CVDE
Initialize $Z^{(0)} \sim \mathrm{Uni}^m(P)$
Compute $\langle p, p \rangle$ **for all** $p \in P$
Compute $\langle z, p \rangle$ **for all** $(z, p) \in Z^{(0)} \times P$
Compute $\langle \sigma, p \rangle$ **for all** $(\sigma, p) \in \Sigma \times P$
for $i = 1$ **to** I **do**
  for all $z \in Z^{(i-1)}$ **do**
    $\sigma \leftarrow \mathrm{Uni}(\Sigma)$, $p \leftarrow z^{(0)}$
    Initialize $l_z(-\sigma) \leftarrow \infty$, $l_z(\sigma) \leftarrow \infty$
    for all $q \in P \setminus \{p\}$ **do**
      $l_z^q(\sigma) \leftarrow \frac{\langle q,q \rangle - \langle p,p \rangle - 2\langle z,q \rangle + 2\langle z,p \rangle}{2\langle \sigma,q \rangle - 2\langle \sigma,p \rangle}$
      if $l_z^q(\sigma) > 0$ **then**
        $l_z(\sigma) \leftarrow \min\{l_z(\sigma), l_z^q(\sigma)\}$
      else
        $l_z(-\sigma) \leftarrow \min\{l_z(-\sigma), -l_z^q(\sigma)\}$
      end if
    end for
    Sample $t \in [-l_z(-\sigma),\ l_z(\sigma)]$
    Add $z + t\sigma$ to $Z^{(i)}$
    Update $\langle z, p \rangle \leftarrow \langle z, p \rangle + t \langle \sigma, p \rangle$ **for all** $p \in P$
  end for
end for

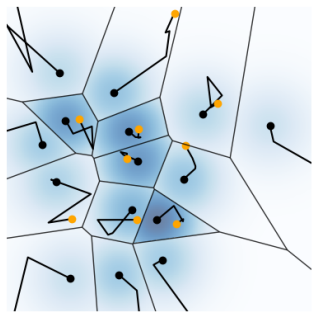


Figure B.5: An illustration of the hit-and-run sampling procedure, with a trajectory of length $I = 4$ for each generator. The sampled points are displayed in orange.

When P is sampled from $\mathbb{P}$ the CVDE can be considered as (the density of) a random probability measure. We denote by $\mathbb{P}_m$ this random measure when the number of generators is m i.e., $\mathbb{P}_m = f\mathrm{d}x$ for $P \sim \rho^m$.

The following is our main theoretical result. It guarantees that $\mathbb{P}_m$ converges to $\mathbb{P}$ with respect to a canonical notion of convergence for random measures, assuming ρ has full support.

Theorem B.4.1. *Suppose that ρ has support in the whole $\mathbb{R}^n$. For any $K \in L^1(\mathbb{R}^n \times \mathbb{R}^n)$ the sequence of random probability measures $\mathbb{P}_m$ converges to $\mathbb{P}$ in distribution w.r.t. x and in probability w.r.t. P. Namely, for any measurable set $E \subseteq \mathbb{R}^n$ the sequence $\mathbb{P}_m(E)$ of random variables converges in probability to the constant $\mathbb{P}(E)$.*

Proof. We outline here an idea of the proof and refer to the Appendix for full details. For a measurable set E, $\mathbb{P}_m(E)$ is equal to

$$\frac{1}{m}|P \cap E| + \text{residue} \tag{B.12}$$

where the residue bounded by (twice) the relative number R of generators whose Voronoi cell intersects the boundary ∂E of E. The variable $\frac{1}{m}|P \cap E|$ tends to $\mathbb{P}(E)$ in probability by the law of large numbers.

We then proceed to show that the boundary term R tends to 0 in probability. To this end, we first prove that the diameters of the Voronoi cells intersecting E tend uniformly to 0, which in turn requires a preliminary result constraining such cells in a neighbour of E (which is assumed to be bounded). Given that, we conclude that R tends to $\mathbb{P}(\partial E)$ by the law of large numbers. By the Portmanteau Lemma [23], we can assume that $\mathbb{P}(\partial E) = 0$ (and that E is bounded), which concludes the proof. □

Note that the above results holds for any (integrable) kernel, thus even for discontinuous ones. The kernel is fixed, and there is no need for an eventual bandwidth (Equation B.5) to vanish asymptotically. This is in contrast with KDE, which requires h to tend to 0 at an appropriate rate in order to obtain convergence to ρ [11, 12]. This is because of the local geometric bias inherent to the KDE, as discussed in Section B.1. In order to obtain convergence, such bias has to be amended with a vanishing bandwidth that annihilates the local geometry of the kernel.

We remark that the assumption on the support of ρ in Theorem B.4.1 is satisfied in the presence of noise, which is realistic in practical scenarios. Assuming that data exhibit, say, Gaussian noise, the actual underlying distribution is of full support even when the ideal one is concentrated on a submanifold of $\mathbb{R}^n$.

B.4.2 Bandwidth Asymptotics

Consider a kernel in the form of Equation B.5. The asymptotics with respect to h (with fixed set of generators P) can be easily deduced:

Proposition B.4.2. *For a continuous $K : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, the following hold:*

- *As h tends to 0, f converges in distribution to the empirical measure $\frac{1}{|P|}\sum_{p\in P}\delta_p$, where delta$_p$ denotes the Dirac's delta centered in p i.e., the probability measure concentrated in the singleton $\{p\}$.*
- *Consider the restriction of the kernel to a bounded region A (i.e., its product with χ_A). As h tends to $+\infty$, f converges in distribution to the VDE $\widetilde{f}$.*

Proof. For the first statement, note that $\frac{1}{h^n}K(\frac{x}{h})$ tends to $K(0)\delta_0$ in distribution by the general theory of approximators of unity. Since $\lim_{h\to 0}\mathrm{Vol}_p(C(x)) = K(0)$ as well for every p, the claim follows from the definition of the CVDE (Equation B.3). As for the second part, observe that $K(x,p)$ tends to $K(0)$ by continuity of K and thus $f(x)$ tends to $\widetilde{f}(x)$ for almost every x. To conclude, pointwise convergence of PDFs implies convergence in distribution (Scheffé's Lemma). □

The asymptotics for small bandwidth are the same as for the KDE. For bandwidth tending to infinity, however, the KDE tends to the uniform distribution over A, while the CVDE still gives reasonable estimates in the form of its non-compactified version.

B.5 Related Work

Non-parametric Density Estimation. The first traces of systematic density estimation date back to the introduction of histograms [24]. Those have been subsequently considered with a variety of cell geometries such as rectangles, triangles [25] and hexagons [26]. The choice of geometry constitutes the main source of bias for the histogram-based density estimator.

Arguably, the most popular density estimator is the KDE, first discussed by [27] and [28]. Numerous extensions have followed, for example, to the multivariate case [29, 30], bandwidth selection methods [31, 32] and algorithms for adaptive bandwidths [33, 34]. The latter aim to partially amend for the local geometric bias of the KDE, which is in line with the present work. However, adapting the bandwidth alone provides a partial solution since it enables different scales of the same local geometry. Among applications, the KDE has been deployed to estimate traffic incidents [35], archeological data [36] and wind speed [37] to name a few.

VDE and its Applications. The VDE has been originally introduced by [5] under the name 'ideal estimator' because of its local geometric adaptivity. Subsequent works have discussed regularization [13] and lower-dimensional aspects [14]. The VDE has seen applications to a variety of real-world densities such as neurons in the brain [7], photons [8] and stars in a galaxy [9]. Although promising, the VDE

has been previously limited to low-dimensional problems.

Theoretical Convergence. Convergence of the VDE has been previously considered in the literature, usually in the language of Poisson point processes. For uniform underlying distribution, pointwise convergence of the averaged estimated density (i.e., unbiasedness: $\lim_{m\to\infty} \mathbb{E}_{P\sim\rho^m}[\widehat{f}(x)] = \rho(x)$ for almost all x) has been proven by [38]. For non-uniform distributions, the same convergence has been shown by [13] with strong continuity assumptions on the density, which allows a reduction to the uniform case. Our theoretical result is based on a different, non-averaged notion of convergence and holds for the more general CVDE with no continuity assumptions.

B.6 Experiements

B.6.1 Dataset Description

In our experiments, we evaluate the CVDE on datasets of different nature: simple *synthetic distributions* of Gaussian type, *image data* in pixel-space, and *sound data* in a frequency space. The datasets we deploy are the following:

Gaussians and Gaussian Mixtures: for synthetic experiments we generate two types of datasets, each containing 1000 training and 1000 test points. The first one consists of samples from an n-dimensional standard Gaussian distribution. The second one is sampled from a Gaussian mixture density $\rho = \frac{1}{2}(\rho_1 + \rho_2)$. Here, ρ_1, ρ_2 are Gaussian distributions with means $\mu_1 = (-0.5, 0, \cdots, 0)$, $\mu_2 = (0.5, 0, \cdots, 0)$ and standard deviations $\sigma_1 = 0.1$, $\sigma_2 = 100$ respectively.

MNIST [39]: the dataset consists of 28×28 grayscale images of handwritten digits which are normalised in order to lie in $[0, 1]^{28\times 28}$. For each experimental run, we sample half of the 60000 training datapoints in order to evaluate the variance of the estimation. The test set size is 10000.

Anuran Calls [40]: the datasets consists of 7195 calls from 10 species of frogs which are represented by 21 normalised mel-frequency cepstral coefficients in $[0, 1]^{21}$. We retain 10% of data for testing and again sample half of the training data at each experimental run.

B.6.2 Comparison with VDE

In this section, we evaluate empirically the necessity of compactification for high-dimensional data. To this end, we visually compare samples from the CVDE (with Gaussian kernel) and from the the VDE. The VDE is implemented with a bounding hypercube $A = [-\frac{7}{2}, \frac{7}{2}]^n$ as described in Section B.2.

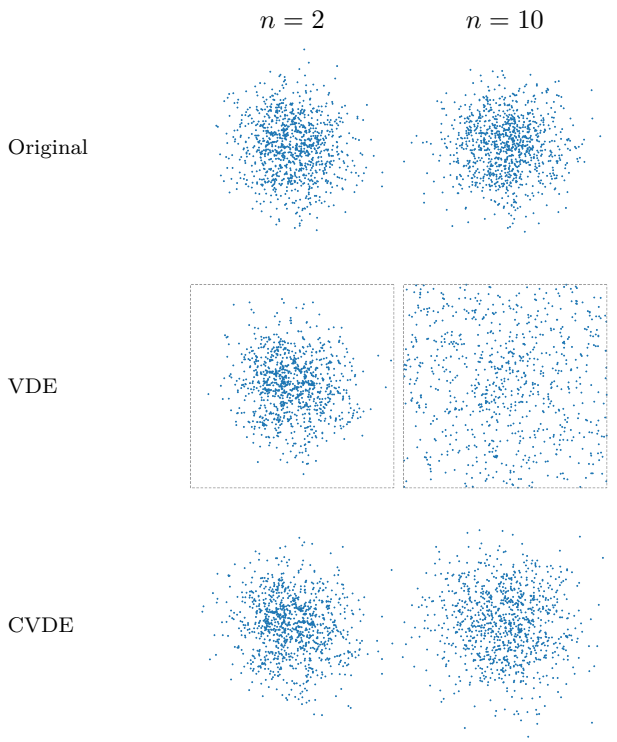
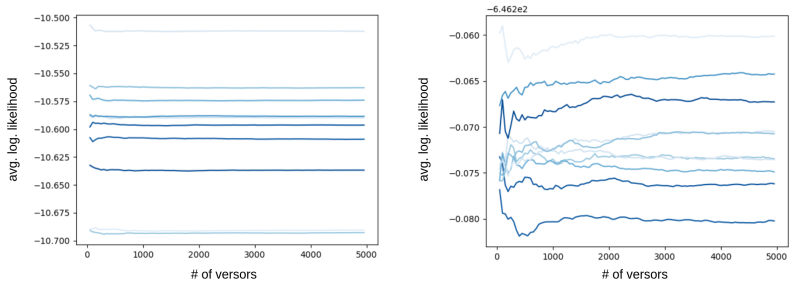


Figure B.6: Visual comparison between samples from the CVDE and the VDE estimating an n-dimensional Gaussian for $n = 2, 10$. In the 10-dimensional case, points are projected onto a plane. In high dimensions, the VDE appears as biased towards a uniform distribution. This is because of abundance of unbounded cells, over which the estimated density is constant.

We consider the Gaussian dataset in $n = 2$ and $n = 10$ dimensions. For both the estimators, 1000 points are sampled via hit-and-run (with trajectories of length $I = 1000$) from the estimated density. The bandwidth for the CVDE is chosen following Scott's rule [2] and amounts to $h = 0.33$ in two dimensions and to 0.66 in ten dimensions.

The results are presented in Figure B.6. In two dimensions, both the estimators produce samples that are visually close to the ground-truth distribution. However, in ten dimensions the sampling quality of VDE drastically decreases, while the CVDE still produces a satisfactory result. In the provided examples, more than 85% of points sampled from the VDE belong to the Voronoi cells intersecting the boundary of A. Since the VDE is uniform within each cell, the estimation and the consequent sampling is biased by the choice of the bounding region A, especially in high dimensions.



10-dimensional Gaussian MNIST

Figure B.7: Stabilization of the Monte Carlo spherical integral. The plots display the average log-likelihood of the estimated density on the training set as the number of sampled versors increases. For each of the 2 datasets, 10 experimental runs are shown.

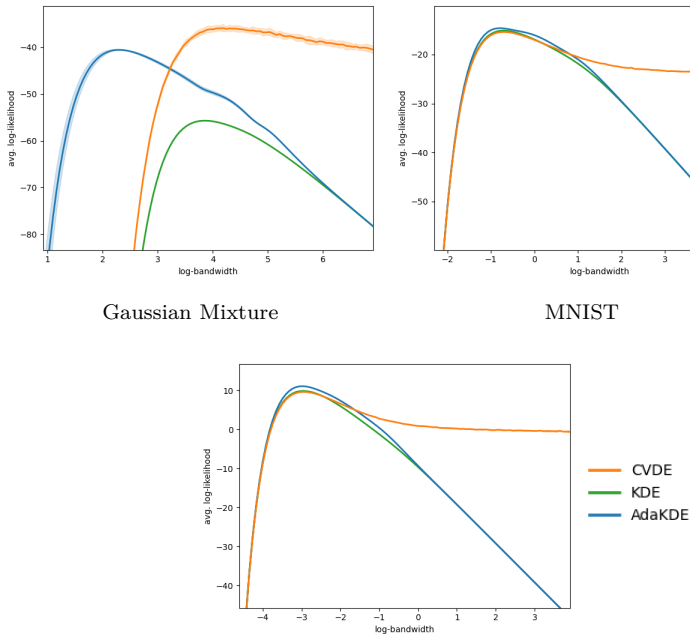
B.6.3 Convergence of the Spherical Integral

We now empirically estimate the amount of Monte Carlo samples required for spherical integration (Equation B.10). To this end, we visualize how the approximation for the volumes in the CVDE (with Gaussian kernel) changes as the number $|\Sigma|$ of versors increases. We consider two datasets: the 10-dimensional Gaussian one and MNIST. Each plot in Figure B.7 displays 10 curves, each corresponding to one experimental run. What is shown is the average log-likelihood of the estimated density on the training set, which correponds up to an additive constant to the average negative logarithmic volume $-\frac{1}{|P|}\sum_{p\in|P|}\log \mathrm{Vol}_p(C(p))$ of the Voronoi cells. The bandwidth is again chosen according to Scott's rule for the Gaussian dataset while it is set to 1 for MNIST. Evidently, all the curves are stable at $|\Sigma| = 5000$ sampled versors, which we fix as a parameter in later experiments.

B.6.4 Comparison with KDE

We now compare the CVDE with the KDE (both with Gaussian kernel) on the synthetic and real-world data described in Section B.6.1. However, the distribution of high-dimensional real-world data is too sparse in the original ambient space to allow for a meaningful comparison. We consequently pre-process the MNIST and the Anuran Calls datasets via Principal Component Analysis (PCA) and orthogonally project them to the 10-dimensional subspace with largest variance. We set the dimension of the synthetic Gaussian mixture to 10 as well.

We compare the CVDE with the standard KDE as well as the KDE with local, adaptive bandwidths (AdaKDE) described in [33]. In the AdaKDE the bandwidth h_p depends on $p \in P$ and is smaller when data is denser around p. Specifically, denote by $\hat{f}(p)$ the standard KDE estimate with a global bandwidth h. Then



Gaussian Mixture MNIST

Anuran Calls

Figure B.8: Empirical comparisons between the CVDE, the KDE and the KDE with adaptive bandwidth (AdaKDE). The plots display the average log-likelihood over the test set as the bandwidth varies. The shadowed region represents standard deviation (with respect to sampling of the dataset) on 5 experimental runs.

$h_p = h\lambda_p$ where $\lambda_p = (g/\hat{f}(p))^{\frac{1}{2}}$ and $g = \prod_{q \in P} \hat{f}(q)^{\frac{1}{|P|}}$.

We score the estimators via the average log-likelihood on a test set P_{test} i.e., $\frac{1}{|P_{\text{test}}|} \sum_{p \in P_{\text{test}}} \log f(p)$. Such score measures the adherence of the estimated density to the ground-truth one and penalizes overfitting thanks to the deployment of the test set.

The results are displayed in Figure B.8 with the bandwidth varying for all the estimators on a logarithmic scale. For AdaKDE we vary the global bandwidth for $\hat{f}$. Sampling of training and test data is repeated for 5 experimental runs, from which mean and standard deviation of the score are displayed.

As can be seen, on the synthetic dataset (Gaussian Mixture) the CVDE outperforms the baselines at the respective best bandwidth. This shows that the local geometric adaptivity of the CVDE leads to density estimates that are closer to the ground-truth distribution. While AdaKDE outperforms the KDE due to its

adaptivity, it still suffers from bias due to the Gaussian kernel (albeit with a local bandwidth) as mentioned in Section B.5. On the real-world datasets (MNIST and Anuran Calls) all the considered estimators exhibit a comparable best performance. We hypothesize that the reason behind this is that on small bandwidths the geometry of the kernel outweighs the adaptivity of the estimators. However, the CVDE outperforms the baselines on larger bandwidths. This is consistent with the discussion in Section B.4.2: the CVDE has better asymptotics than the KDE since it tends to the VDE while the KDE degenerates to a uniform estimate.

B.7 Conclusions and Future Work

In this work, we defined an extension of the Voronoi Density Estimator suitable for high-dimensional data, providing efficient methods for approximate computation and sampling. Additionally, we proved convergence to the underlying data density.

A promising line of future research lies in exploring both theory and applications of the VDE and CVDE to metric spaces beyond the Euclidean one, in particular higher-dimensional Riemannian manifolds. Spheres, for example, naturally appear in the context of normalised data, while complex projective spaces of arbitrary dimension arise as Kendall shape spaces on the plane [41].

B.8 Acknowledgements

This work was supported by the Swedish Research Council, the Knut and Alice Wallenberg Foundation and the European Research Council (ERC-BIRD-884807).

B.9 Appendix

We provide here a proof of our main theoretical result with full details.

Theorem B.9.1. *Suppose that ρ has support in the whole $\mathbb{R}^n$. For any $K \in L^1(\mathbb{R}^n \times \mathbb{R}^n)$ the sequence of random probability measures $\mathbb{P}_m = f\mathrm{d}x$ defined by the CVDE with m generators converges to $\mathbb{P}$ in distribution w.r.t. x and in probability w.r.t. P. Namely, for any measurable set $E \subseteq \mathbb{R}^n$ the sequence $\mathbb{P}_m(E)$ of random variables over P sampled from ρ converges in probability to the constant $\mathbb{P}(E)$.*

We shall first build up some machinery necessary for the proof. First of all, the following fact on higher-dimensional Euclidean geometry will come in hand.

Proposition B.9.2 ([42], Lemma 5.3)**.** *Let $x \in \mathbb{R}^n$, $\delta > 0$. There exist constants $1 < c_1 < c_2 - 1 < 31$ such that for any open cone $K \subseteq \mathbb{R}^n$ centered at x of solid angle $\frac{\pi}{12}$ and any $p, q, z \in K$, if*

$$d(x,p) < \delta,\ c_1\delta \leq d(x,q) < c_2\delta,\ d(x,z) \geq 32\delta$$

then $d(z,q) < d(z,p)$.

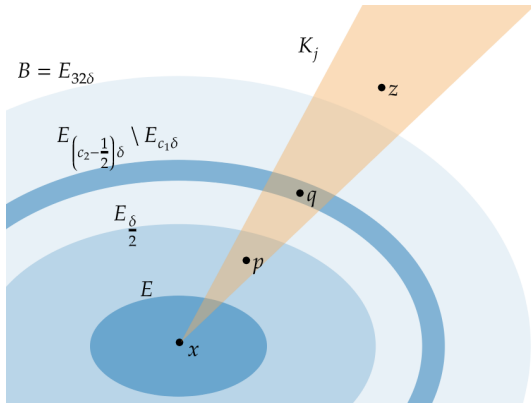


Figure B.9: Graphical depiction of sets and points appearing in the proof of Proposition B.9.3.

We can now deduce the following.

Proposition B.9.3. *Let* $\emptyset \neq E \subseteq \mathbb{R}^n$ *be a bounded measurable set. There exists a bounded measurable set* $B \supseteq E$ *such that as* $m = |P|$ *tends to* ∞*, the probability with respect to* $P \sim \rho^m$ *that every Voronoi cell intersecting* E *is contained in* B *tends to* 1.

Proof. Let $\delta = 2\text{diam}\, E = 2\sup_{x,y\in E} d(x,y)$ be twice the diameter of E. For $L > 0$, consider the L-neighbourhood of E

$$E_L = \{x \in X \mid d(x,E) < L\}.$$

First of all, if E has vanishing measure, we can replace it without loss of generality by some E_L, which has nonempty interior.

We claim that $B = E_{32\delta}$ is as desired. To see that, consider an arbitrary $x \in E$ and let $\{K_j\}_j$ be a finite minimal set of open cones centered at x of solid angle $\frac{\pi}{12}$ whose closures cover $\mathbb{R}^n$. As m tends to ∞, since ρ has support in the whole $\mathbb{R}^n$, by the law of large numbers the probability of the following tends to 1:

- P intersects E (recall that E has non-vanishing measure),
- for every j, P intersects $(E_{(c_2-\frac{1}{2})\delta} \setminus E_{c_1\delta}) \cap K_j$, where c_1, c_2 are the constants from Proposition B.9.2.

To prove our claim, we can thus conditionally assume the above. Consider now a Voronoi cell intersecting E and suppose by contradiction that z is an element of

the cell not contained in B. Let $q \in P$ be a generator in $(E_{(c_2-\frac{1}{2})\delta} \setminus E_{c_1\delta}) \cap K_j$ where K_j is the cone containing z. Since P intersects E, the generator p of the cell lies in $E_{\mathrm{diam}(E)} = E_{\frac{\delta}{2}}$ and consequently $d(x,p) < \delta$. If $p \notin K_j$, then one can replace it with its orthogonal projection on the line passing through x and z. The hypotheses of Proposition B.9.2 are then satisfied and we conclude that $d(z,q) < d(z,p)$. This is absurd since p is the generator of $C(z)$. □

For a bounded measurable set $E \subseteq \mathbb{R}^n$, denote by

$$D_E = \max_{\substack{p \in P \\ C(p) \cap E \neq \emptyset}} \mathrm{diam}\ C(p)$$

the maximum diameter of a Voronoi cell intersecting E.

Proposition B.9.4. *D_E, thought as a random variable in P, converges in probability to 0 as $m = |P|$ tends to ∞.*

Proof. The proof is inspired by Theorem 4 in [43]. Consider a finite minimal set of open cones $\{K_j\}_j$ centered at 0 of solid angle $\frac{\pi}{12}$ whose closures cover $\mathbb{R}^n$. Then there is a constant $c > 0$ such that for each $p \in P$

$$\mathrm{diam}\ C(p) \leq c \max_j R_{p,j}$$

where $R_{p,j} = \min_{q \in P \cap (p+K_j)} d(p,q)$ denotes the distance from p to its closest neighbour in the cone K_j centered in p (and $R_{p,j} = \infty$ if $P \cap (p + K_j) = \emptyset$). This follows from Proposition B.9.2 applied with $x = p$ to all the cones centered at the generators, with an opportune δ for each of them. For each $\varepsilon > 0$ we thus have an inclusion of events

$$\{D_E > \varepsilon\} \subseteq \left\{ \max_{\substack{p,j \\ C(p) \cap E \neq \emptyset}} R_{p,j} > \frac{\varepsilon}{c} \right\} \subseteq$$
$$\subseteq \bigcup_{i,j} \left\{ P \cap (p_i + K_j) \cap B\left(p_i, \frac{\varepsilon}{c}\right) = \emptyset \text{ and } C(p_i) \cap E \neq \emptyset \right\}$$

where $B(x,r)$ is the open ball centered in x of radius r. In the above, we assumed that the set P is equipped with an ordering. For $x \in \mathbb{R}^n$ denote by $E_{x,j}$ the event appearing at the right member of the above expression for $x = p_i$. We can then bound the probability with respect to a random $P \sim \rho^m$, with $m = |P|$ fixed, as

$$\mathbb{P}_{P \sim \rho^m}(D_E > \varepsilon) \leq \sum_{i,j} \mathbb{P}_{P \sim \rho^m}(E_{p_i,j}) = m \sum_j \int_{\mathbb{R}^n} \rho(x) \mathbb{P}_{P \sim \rho^m}(E_{x,j} \mid p_1 = x)\ \mathrm{d}x.$$

Since the points in P are sampled independently we have

$$\mathbb{P}_{P\sim\rho^m}(E_{x,j} \mid p_1 = x,\ C(x)\cap E \neq \emptyset) = \left(1 - \mathbb{P}\left((x+K_j)\cap B\left(x, \frac{\varepsilon}{c}\right)\right)\right)^{m-1} := \\ := (1 - M(x))^{m-1}.$$

Pick the set B guaranteed by Proposition B.9.3. We can then conditionally assume that every Voronoi cell intersecting E is contained in B, which implies $\mathbb{P}_{P\sim\rho^m}(E_{x,j}) = 0$ for $x \notin B$. The limit we wish to estimate reduces to

$$\lim_{m\to\infty} m\sum_j \int_{\mathbb{R}^n} \rho(x)\mathbb{P}_{P\sim\rho^m}(E_{x,j} \mid p_1 = x)\,\mathrm{d}x = \sum_j \lim_{m\to\infty} \int_B \rho(x)m(1-M(x))^{m-1}\,\mathrm{d}x.$$

Since B is bounded and ρ has support in the whole $\mathbb{R}^n$, $M(x)$ is (essentially) bounded from below by a strictly positive constant as x varies in B. The limit can thus be brought under the integral and putting everything together we get:

$$\lim_{m\to\infty} \mathbb{P}_{P\sim\rho^m}(D_E > \varepsilon) \leq \sum_j \int_B \rho(x) \lim_{m\to\infty} m(1-M(x))^{m-1}\,\mathrm{d}x = 0.$$

□

We are now ready to prove Theorem B.9.1.

Proof. By the Portmanteau Lemma ([23]), it is sufficient to that $\mathbb{P}_m(E)$ converges to $\mathbb{P}(E)$ in probability for any bounded measurable set $E \subseteq \mathbb{R}^n$ which is a continuity set for $\mathbb{P}$ i.e., $\mathbb{P}(\partial E) = 0$ where ∂E is the (topological) boundary of E. Pick such E. By definition of the CVDE, for a fixed set P of generators we have that

$$\mathbb{P}_m(E) = \frac{1}{m}|\{p \in P \mid C(p) \subseteq E\}| + \overbrace{\frac{1}{m}\sum_{\substack{p\in P\\ C(p)\not\subseteq E\\ C(p)\cap E\neq\emptyset}} \frac{\mathrm{Vol}_p(C(p)\cap E)}{\mathrm{Vol}_p(C(p))}}^{\overline{R}} \\ = \frac{1}{m}|P\cap E| + \overline{R} - \frac{1}{m}|\{p \in P\cap E \mid C(p) \not\subseteq E\}|.$$

Since the Voronoi cells are closed, any cell intersecting E not contained in E intersects ∂E. Thus $\left|\overline{R} - \frac{1}{m}|\{p \in P\cap E \mid C(p) \not\subseteq E\}|\right| \leq 2R$ where $R := \frac{1}{m}|\{p \in P \mid C(p)\cap\partial E \neq \emptyset\}|$. Now, the random variable $\frac{1}{m}|P\cap E|$ tends to $\mathbb{P}(E)$ in probability as m tends to ∞ by the law of large numbers. In order to conclude, we need to show that R tends to 0 in probability.

Fix $\varepsilon > 0$. For $L > 0$, consider the L-neighbor $\partial E_L = \{x \in X \mid d(x,\partial E) < L\}$ of the boundary ∂E. If the diameter of the Voronoi cells intersecting ∂E is less than L then all such cells are contained in ∂E_L. Thus:

$$\begin{aligned}\mathbb{P}_{P\sim\rho^m}(R>\varepsilon) &\leq \mathbb{P}_{P\sim\rho^m}\left(\frac{1}{m}|P\cap\partial E_L|>\varepsilon \text{ and } D_{\partial E}<L\right)+\mathbb{P}_{P\sim\rho^m}(D_{\partial E}\geq L)\\ &\leq \mathbb{P}_{P\sim\rho^m}\left(\frac{1}{m}|P\cap\partial E_L|>\varepsilon\right)+\mathbb{P}_{P\sim\rho^m}(D_{\partial E}\geq L)\\ &\leq \mathbb{P}_{P\sim\rho^m}\left(\left|\mathbb{P}(\partial E_L)-\frac{1}{m}|P\cap\partial E_L|\right|>\varepsilon-\mathbb{P}(\partial E_L)\right)+\\ &+\mathbb{P}_{P\sim\rho^m}(D_{\partial E}\geq L)\,.\end{aligned}$$

Since ∂E is closed, $\partial E=\cap_{L>0}\partial E_L$ and thus $\lim_{L\to 0}\mathbb{P}(\partial E_L)=\mathbb{P}(\cap_L\partial E_L)=\mathbb{P}(\partial E)=0$ since E is a continuity set. This implies that there is an L such that $\varepsilon>\mathbb{P}(\partial E_L)$. The right hand side of the Equation above tends then to 0 by the law of large numbers and Proposition B.9.4, which concludes the proof.

□

References

[1] P. J. Diggle, *Statistical analysis of spatial and spatio-temporal point patterns*. CRC press, 2013.

[2] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[3] A. Gramacki, *Nonparametric kernel density estimation and its computational aspects*. Springer, 2018.

[4] D. Freedman and P. Diaconis, "On the histogram as a density estimator: L 2 theory," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 57, no. 4, pp. 453–476, 1981.

[5] J. Ord, "How many trees in a forest," *Mathematical Scientist*, vol. 3, pp. 23–33, 1978.

[6] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*, vol. 501. John Wiley & Sons, 2009.

[7] C. Duyckaerts, G. Godefroy, and J.-J. Hauw, "Evaluation of neuronal numerical density by dirichlet tessellation," *Journal of neuroscience methods*, vol. 51, no. 1, pp. 47–69, 1994.

[8] H. Ebeling and G. Wiedenmann, "Detecting structure in two dimensions combining voronoi tessellation and percolation," *Physical Review E*, vol. 47, no. 1, p. 704, 1993.

[9] I. Vavilova, A. Elyiv, D. Dobrycheva, and O. Melnyk, "The voronoi tessellation method in astronomy," in *Intelligent Astrophysics*, pp. 57–79, Springer, 2021.

[10] M.-H. Chen and B. W. Schmeiser, "General hit-and-run monte carlo sampling for evaluating multidimensional integrals," *Operations Research Letters*, vol. 19, no. 4, pp. 161–169, 1996.

[11] L. Devroye and T. Wagner, "The l1 convergence of kernel density estimates," *The Annals of Statistics*, pp. 1136–1139, 1979.

[12] H. Jiang, "Uniform convergence rates for kernel density estimation," in *International Conference on Machine Learning*, pp. 1694–1703, PMLR, 2017.

[13] M. M. Moradi, O. Cronie, E. Rubak, R. Lachieze-Rey, J. Mateu, and A. Baddeley, "Resample-smoothing of voronoi intensity estimators," *Statistics and computing*, vol. 29, no. 5, pp. 995–1010, 2019.

[14] C. D. Barr and F. P. Schoenberg, "On the voronoi estimator for the intensity of an inhomogeneous planar poisson process," *Biometrika*, vol. 97, no. 4, pp. 977–984, 2010.

[15] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," *Journal of the American Mathematical Society*, vol. 29, no. 4, pp. 983–1049, 2016.

[16] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 826–840, 2013.

[17] S. A. Mitchell, M. S. Ebeida, M. A. Awad, C. Park, A. Patney, A. A. Rushdi, L. P. Swiler, D. Manocha, and L.-Y. Wei, "Spoke-darts for high-dimensional blue-noise sampling," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 2, pp. 1–20, 2018.

[18] M. S. Ebeida, "Generating an implicit voronoi mesh to decompose a domain of arbitrarily many dimensions," May 28 2019. US Patent 10,304,243.

[19] V. Polianskii and F. T. Pokorny, "Voronoi graph traversal in high dimensions with applications to topological data analysis and piecewise linear interpolation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2154–2164, 2020.

[20] V. Polianskii and F. T. Pokorny, "Voronoi boundary classification: A high-dimensional geometric approach via weighted monte carlo integration," in *International Conference on Machine Learning*, pp. 5162–5170, PMLR, 2019.

[21] N. Winovich, A. Rushdi, E. T. Phipps, J. Ray, G. Lin, and M. S. Ebeida, "Rigorous data fusion for computationally expensive simulations.," tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States); Sandia ..., 2019.

[22] C. J. Bélisle, H. E. Romeijn, and R. L. Smith, "Hit-and-run algorithms for generating multivariate distributions," *Mathematics of Operations Research*, vol. 18, no. 2, pp. 255–266, 1993.

[23] A. W. Van der Vaart, *Asymptotic statistics*, vol. 3. Cambridge university press, 2000.

[24] K. Pearson, "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894.

[25] D. W. Scott, "A note on choice of bivariate histogram bin shape," *Journal of Official Statistics*, vol. 4, no. 1, p. 47, 1988.

[26] D. B. Carr, A. R. Olsen, and D. White, "Hexagon mosaic maps for display of univariate and bivariate geographical data," *Cartography and Geographic Information Systems*, vol. 19, no. 4, pp. 228–236, 1992.

[27] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[28] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[29] A. J. Izenman, "Review papers: Recent developments in nonparametric density estimation," *Journal of the american statistical association*, vol. 86, no. 413, pp. 205–224, 1991.

[30] K. Dehnad, "Density estimation for statistics and data analysis," 1987.

[31] J. Marron, "A comparison of cross-validation techniques in density estimation," *The Annals of Statistics*, pp. 152–162, 1987.

[32] M. P. Wand, M. C. Jones, *et al.*, "Multivariate plug-in bandwidth selection," *Computational Statistics*, vol. 9, no. 2, pp. 97–116, 1994.

[33] B. Wang and X. Wang, "Bandwidth selection for weighted kernel density estimation," *arXiv preprint arXiv:0709.1616*, 2007.

[34] C. M. van der Walt and E. Barnard, "Variable kernel density estimation in high-dimensional feature spaces," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[35] Z. Xie and J. Yan, "Kernel density estimation of traffic accidents in a network space," *Computers, environment and urban systems*, vol. 32, no. 5, pp. 396–406, 2008.

[36] M. J. Baxter, C. C. Beardah, and R. V. Wright, "Some archaeological applications of kernel density estimates," *Journal of Archaeological Science*, vol. 24, no. 4, pp. 347–354, 1997.

[37] H. Bo, L. Yudun, Y. Hejun, and W. He, "Wind speed model based on kernel density estimation and its application in reliability assessment of generating systems," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 2, pp. 220–227, 2017.

[38] G. Last, "Stationary random measures on homogeneous spaces," *Journal of Theoretical Probability*, vol. 23, no. 2, pp. 478–497, 2010.

[39] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[40] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[41] K. V. Mardia and P. E. Jupp, *Directional statistics*, vol. 494. John Wiley & Sons, 2009.

[42] I. Gibbs and L. Chen, "Asymptotic properties of random voronoi cells with arbitrary underlying density," *Advances in Applied Probability*, vol. 52, no. 2, pp. 655–680, 2020.

[43] L. Devroye, L. Györfi, G. Lugosi, and H. Walk, "On the measure of voronoi cells," *Journal of Applied Probability*, vol. 54, 12 2015.

Paper C

An Efficient and Continuous Voronoi Density Estimator

Giovanni Luca Marchetti, Vladislav Polianskii, Anastasiia Varava, Florian T. Pokorny, Danica Kragic

Abstract

We introduce a non-parametric density estimator deemed Radial Voronoi Density Estimator (RVDE). RVDE is grounded in the geometry of Voronoi tessellations and as such benefits from local geometric adaptiveness and broad convergence properties. Due to its radial definition RVDE is continuous and computable in linear time with respect to the dataset size. This amends for the main shortcomings of previously studied VDEs, which are highly discontinuous and computationally expensive. We provide a theoretical study of the modes of RVDE as well as an empirical investigation of its performance on high-dimensional data. Results show that RVDE outperforms other non-parametric density estimators, including recently introduced VDEs.

C.1 Introduction

The problem of estimating a Probability Density Function (PDF) from a finite set of samples lies at the heart of statistics and arises in several practical scenarios [1, 2]. Among density estimators, the non-parametric ones aim to infer a PDF through a closed formula. Differently from parametric methods, they do not require optimization and ideally provide an estimated PDF which is simple, interpretable and computationally efficient. Two traditional examples of non-parametric density estimators are the Kernel Density Estimator (KDE; [3, 4]) and histograms [5, 6]. KDE consists of a mixture of local copies of a kernel around each datapoint while histograms partition the ambient space into local cells ('bins') where the estimated

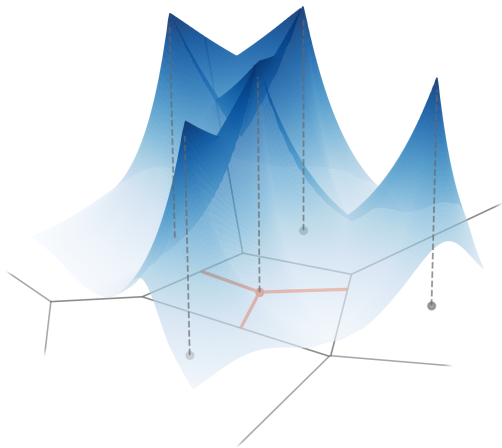


Figure C.1: An example of a density estimated via RVDE. The Voronoi tessellation is depicted in solid gray. The estimated density is defined by the property that its conical integral over the rays originating from the datapoints (orange) is constant.

PDF is constant.

Both histograms and KDE suffer from bias due to the prior choice of a local geometric structure i.e., the bins and the kernel respectively. This bias gets exacerbated in high-dimensional ambient spaces. The reason is that datasets grow exponentially in terms of geometric complexity, making a fixed simple geometry unsuitable for estimating high-dimensional densities. This has led to the introduction of the *Voronoi Density Estimator* (VDE; [7]). VDE relies on the geometric adaptiveness of Voronoi cells, which are convex polytopes defined locally by the data [8]. The PDF estimated by VDE is constant on such cells, thus behaving as an adaptive version of histograms. Due to its local geometric properties, VDE possesses convergence guarantees to the ground-truth PDF which are more general than the ones of KDE.

The geometric benefits of VDE, however, come with a number of shortcomings. First, the Voronoi cells and in particular their volumes are computationally expensive to compute in high dimensions. Although this has been recently attenuated by proposing Monte Carlo approximations [9], VDE falls behind methods such as KDE in terms of computational complexity. Second, VDE (together with its generalized version from [9] deemed CVDE) is highly discontinuous on the boundaries of Voronoi cells. The estimated PDF consequently suffers from large variance and instability with respect to the dataset. This is again in contrast to KDE, which is

continuous in its ambient space.

In this work, we propose a novel non-parametric density estimator deemed *Radial Voronoi Density Estimator* (RVDE) which addresses the above challenges. Similarly to VDE, RVDE integrates to a constant on Voronoi cells and thus shares its local geometric advantages and convergence properties. In contrast to VDE, RVDE is continuous and computable in linear time with respect to the dataset size. The central idea behind RVDE is to define the PDF radially from the datapoints so that the (conical) integral over the ray cast in the corresponding Voronoi cell is constant (see Figure C.1). This is achieved via a 'radial bandwidth' which is defined implicitly by an integral equation. Intuitively, the radial approach reduces the high-dimensional geometric challenge of defining a Voronoi-based estimator to a one-dimensional problem. This avoids the expensive volume computations of the original VDE and guarantees continuity because of the fundamental properties of Voronoi tessellations. Another important aspect of RVDE is its geometric distribution of modes. We show that the modes either coincide with the datapoints or lie along the edges of the Gabriel graph [10] depending on a hyperparameter analogous to the bandwidth in KDE.

We compare RVDE with CVDE, KDE and the adaptive version of the latter in a series of experiments. RVDE outperforms the baselines in terms of the quality of the estimated density on a variety of datasets. Moreover, it runs significantly faster and with lower sampling variance compared to CVDE. This empirically confirms that the geometric and continuity properties of RVDE translate into benefits for the estimated density in a computationally efficient manner. We provide an implementation of RVDE (together with baselines and experiments) in *C++* at a publicly available repository [1]. The code is parallelized via the OpenCL framework and comes with a Python interface. In summary our contributions include:

- A novel density estimator (RVDE) based on the geometry of Voronoi tessellations which is continuous and computationally efficient.
- A complete study of the modes of RVDE and their geometric distribution.
- An empirical investigation comparing RVDE to KDE (together with its adaptive version) and previously studied VDEs.

C.2 Related Work

Non-Parametric Density Estimation. Non-parametric methods for density estimation trace back to the introduction of histograms [6]. Histograms have been extended by considering bin geometries beyond the canonical rectangular one, for

[1] `https://github.com/giovanni-marchetti/rvde`

example triangular [11] and hexagonal [12] geometries. Another popular density estimator is KDE, first discussed by [4] and [13]. The estimated density is a mixture of copies of a priorly chosen distribution ('kernel') centered at the datapoints. KDE has been extended to the multivariate case [14, 15] and has seen improvements such as bandwidth selection methods [16, 17] and algorithms for adaptive bandwidths [18, 19]. Applications of KDE include estimation of traffic incidents [20], of archaeological data [21] and of wind speed [22] to name a few. As discussed in Section C.1, both KDE and histograms suffer from lack of geometric adaptiveness due to the choice of prior local geometries.

Another class of non-parametric methods are the orthogonal density estimators [23, 24]. Those consist of choosing a discretized orthonormal basis of functions and computing the coefficients of the ground-truth density via Monte-Carlo integration over the dataset. When the basis is the Fourier one, the estimator is referred to as 'wavelet estimator'. The core drawback is that orthogonal density estimators do not scale efficiently to higher dimensions. When considering canonical tensor product bases the complexity grows exponentially w.r.t. the dimensionality [25], making the estimator unfeasible to compute.

Voronoi Density Estimators. The first Voronoi Density Estimator (VDE) has been pioneered by [7]. The estimated density relies on Voronoi tessellations in order to achieve local geometric adaptiveness. This is the main advantage of VDE over methods such as KDE. The original VDE has seen applications to real-world densities such as neurons in the brain [26], photons [27] and stars in a galaxy [28]. However, the method is not immediately extendable to high-dimensional spaces because of unfeasible computational complexity of volumes and abundance of unbounded Voronoi cells. This has been only recently amended by [9] by introducing approximate numerical algorithms and by shaping of the density via a kernel. In the present work, we aim to design an alternative version of the original VDE which is continuous and does not rely on volume computations. The resulting estimator is thus stable and computationally efficient while still benefiting from the geometry of Voronoi tessellations.

C.3 Background

In this section we recall the class of non-parametric density estimators which we will be interested in throughout the present work. To this end, let $P \subseteq \mathbb{R}^n$ be a finite set and consider the following central notion from computational geometry.

Definition C.3.1. The *Voronoi cell*[2] of $p \in P$ is:

$$C(p) = \{x \in \mathbb{R}^n \mid \forall q \in P \; d(x,q) \geq d(x,p)\}. \tag{C.1}$$

[2]Sometimes referred to as *Dirichlet cell*.

The Voronoi cells are convex polytopes that intersect at the boundary and cover the ambient space $\mathbb{R}^n$. The collection $\{C(p)\}_{p\in P}$ is referred to as *Voronoi tessellation* generated by P. Note that although the Voronoi tessellations are defined in an arbitrary metric space, the resulting cells might be non-convex for distances different from the Euclidean one. Since convexity will be crucial for the following constructions, we stick to the Euclidean metric for the rest of the work.

We call *density estimator* any mapping associating a probability density function $f_P \in L^1(\mathbb{R}^n)$ to a finite set $P \subseteq \mathbb{R}^n$. The following class of density estimators generalizes the original one by [7].

Definition C.3.2. A *Voronoi Density Estimator* (VDE) is a density estimator $P \mapsto f_P$ such that for each $p \in P$:

$$\int_{C(p)} f_P(x)\ \mathrm{d}x = \frac{1}{|P|}. \tag{C.2}$$

VDEs stand out among density estimators for their geometric properties. This is because the Voronoi cells are arbitrary polytopes that are adapted to the local geometry of data. For VDEs all the Voronoi cells have the same estimated probability, making such estimators locally adaptive from a geometric perspective. This is reflected, for example, by the general convergence properties of VDEs. The following is the main theoretical result from [9].

Theorem C.3.1. *Let $P \mapsto f_P$ be a VDE and suppose that P is sampled from a probability density $\rho \in L^1(\mathbb{R}^n)$ with support in the whole $\mathbb{R}^n$. For P of cardinality m consider the probability measure $\mathbb{P}_m = f_P \mathrm{d}x$ which is random in P. Then the sequence $\mathbb{P}_m$ converges to $\mathbb{P} = \rho \mathrm{d}x$ in distribution w.r.t. x and in probability w.r.t. P. Namely, for any measurable set $E \subseteq \mathbb{R}^n$ the sequence of random variables $\mathbb{P}_m(E)$ converges in probability to the constant $\mathbb{P}(E)$.*

In contrast, the convergence of other density estimators such as KDE requires the kernel bandwidth to vanish asymptotically [29]. The bandwidth vanishing is necessary in order to amend for the local geometric bias inherent in KDE as discussed in Section C.1.

The following canonical construction of a VDE deemed Compactified Voronoi Density Estimator (CVDE) is discussed by [9]. Given an integrable kernel $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{>0}$ the estimated density is defined as

$$f_P(x) = \frac{K(p, x)}{|P|\,\mathrm{Vol}_p(C(p))} \tag{C.3}$$

where p is the closest point in P to x and $\mathrm{Vol}_p(C(p)) = \int_{C(p)} K(p, y)\ \mathrm{d}y$. The latter volumes are approximated via Monte Carlo methods since they become unfeasible to compute exactly as dimensions grow. The resulting density inherits the same

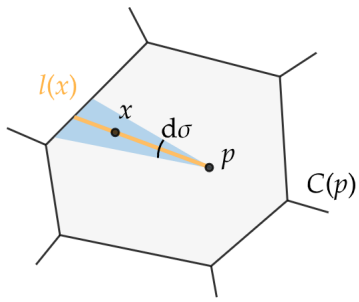


Figure C.2: Illustration of the quantity $l(x)$ involved in the definition of RVDE (Definition C.4.1). The estimated density integrates to a constant over all the infinitesimal cones (blue) in $C(p)$ originating from p.

regularity as K when restricted to each Voronoi cell but jumps *discontinuously* when crossing the boundary of Voronoi cells (see Figure C.3). Motivated by this, the goal of the present work is to introduce a continuous and efficient VDE.

C.4 Method

C.4.1 Radial Voronoi Density Estimator

In this section we outline a general way of constructing a VDE with continuous density function. Our central idea is to define the latter *radially* w.r.t. the datapoints $p \in P$. We start by rephrasing the integral over a Voronoi cell (Equation C.2) in spherical coordinates:

$$\int_{C(p)} f_P(x)\, \mathrm{d}x = \int_{\mathbb{S}^{n-1}} \underbrace{\int_0^{l(p+\sigma)} t^{n-1} f_P(p+t\sigma)\, \mathrm{d}t}_{\text{Conical Integral}}\, \mathrm{d}\sigma. \tag{C.4}$$

Here $\mathbb{S}^{n-1} \subseteq \mathbb{R}^n$ denotes the unit sphere and $l(x) \in [0, +\infty]$ denotes the length of the segment contained in $C(p)$ of the ray cast from p passing through x i.e.,

$$l(x) = \sup \left\{ t \geq 0 \mid p + t\frac{x-p}{d(x,p)} \in C(p) \right\}. \tag{C.5}$$

We refer to Figure C.2 for a visual illustration. Note that $l(x)$ is defined for $x \neq p$ and is continuous in its domain since $l(x) = d(x,p) = d(x,q)$ for $x \in C(p) \cap C(q)$.

We aim to solve Equation C.4 by forcing the conical integral in Equation C.4 to be *constant*. To this end, we fix a continuous and strictly decreasing function

$K : \mathbb{R}_{>A} \to \mathbb{R}_{\geq 0}$ (a 'kernel') defined on a half-line $\mathbb{R}_{>A}$, $A < 0$, with the property that $t^{n-1}K(t)$ is integrable on $\mathbb{R}_{>0}$. By an ansatz we look for a density in the form

$$f_P(x) = \frac{K\left(\beta(l(x))d(x,p)\right)}{\alpha|P|\mathrm{Vol}(\mathbb{S}^{n-1})} \tag{C.6}$$

where $\alpha > 0$ is a hyperparameter and $\beta : \mathbb{R}_{>0} \to \mathbb{R}$ is a function that we would like to determine. The latter intuitively represents a *radial bandwidth*. The density f_P is continuous since the discontinuity of l at $x = p$ is amended by the vanishing of $d(x,p)$. Equation C.2 is satisfied if for every $l > 0$:

$$\int_0^l t^{n-1}K(\beta(l)t)\ \mathrm{d}t = \alpha. \tag{C.7}$$

Since K is strictly decreasing, the above expression always has a unique solution $\beta(l) > \frac{A}{l}$ assuming that $t^{n-1}K(t)$ is not integrable around A. Such a guaranteed solution can be computed via any root-finding algorithm and is continuous w.r.t. l. We provide an analysis of the function β and a discussion of the Newton-Raphson method for its computation in Section C.4.3.

The derivations above bring us to the following definition.

Definition C.4.1. Fix an $\alpha > 0$ and a continuous function $K : \mathbb{R}_{>A} \to \mathbb{R}_{>0}$ with the domain bound $A < 0$. Assume the following:

- $K(0) = 1$,
- K is strictly decreasing,
- $|t|^{n-1}K(t)$ is integrable around $+\infty$ but not integrable around A.

The *Radial Voronoi Density Estimator* (RVDE) is the density estimator defined by Equation C.6 where β is the function defined implicitly by Equation C.7.

The following two standard families of kernels K satisfy the above requirements:

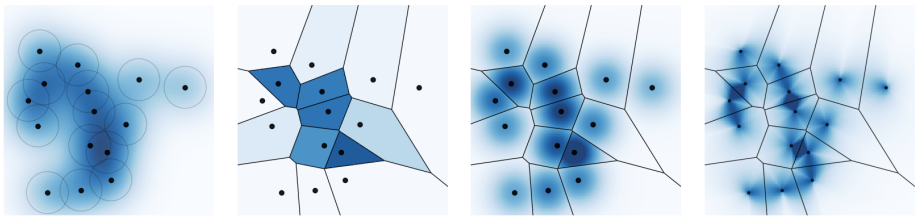
$$\begin{array}{cc} \textit{Exponential} & \textit{Rational} \\ K(t) = e^{-t} & K(t) = \frac{1}{(t+1)^k} \end{array} \tag{C.8}$$

where $k > n$. The domain bounds are $A = -\infty$ and $A = -1$ respectively. When $n = 1$ and K is the exponential kernel, the function β is closely related to the Lambert W function [30] via the expression:

$$\beta(l) = \frac{1}{\alpha} + W\left(-\frac{l}{\alpha}e^{-\frac{l}{\alpha}}\right). \tag{C.9}$$

We provide an empirical comparison between the two kernels from Equation C.8 in Section C.5.3.

The intuition behind the hyperparameter α is that it controls the trade-off between the amount of density concentrated around P and away from it (i.e., on the boundary of Voronoi cells). Indeed as $\alpha \to 0^+$ RVDE tends (in distribution) to the discrete empirical measure over P while as $\alpha \to +\infty$ it tends to a measure concentrated on the boundary of Voronoi cells. This can be deduced from Equation C.7 since $\beta(l)$ tends to $+\infty$ and to A/l respectively and thus Equation C.6 tends to 0 for $d(x,p) \neq 0, l(x)$. This intuition around α will be corroborated by Proposition C.4.3, where we study how it controls the distribution of modes of RVDE and consequently propose a heuristic selection procedure.



KDE [4] | Original VDE [7] | CVDE [9] | RVDE Ours

Figure C.3: From left to right: heatmaps of KDE, of the two VDEs from the literature and of our RVDE.

C.4.2 Computational Complexity and Sampling

We now discuss the computational cost of evaluating RVDE at a point $x \in \mathbb{R}^n$. To begin with, the closest $p \in P$ to x can be found in logarithmic time w.r.t. $|P|$ by organizing P in an efficient data structure for nearest neighbor lookups such as a k-d tree. Then $l(x)$ can be computed in linear time via the following closed expression ([9]):

$$l(x) = \min_{q \neq p,\ l^q(x) \geq 0} l^q(x) \tag{C.10}$$

where

$$l^q(x) = \frac{d(q,p)^2}{2\left\langle \frac{x-p}{d(x,p)}, q-p \right\rangle}. \tag{C.11}$$

The computational cost of evaluating $f_P(x)$ is thus linear w.r.t. $|P|$. The remaining compute essentially reduces to solving Equation C.7, which depends on the integrator, the root-finder algorithm adopted and the desired precision.

The formulation of RVDE enables a simple and efficient procedure for sampling from the estimated density. In order to sample, one first chooses a $p \in P$ uniformly since f_P integrates to $\frac{1}{|P|}$ on each Voronoi cell (Equation C.2). Since

$t^{n-1} f_P(p+t\sigma)$ integrates to a constant on the ray $r = \{p+t\sigma\}_{t\geq 0} \cap C(p)$ for every $\sigma \in \mathbb{S}^{n-1}$, one then samples σ uniformly from the sphere. Finally one samples t from the one-dimensional density $t^{n-1}K(t)$ restricted to the interval $[0, l(p+\sigma)]$. The computational complexity of the latter step depends on the kernel as well as of the sampling method. The result of the sampling is $p + t\sigma$. Because of the computational cost of $l(p+\sigma)$, the sampling complexity of RVDE is linear w.r.t. $|P|$.

RVDE is more efficient than the VDE discussed by [9] (see the end of Section C.3). The latter relies on Monte Carlo integration for numerical approximation of volumes of Voronoi cells and has complexity $O(\Sigma|P|^2)$ where Σ is the number of Monte Carlo samples. Compared to KDE, RVDE has the same computational complexity (for both evaluation and sampling) while retaining the geometric benefits of a VDE.

C.4.3 Study of β and Modes

In this section we discuss qualitative properties and computational aspects of the function β defined implicitly by Equation C.7 and consequently characterize the modes of RVDE. We start by presenting an explicit expression of the Newton-Raphson iteration for the computation of $\beta(l)$.

Proposition C.4.1. *Fix $l > 0$ and suppose $K \in C^1(\mathbb{R}_{>A})$ i.e., it is continuously differentiable. Then the iteration β_{m+1} of the Newton-Raphson method for computing $\beta(l)$ by solving Equation C.7 takes form:*

$$\beta_{m+1} = \beta_m + \frac{\beta_m}{n}\left(1 - \frac{l^n K(\beta_m l) - n\alpha}{l^n K(\beta_m l) - n\int_0^l t^{n-1} K(\beta_m t)\ \mathrm{d}t}\right). \tag{C.12}$$

Moreover, if K is convex then the Newton-Raphson method converges for any initial value β_0 i.e., $\lim_{m\to\infty} \beta_m = \beta(l)$.

We refer to the Appendix for a proof. Note that the convexity assumption is satisfied by both the kernels from Equation C.8. Proposition C.4.1 enables to compute $\beta(l)$ and together with Section C.4.2 provides all the algorithmic details for implementing RVDE.

Next, we outline a qualitative study of the function $l \mapsto \beta(l)$.

Proposition C.4.2. *The function $\beta : \mathbb{R}_{>0} \to \mathbb{R}$ is increasing, has a zero at $l = (n\alpha)^{\frac{1}{n}}$ and has an horizontal asymptote:*

$$\lim_{l\to+\infty} \beta(l) = \left(\frac{1}{\alpha}\int_0^\infty t^{n-1} K(t)\ \mathrm{d}t\right)^{\frac{1}{n}}. \tag{C.13}$$

Moreover if $K \in C^1(\mathbb{R}_{>A})$ *then* $\beta \in C^1(\mathbb{R}_{>0})$ *and it satisfies the differential equation:*

$$\left(l - \frac{n\alpha}{l^{n-1}K(\beta(l)l)}\right)\frac{\mathrm{d}\beta}{\mathrm{d}l}(l) = -\beta(l). \tag{C.14}$$

We refer to the Appendix for a proof. As discussed in Section C.4.1, β generalizes the Lambert W function. The properties and the differential equation from Proposition C.4.2 generalize their well-known instances for the W function [30].

We now focus on the study of modes. Our goal is to describe the modes of RVDE completely. This is an advantage over density estimators such as KDE, where the modes are challenging to describe and to compute approximately [31,32]. Denote by $\varepsilon = (n\alpha)^{\frac{1}{n}}$ the zero of β. Proposition C.4.2 implies that for $x \in \mathbb{R}^n$, the density f_P decreases radially w.r.t. p in the direction of x if $l(x) > \varepsilon$ and increases otherwise. This leads to the following result.

Proposition C.4.3. *The modes of* f_P *are classified as follows:*

(1) $p \in P$ *if* $d(p,q) > 2\varepsilon$ *for every Voronoi cell* $C(q)$ *adjacent to* $C(p)$*,*

(2) $\frac{p+q}{2}$ *for* $p, q \in P$ *if* $\frac{p+q}{2} \in C(p) \cap C(q)$ *and* $d(p,q) < 2\varepsilon$*,*

(3) all points belonging to the segment $[p,q]$ *for* $p, q \in P$ *if* $\frac{p+q}{2} \in C(p) \cap C(q)$ *and* $d(p,q) = 2\varepsilon$*.*

We refer to the Appendix for a proof and to Figure C.4 for an illustration. Since ε depends monotonically on the hyperparameter α, the latter controls the threshold for distances between adjacent points in P below which the mode gets pushed away from such points towards the boundary of the Voronoi cells. Intuitively, α determines the extent by which points in P are considered 'isolated' (i.e., a mode) or otherwise get 'merged' by placing a mode between them.

An alternative geometric formulation of Proposition C.4.3 is the following. Consider the *Gabriel graph* of P [10] containing an edge between p and q iff $\frac{p+q}{2} \in C(p) \cap C(q)$ and discard all the edges of length greater than 2ε. The modes of RVDE are then associated with (1) all isolated vertices, (2) midpoints of edges and (3) whole edges of length 2ε. Intuitively, the modes of RVDE are distributed geometrically according to the truncated Gabriel graph.

This suggests a possible heuristic procedure for *hyperparameter selection* of α. An option is to consider statistics of lengths of edges in the Gabriel graph and choose 2ε (and thus α) as a percentile. The percentile we suggest is $\frac{|P|-1}{|E|}$ where E denotes the set of edges of the Gabriel graph. The intuition is that we wish to avoid modes distributed in cycles. The number of cycles in the Gabriel graph is $|P| - |E| + 1$, from which our suggested percentile follows. This procedure enables to select α automatically and we evaluate it empirically in Section C.5. However, it comes with a number of limitations. First, the computational complexity of such

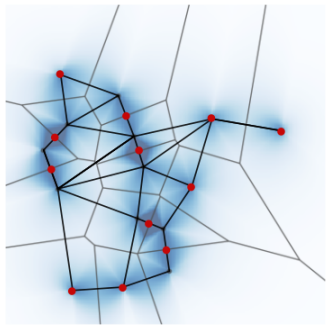


Figure C.4: Illustration of the modes of RVDE (red) together with the Gabriel graph (black).

a procedure is $O(|P|^3)$ because of the construction of the Gabriel graph, which is feasible but might become expensive for large datasets. Another limitation is its independence from the kernel K. The selection of α might be satisfying for some kernels but not for others. In our empirical evaluation from Section C.5 we show that for the rational kernel the selected α is close to the optimal one in practice, while for the exponential kernel the selection is further from optimality.

C.5 Experiments

Our empirical investigation is organized as follows. First we study RVDE on its own by comparing the different choices of the kernel. We then compare RVDE with other non-parametric density estimators on a variety of datasets.

C.5.1 Evaluation Metrics and Baselines

We evaluate all the density estimators f_P via average log-likelihood on a test set P_{test} i.e.,

$$\frac{1}{|P_{\text{test}}|} \sum_{x \in P_{\text{test}}} \log f_P(x). \tag{C.15}$$

This measures whether the estimator assigns high density values to points outside of P but sampled from the same distribution. In order to empirically evaluate the computational complexity, we additionally include runtimes for all the experiments. Our implementations of all the considered density estimators share the same programming framework and are parallelized to a similar degree, making the raw runtimes a fair comparison. We perform experiments on a machine with an AMD Ryzen 9 5950X 16-core CPU and a GeForce RTX 3090 GPU.

We deploy the following non-parametric density estimators as baselines in the experiments.

Kernel Density Estimator (KDE): given a (normalized) kernel $K : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ the density is estimated as:

$$f_P(x) = \frac{1}{|P|h^n} \sum_{p \in P} K\left(\frac{x-p}{h}\right) \tag{C.16}$$

where h is the bandwidth hyperparameter.

Adaptive Kernel Density Estimator (AdaKDE; [18]): a version of KDE where the bandwidth h_p depends on $p \in P$ and is smaller when data is denser around p. Specifically, if $f_P(p)$ denotes the standard KDE estimate with a global bandwidth h then $h_p = h\lambda_p$, where:

$$\lambda_p = (g/f_P(p))^{\frac{1}{2}}, \qquad g = \prod_{q \in P} f_P(q)^{\frac{1}{|P|}}. \tag{C.17}$$

Compactified Voronoi Density Estimator (CVDE; [9]): the VDE described at the end of Section C.3. It depends on a kernel K (together with a bandwidth) and is discontinuous on the boundary of Voronoi cells.

C.5.2 Datasets

In our experiments we consider data of varying nature. This includes both simple synthetic distributions and real-world datasets in high dimensions. For the latter, we consider sound data ($n = 21$) and image data ($n = 100$). Our datasets are the following.

Synthetic Datasets: datasets generated from a number of simple densities in $n = 10$ dimensions. Both P and P_{test} contain 1000 points in all the cases. The densities we consider are: a standard Gaussian distribution, a standard Laplace distribution, a Dirichlet distribution with parameters $\alpha_i = \frac{1}{n+1}$ and a mixture of two Gaussians with means $\mu_1 = (-0.5, 0, \cdots, 0)$, $\mu_2 = (0.5, 0, \cdots, 0)$ and standard deviations $\sigma_1 = 0.1$, $\sigma_2 = 10$ respectively.

MNIST [33]: a dataset consisting of 28×28 grayscale images of handwritten digits which are normalized in order to lie in $[0, 1]^{28 \times 28}$. In order to densify the data and obtain more meaningful estimates, we downscale the images to resolution 10×10. For each experimental run, we sample half of the 60000 training datapoints in order to evaluate the variance of the estimation. The test set size is 10000.

Anuran Calls [34]: a dataset consisting of 7195 calls from 10 species of frogs which are represented by 21 normalized mel-frequency cepstral coefficients in $[0, 1]^{21}$. We

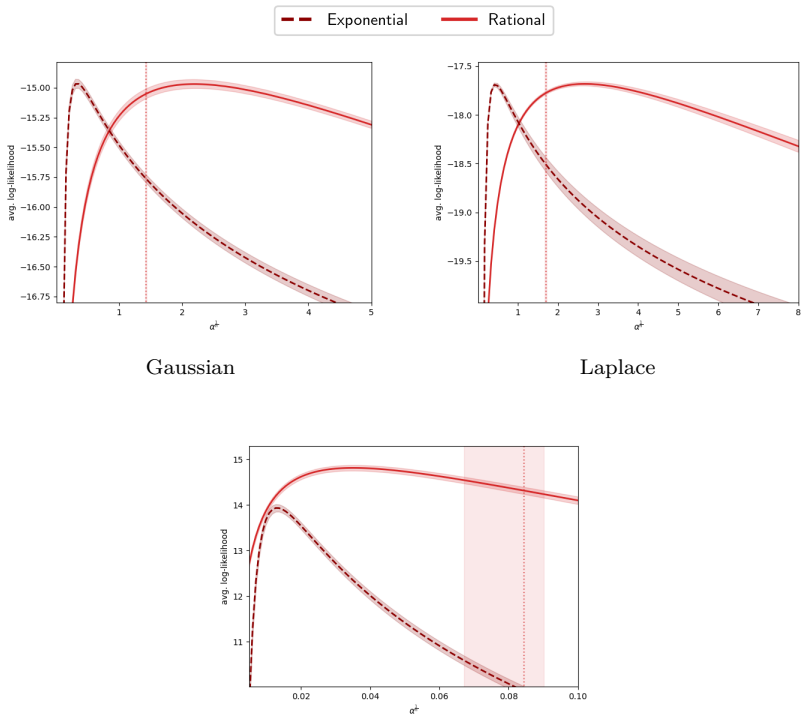


Figure C.5: Comparison of the two kernels for RVDE (Equation C.8) on three simple distributions in 10 dimensions.

retain 10% of data for testing and sample half of the training data at each experimental run.

C.5.3 Comparison of Kernels

Our first experiment consists of a comparison between the rational and the exponential kernel for RVDE (Equation C.8) on the synthetic datasets. In what follows the exponent k for the rational kernel is set to $k = n + 1$ for simplicity, where n is the dimension of the ambient space of the dataset considered (in this experiment, $n = 10$).

The results are presented in Figure C.5. The plot displays the test log-likelihood (Equation C.15) as the hyperparameter α varies. The latter is scaled as $\alpha^{\frac{1}{n}}$ in order to be consistent with the visualizations in the following section. The curves on the plot represent mean and standard deviation (shaded areas) over 5 experimental

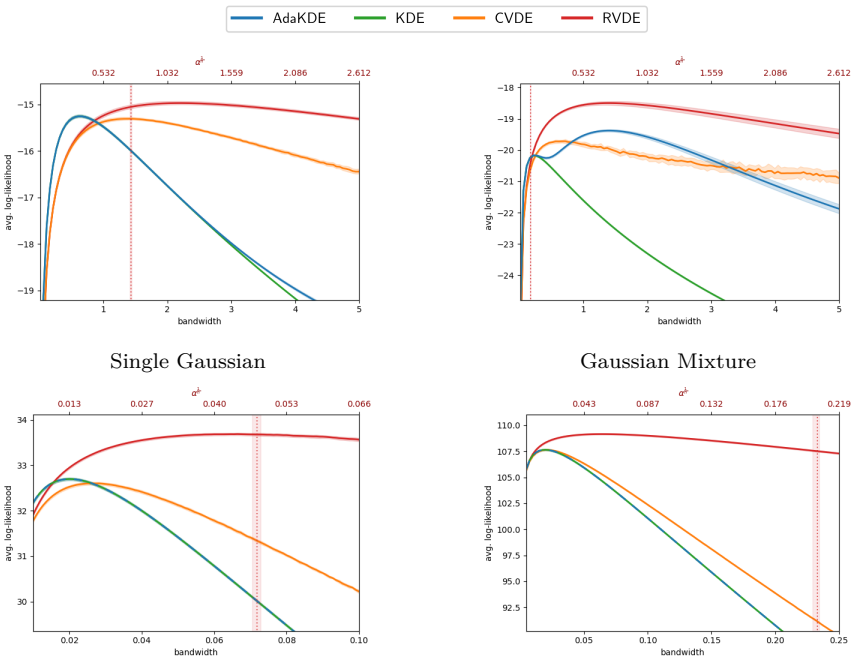


Figure C.6: Comparison of the estimators as the bandwidth varies. All the estimators implement the rational kernel.

runs for 100 bandwidths. The additional vertical lines correspond to the value of the hyperparameter selection heuristic discussed at the end of Section C.4.3. As can be seen, the performance of the rational kernel is more stable w.r.t. the hyperparameter α. The exponential kernel, however, achieves a slightly higher test score with its best hyperparameter on the Gaussian and Laplace datasets. Note that the heuristically chosen α aligns well with the one that achieves the best performance for the rational kernel, but is misaligned for the exponential one. We conclude that the rational kernel is generally a better option unless an extensive hyperparameter search is performed. In what follows we consequently stick to the rational kernel for RVDE.

C.5.4 Comparison with Baselines

In our main experiment we compare the performance of RVDE with the baselines described in Section C.5.1. We consider the test log-likelihood (Equation C.5.1), the standard deviation of the latter and the runtimes. In order to make the comparison as fair as possible, all the estimators are implemented with the rational kernel. We

found out that the performances drop with the more standard Gaussian kernel (which does not apply to RVDE). We include the results with both the Gaussian kernel and the exponential kernel in the Appendix.

Table C.1: Average runtimes (in seconds) per one full train-test run with fixed bandwidth. RVDE is highlighted in blue.

	RVDE	CVDE	KDE	AdaKDE
Gaussian	0.0376	0.265	0.0340	0.266
Anuran Calls	0.0581	0.490	0.0787	0.870
MNIST	17.4	408	12.5	75.0

The plot in Figure C.6 displays the (test) log-likelihood for all the estimators as the bandwidth hyperparameter h varies (see the definition of the baselines in Section C.5.1). In order to compare RVDE on the same scale as the other estimators, we convert h to α via:

$$\alpha = \int_0^\infty K\left(\frac{t}{h}\right)\,\mathrm{d}t = h^n \int_0^\infty K(t)\,\mathrm{d}t. \tag{C.18}$$

As can be seen, RVDE outperforms the baselines (each with its respective best bandwidth) in all the cases considered. The margin between RVDE and the baselines is especially evident on the more complex and high-dimensional datasets (Anuran Calls and MNIST). This confirms that the geometric benefits and the continuity properties of RVDE translate into better estimates for densities of different nature and increasing dimensionality.

Table C.1 reports the average runtime for an experimental run (with a single fixed bandwidth) for each estimator. RVDE outperforms the CVDE as well as AdaKDE by an extremely large margin. KDE achieves comparable runtimes to RVDE: it is slightly faster on Gaussian and MNIST while it is slightly slower on Anuran Calls. This confirms empirically the discussion from Section C.4.2: RVDE is significantly more efficient than CVDE and has the same (asymptotic) complexity as KDE.

Table C.2 separately reports the standard deviation of the log-likelihood (averaged over P_{test}) w.r.t. the dataset sampling. For each estimator, we consider its best bandwidth according to the results from Figure C.6. We first observe that

Table C.2: Standard deviations of the (test) log-likelihood over 5 experimental runs. RVDE is highlighted in blue. Each estimator is considered with its best bandwidth.

	RVDE	CVDE	KDE	AdaKDE
Gaussian	0.788	0.843	0.572	0.572
Anuran Calls	1.170	1.253	1.152	1.152
MNIST	5.507	5.767	5.735	5.735

RVDE achieves lower standard deviation than CVDE on all the datasets. This corroborates the hypothesis that the continuity of RVDE results in more stable estimates than those obtained by the highly-discontinuous CVDE. KDE and AdaKDE achieve the lowest standard deviations on the Gaussian and Anuran Calls datasets. This is likely due to the smoothness of such estimators and again confirms the benefit of regularity biases in terms of stability. However, on the most complex dataset considered (MNIST) RVDE outperforms the baselines. This suggests that for articulated densities the biases of geometric nature become more beneficial than generic biases such as smoothness.

C.6 Conclusions and Future Work

In this work we introduced a non-parametric density estimator (RVDE) benefiting from the geometric properties of Voronoi tessellations while being continuous and computationally efficient. We provided both theoretical and empirical investigations of RVDE.

An interesting line for future investigation is to explore the radial construction of RVDE on Riemannian manifolds beyond the Euclidean space. In this generality the rays correspond to geodesics defined via the exponential map of the given manifold. A variety of Riemannian manifolds arise in statistics and machine learning. For example, data on spheres are the object of study of directional statistics [35], hyperbolic spaces are routinely deployed to represent hierarchical data [36] and complex projective spaces correspond to Kendall shape spaces from computer vision [37]. Those areas of research can potentially benefit from the geometric characteristics and the computational efficiency of an extension of RVDE to Riemannian manifolds.

C.7 Acknowledgements

This work was supported by the Swedish Research Council, Knut and Alice Wallenberg Foundation and the European Research Council (ERC-BIRD-884807).

C.8 Appendix

C.8.1 Proofs of Results from Section C.4.3

Proposition C.8.1. *Fix $l > 0$ and suppose $K \in C^1(\mathbb{R}_{>A})$. Then the iteration β_{m+1} of the Newton-Raphson method for computing $\beta(l)$ by solving Equation C.7 takes form:*

$$\beta_{m+1} = \beta_m \left(1 + \frac{1}{n}\left(1 - \frac{l^n K(\beta_m l) - n\alpha}{l^n K(\beta_m l) - n\int_0^l t^{n-1} K(\beta_m t)\ \mathrm{d}t}\right)\right). \tag{C.19}$$

Moreover, if K is convex then the Newton-Raphson method converges for any initial value β_0 i.e., $\lim_{m\to\infty} \beta_m = \beta(l)$.

Proof. Consider

$$F(\beta) = \int_0^l t^{n-1} K(\beta t)\ \mathrm{d}t - \alpha. \tag{C.20}$$

The iteration of the Newton-Rhapson method for solving $F(\beta) = 0$ takes form:

$$\beta_{m+1} = \beta_m - \frac{F(\beta_m)}{\frac{\mathrm{d}F}{\mathrm{d}\beta}(\beta_m)}. \tag{C.21}$$

Via integration by parts we obtain:

$$\frac{\mathrm{d}F}{\mathrm{d}\beta}(\beta) = \int_0^l t^n \frac{\mathrm{d}K}{\mathrm{d}t}(\beta t)\ \mathrm{d}t = \frac{1}{\beta}\left(l^n K(\beta l) - n\int_0^l t^{n-1} K(\beta t)\ \mathrm{d}t\right). \tag{C.22}$$

Equation C.19 follows then from Equation C.21 by elementary algebraic manipulations. The convergence guarantee follows from the fact that if K is convex then F is easily seen to be convex as well. The Newton-Raphson method is well-known to be convergent for convex functions ([38]). □

Proposition C.8.2. *The function $\beta : \mathbb{R}_{>0} \to \mathbb{R}$ is increasing, has a zero at $l = (n\alpha)^{\frac{1}{n}}$ and has an horizontal asymptote:*

$$\lim_{l\to+\infty} \beta(l) = \left(\frac{1}{\alpha}\int_0^\infty t^{n-1} K(t)\ \mathrm{d}t\right)^{\frac{1}{n}}. \tag{C.23}$$

Moreover if $K \in C^1(\mathbb{R}_{>A})$ then $\beta \in C^1(\mathbb{R}_{>0})$ and it satisfies the differential equation:

$$\left(l - \frac{n\alpha}{l^{n-1} K(\beta(l) l)}\right)\frac{\mathrm{d}\beta}{\mathrm{d}l}(l) = -\beta(l). \tag{C.24}$$

Proof. The claim on the monotonicity of β follows directly from its definition (Equation C.7) and the hypothesis that K is decreasing. In order to compute its zero, note that $\beta(l) = 0$ implies $\alpha = \int_0^l K(0)t^{n-1} = \frac{l^n}{n}$ and thus $l = (n\alpha)^{\frac{1}{n}}$. For the asymptote note that for $l = +\infty$ Equation C.7 becomes by a change of variables:

$$\int_0^\infty t^{n-1}K(\beta(+\infty)t)\ \mathrm{d}t = \frac{1}{\beta(+\infty)^n}\int_0^\infty t^{n-1}K(t)\ \mathrm{d}t = \alpha. \tag{C.25}$$

Lastly, in order to obtain the differential equation for β we differentiate Equation C.7 on both sides and get:

$$\begin{aligned} 0 = \frac{\mathrm{d}}{\mathrm{d}l}\int_0^l t^{n-1}K(\beta(l)t)\ \mathrm{d}t &= l^{n-1}K(\beta(l)l) + \int_0^l t^{n-1}\frac{\mathrm{d}}{\mathrm{d}l}K(\beta(l)t)\ \mathrm{d}t \\ &= l^{n-1}K(\beta(l)l) + \frac{\mathrm{d}\beta}{\mathrm{d}l}(l)\int_0^l t^n\frac{\mathrm{d}K}{\mathrm{d}t}(\beta(l)t)\ \mathrm{d}t \\ &= l^{n-1}K(\beta(l)l) + \frac{\mathrm{d}\beta}{\mathrm{d}l}(l)\frac{l^nK(\beta(l)l) - n\alpha}{\beta(l)} \end{aligned} \tag{C.26}$$

where in the first identity we deployed the (distributional) Leibniz rule while in the last one we deployed integration by parts.

□

Proposition C.8.3. *The modes of f_P are as follows:*

(1) $p \in P$ if $d(p,q) > 2\varepsilon$ for every Voronoi cell $C(q)$ adjacent to $C(p)$,

(2) $\frac{p+q}{2}$ for $p, q \in P$ if $\frac{p+q}{2} \in C(p) \cap C(q)$ and $d(p,q) < 2\varepsilon$,

(3) all points belonging to the segment $[p,q]$ for $p, q \in P$ if $\frac{p+q}{2} \in C(p) \cap C(q)$ and $d(p,q) = 2\varepsilon$.

Proof. Pick $p \in P$. If p satisfies the hypothesis of the first claim then $l(x) > \varepsilon$ for every $x \in C(p)$ and thus $\beta(l(x)) > 0$ by Proposition C.8.2. Since K is decreasing, f_P decreases radially w.r.t. p in $C(p)$ and the first claim follows. If p does not satisfy the hypothesis of the first claim then $\beta(l(x)) \leq 0$ for some $x \in C(p)$. With the exception of the case $\beta(l) = 0$, the modes lie then on the boundary and are of the form $K(\beta(l)l)$ up to a multiplicative constant. The function $\beta(l)l$ is increasing in l since by appealing to Proposition C.8.2 we can compute its derivative:

$$\frac{\mathrm{d}\beta(l)l}{\mathrm{d}l} = \beta(l) + l\frac{\mathrm{d}\beta(l)}{\mathrm{d}l} = \beta(l)\frac{n\alpha}{n\alpha - l^nK(\beta(l)l)} \geq 0. \tag{C.27}$$

Since $l(x)$ has local minima at midpoints of segments connecting points in P, $K(\beta(l)l)$ is locally maximized therein and the second claim follows. In the hypothesis of the third claim β vanishes on the segment and the density is thus constant.

□

C.8.2 Additional Experiments

In this section we report additional experimental results complementing the ones in the main of the manuscript. For completeness, we evaluate the density estimators on different kernels. Figure C.7 displays comparative results for all the estimators with the exponential and the Gaussian kernel (note that the latter does not apply to RVDE). Moreover, we experiment with different dimensions and evaluation metrics other than average log-likelihood. This is possible only on a synthetic dataset where the dimension can vary and where the ground-truth density ρ is known. The latter is necessary for the metric considered. Figure C.8 displays a comparison on a high-dimensional Gaussian mixture ($n = 30$) as well as a comparison on the Gaussian mixture as in Section C.5 ($n = 10$) where the evaluation metric is the empirical *Hellinger distance* on the test set:

$$\frac{1}{2|P_{\text{test}}|} \sum_{x \in P_{\text{test}}} \left(f_P(x)^{\frac{1}{2}} - \rho(x)^{\frac{1}{2}} \right)^2 . \tag{C.28}$$

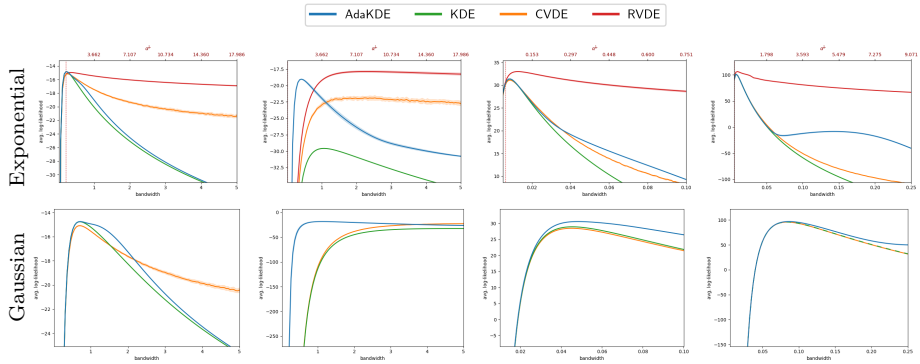


Figure C.7: Comparison of the estimators with the exponential and Gaussian kernel as the bandwidth varies.

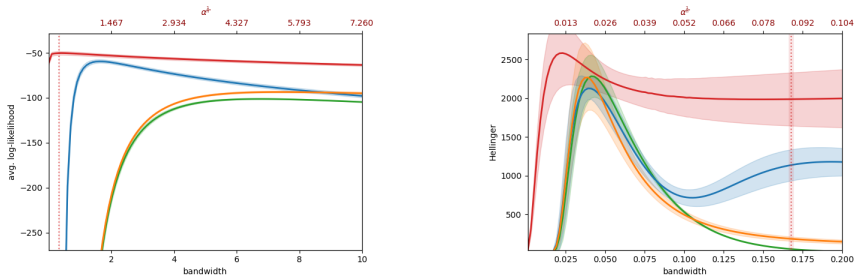


Figure C.8: Comparison of the estimators on a 30-dimensional Gaussian mixture (left) and on a 10-dimensional Gaussian mixture with the Hellinger distance as a metric (right).

References

[1] P. J. Diggle, *Statistical analysis of spatial and spatio-temporal point patterns*. CRC press, 2013.

[2] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[3] A. Gramacki, *Nonparametric kernel density estimation and its computational aspects*. Springer, 2018.

[4] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[5] D. Freedman and P. Diaconis, "On the histogram as a density estimator: L 2 theory," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 57, no. 4, pp. 453–476, 1981.

[6] K. Pearson, "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894.

[7] J. K. Ord, "How many trees in a forest," *Mathematical Scientist*, vol. 3, pp. 23–33, 1978.

[8] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*, vol. 501. John Wiley & Sons, 2009.

[9] V. Polianskii, G. L. Marchetti, A. Kravberg, A. Varava, F. T. Pokorny, and D. Kragic, "Voronoi density estimator for high-dimensional data: Computation, compactification and convergence," in *Uncertainty in Artificial Intelligence*, pp. 1644–1653, PMLR, 2022.

[10] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic zoology*, vol. 18, no. 3, pp. 259–278, 1969.

[11] D. W. Scott, "A note on choice of bivariate histogram bin shape," *Journal of Official Statistics*, vol. 4, no. 1, p. 47, 1988.

[12] D. B. Carr, A. R. Olsen, and D. White, "Hexagon mosaic maps for display of univariate and bivariate geographical data," *Cartography and Geographic Information Systems*, vol. 19, no. 4, pp. 228–236, 1992.

[13] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[14] A. J. Izenman, "Review papers: Recent developments in nonparametric density estimation," *Journal of the american statistical association*, vol. 86, no. 413, pp. 205–224, 1991.

[15] K. Dehnad, "Density estimation for statistics and data analysis," 1987.

[16] J. S. Marron, "A comparison of cross-validation techniques in density estimation," *The Annals of Statistics*, pp. 152–162, 1987.

[17] M. P. Wand, M. C. Jones, *et al.*, "Multivariate plug-in bandwidth selection," *Computational Statistics*, vol. 9, no. 2, pp. 97–116, 1994.

[18] B. Wang and X. Wang, "Bandwidth selection for weighted kernel density estimation," *arXiv preprint arXiv:0709.1616*, 2007.

[19] C. M. van der Walt and E. Barnard, "Variable kernel density estimation in high-dimensional feature spaces," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[20] Z. Xie and J. Yan, "Kernel density estimation of traffic accidents in a network space," *Computers, environment and urban systems*, vol. 32, no. 5, pp. 396–406, 2008.

[21] M. J. Baxter, C. C. Beardah, and R. V. S. Wright, "Some archaeological applications of kernel density estimates," *Journal of Archaeological Science*, vol. 24, no. 4, pp. 347–354, 1997.

[22] H. Bo, L. Yudun, Y. Hejun, and W. He, "Wind speed model based on kernel density estimation and its application in reliability assessment of generating systems," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 2, pp. 220–227, 2017.

[23] M. Vannucci, *Nonparametric density estimation using wavelets*. Institute of Statistics & Decision Sciences, Duke University, 1995.

[24] E. Masry, "Multivariate probability density estimation by wavelet methods: Strong consistency and rates for stationary time series," *Stochastic processes and their applications*, vol. 67, no. 2, pp. 177–193, 1997.

[25] G. G. Walter, "Estimation with wavelets and the curse of dimensionality," *Manuscript, Department of Mathematical Sciences, University of Wisconsin-Milwaukee*, 1995.

[26] C. Duyckaerts, G. Godefroy, and J.-J. Hauw, "Evaluation of neuronal numerical density by dirichlet tessellation," *Journal of neuroscience methods*, vol. 51, no. 1, pp. 47–69, 1994.

[27] H. Ebeling and G. Wiedenmann, "Detecting structure in two dimensions combining voronoi tessellation and percolation," *Physical Review E*, vol. 47, no. 1, p. 704, 1993.

[28] I. Vavilova, A. Elyiv, D. Dobrycheva, and O. Melnyk, "The voronoi tessellation method in astronomy," in *Intelligent Astrophysics*, pp. 57–79, Springer, 2021.

[29] L. P. Devroye and T. J. Wagner, "The l1 convergence of kernel density estimates," *The Annals of Statistics*, pp. 1136–1139, 1979.

[30] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the lambertw function," *Advances in Computational mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[31] J. C. H. Lee, J. Li, C. Musco, J. M. Phillips, and W. M. Tai, "Finding an approximate mode of a kernel density estimate," in *29th Annual European Symposium on Algorithms (ESA 2021)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[32] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[33] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[34] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[35] K. V. Mardia and P. E. Jupp, *Directional statistics*, vol. 494. John Wiley & Sons, 2009.

[36] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," *Advances in neural information processing systems*, vol. 30, 2017.

[37] C. P. Klingenberg, "Walking on kendall's shape space: understanding shape spaces and their coordinate systems," *Evolutionary Biology*, vol. 47, no. 4, pp. 334–352, 2020.

[38] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

Paper D

Equivariant Representation Learning via Class-Pose Decomposition

Giovanni Luca Marchetti*, Gustaf Tegnér*, Anastasiia Varava, Danica Kragic

Abstract

We introduce a general method for learning representations that are equivariant to symmetries of data. Our central idea is to decompose the latent space into an invariant factor and the symmetry group itself. The components semantically correspond to intrinsic data classes and poses respectively. The learner is trained on a loss encouraging equivariance based on supervision from relative symmetry information. The approach is motivated by theoretical results from group theory and guarantees representations that are lossless, interpretable and disentangled. We provide an empirical investigation via experiments involving datasets with a variety of symmetries. Results show that our representations capture the geometry of data and outperform other equivariant representation learning frameworks.

D.1 Introduction

For an intelligent agent aiming to understand the world and to operate therein, it is crucial to construct rich *representations* reflecting the intrinsic structures of the perceived data [1]. A variety of self-supervised approaches have been proposed to address the problem of representation learning such as (variational) auto-encoders [2,3] and contrastive learning methods [4, 5]. These approaches are applicable to a wide range of scenarios due to their generality but often fail to capture fundamental

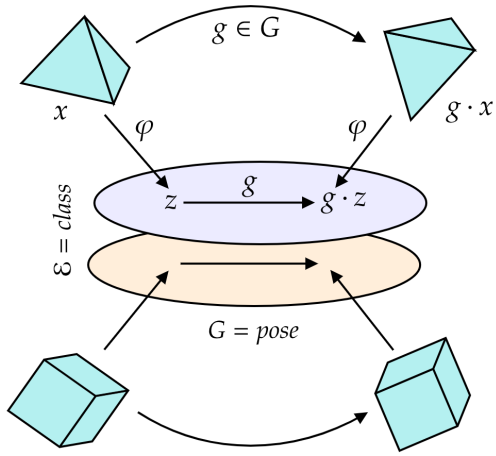


Figure D.1: An illustration of our equivariant representation decomposing intrinsic class and pose of data.

aspects hidden in data. For example, it has been recently shown that *disentangling* intrinsic factors of variation requires specific biases or some form of supervision [6]. This raises the need for representation learning paradigms leveraging upon specific structures carried by data.

A fundamental geometric structure of datasets consists of their *symmetries* [7]. Such structure arises in several practical scenarios. Consider for example images depicting rigid objects in different poses. In this case the symmetries are rigid transformations (translations and rotations) that *act* on datapoints by transforming the pose of the depicted object. Symmetries not only capture the geometry of the pose but additionally preserve the object's shape, partitioning the dataset into intrinsic *invariant* classes. The joint information of shape and pose describes the data completely and is recoverable from the symmetry structure alone. Another example of symmetries arises in the context of an agent exploring an environment. Actions performed by a mobile robot can be interpreted as changes of frame i.e., symmetries transforming the data the agent perceives (see Figure D.2). Assuming the agent is capable of odometry (measurement of its own movement), such symmetries are collectable and available for learning. All this motivates the design of representations that rely on symmetries and behave coherently with respect to them – a property known as *equivariance* [8, 9].

In this work we introduce a general framework for equivariant representation learning. Our central idea is to encode *class and pose separately* by decomposing the latent space into an invariant factor $\mathcal{E}$ and a symmetry component G (see Figure D.1). The pose component G extracts geometry from data while the class compo-

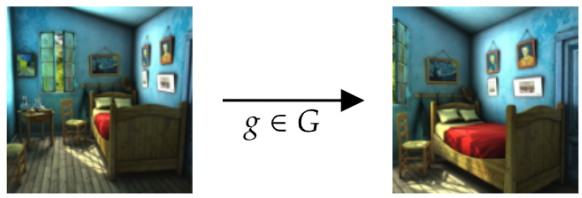


Figure D.2: Actions performed by a mobile agent can be seen as symmetries of the perceived data.

nent is interpretable and necessary for a lossless representation. We then train a representation learner $\varphi : \mathcal{X} \to \mathcal{E} \times G$ via a loss encouraging equivariance relying on supervision from relative symmetries between datapoints. Our methodology is based on a theoretical result guaranteeing that under mild assumptions an ideal learner achieves isomorphic representations by being trained on equivariance alone. Another advantage of our framework in the presence of multiple symmetry factors is that each of them can be varied independently by acting on the pose component. This realizes disentanglement in the sense of [9] which, as mentioned, would not be possible without the information carried by symmetries.

We rely on the abstract language of (Lie) *group theory* in order to formalize symmetries and equivariance. As a consequence, our framework is general and applicable to arbitrary groups of symmetries and to data of arbitrary nature. This is in contrast with previous works on equivariance often focusing on specific scenarios. For example, a number of works focus on Euclidean representations and linear or affine symmetry groups [10–12] while others enforce equivariance via group *convolutions* [13, 14]. The latter are only applicable when data consists of signals over a base space P (e.g., a pixel plane or a voxel grid) and symmetries are induced by the ones of P, which is limiting and does not lead to interpretable and structured representations. On the other hand, some recently introduced frameworks aim to jointly learn the equivariant representation together with the latent dynamics/symmetries [15, 16]. Although this has the advantage that the group of symmetries is not assumed to be known a priori, the obtained representation is again unstructured, uninterpretable, and comes with no theoretical guarantees.

We empirically investigate our framework on image datasets with a variety of symmetries including translations, dilations and rotations. We moreover provide both qualitative and quantitative comparisons with competing equivariant representation learning frameworks. Results show that our representations exhibit more structure and outperform the baselines in terms of latent symmetry prediction. Moreover, we show how the preservation of geometry of our framework can be applied to a *mapping* task: for data collected by a mobile agent our representation can be used to extract maps of multiple environments simultaneously. We provide

a Python implementation together with data and code for all the experiments at a publicly available repository [1]. In summary, our contributions include:

- A method for learning equivariant representations separating intrinsic data classes from poses.
- A general mathematical formalism based on group theory, which ideally guarantees lossless and disentangled representations.
- An empirical investigation via a set of experiments involving various group actions, together with applications to scene mapping through visual data.

D.2 The Mathematics of Symmetries

We now introduce the necessary mathematical background on symmetries and equivariance. The modern axiomatization of symmetries relies on their algebraic structure i.e., composition and inversion. The properties of those operations are captured by the abstract concept of a *group* [17].

Definition D.2.1. A group is a set G equipped with a *composition map* $G \times G \to G$ denoted by $(g, h) \mapsto gh$, an *inversion map* $G \to G$ denoted by $g \mapsto g^{-1}$, and a distinguished *identity element* $1 \in G$ such that for all $g, h, k \in G$:

Associativity	*Inversion*	*Identity*
$g(hk) = (gh)k$	$g^{-1}g = gg^{-1} = 1$	$g1 = 1g = g$

Examples of groups include the permutations of a set and the general linear group $\mathrm{GL}(n)$ of $n \times n$ invertible real matrices, both equipped with usual composition and inversion of functions. An interesting subgroup of the latter is the special orthogonal group $\mathrm{SO}(n) = \{A \in \mathrm{GL}(n) \mid AA^T = 1,\ \det(A) = 1\}$, which consists of linear orientation-preserving isometries of the Euclidean space. An example of a commutative group (i.e., such that $gh = hg$ for all $g, h \in G$) is $\mathbb{R}^n$ equipped with vector sum as composition.

The idea of a space $\mathcal{X}$ having G as a group of symmetries is abstracted by the notion of group *action*.

Definition D.2.2. An action by a group G on a set $\mathcal{X}$ is a map $G \times \mathcal{X} \to \mathcal{X}$ denoted by $(g, x) \mapsto g \cdot x$, satisfying for all $g, h \in G,\ x \in \mathcal{X}$:

Associativity	*Identity*
$g \cdot (h \cdot x) = (gh) \cdot x$	$1 \cdot x = x$

[1] `https://github.com/equivariant-ml/equivariant-representation-learning`

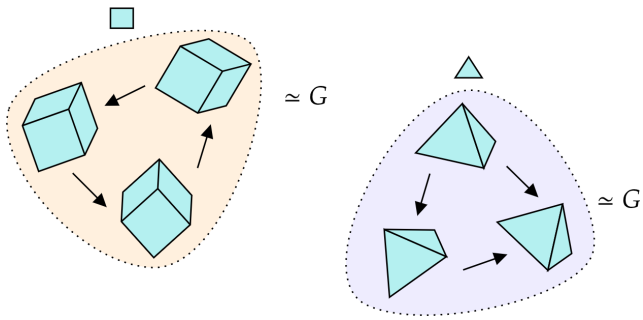


Figure D.3: Orbits of a (free) group action represent intrinsic classes of data. Each orbit is isomorphic to the symmetry group G itself.

In general, the following actions can be defined for arbitrary groups: G acts on any set *trivially* by $g \cdot x = x$, and G acts on itself seen as a set via (left) *multiplication* by $g \cdot h = gh$. A further example of group action is $\mathrm{GL}(n)$ acting on $\mathbb{R}^n$ by matrix multiplication.

Maps which preserve symmetries are called *equivariant* and will constitute the fundamental notion of our representation learning framework.

Definition D.2.3. A map $\varphi\colon \mathcal{X} \to \mathcal{Z}$ between sets acted upon by G is called *equivariant* if $\varphi(g \cdot x) = g \cdot \varphi(x)$ for all $g \in G, x \in \mathcal{X}$. It is called *invariant* if moreover G acts trivially on $\mathcal{Z}$ or, explicitly, if $\varphi(g \cdot x) = \varphi(x)$. It is called *isomorphism* if it is bijective.

Now, group actions induce classes in $\mathcal{X}$ called *orbits* by identifying points related by a symmetry.

Definition D.2.4. Consider the equivalence relation on $\mathcal{X}$ given by deeming x and y equivalent if $y = g \cdot x$ for some $g \in G$. The induced equivalence classes are called *orbits*, and the set of orbits is denoted by $\mathcal{X}/G$.

For example the orbits of the trivial action are singletons, while the multiplication action has a single orbit. Data-theoretically, an orbit may be seen as an invariant, maximal class of data induced by the symmetry structure. In the example of rigid objects acted upon by translations and rotations, orbits indeed correspond to shapes (see Figure D.3).

It is intuitive to assume that a nontrivial symmetry $g \neq 1 \in G$ has to produce a change in data. If no difference is perceived, one might indeed consider the given transformation as trivial. We can thus assume that no point in $\mathcal{X}$ is fixed by an element of G different from the identity or, in other words, $g \cdot x \neq x$ for $g \neq 1$. Such actions are deemed as *free* and will be the ones relevant to the present work.

Assumption D.2.1. *The action by G on $\mathcal{X}$ is free.*

The following is the core theoretical result motivating our representation learning framework, which we will discuss in the following section. The result guarantees a general decomposition into a trivial and a multiplicative action and describes all the equivariant isomorphisms of such a decomposition.

Proposition D.2.2. *The following holds:*

- *There is an equivariant isomorphism*

$$\mathcal{X} \simeq (\mathcal{X}/G) \times G \tag{D.1}$$

  where G acts trivially on the orbits and via multiplication on itself, i.e., $g \cdot (e, h) = (e, gh)$ for $g, h \in G$, $e \in \mathcal{X}/G$. In other words, each orbit can be identified equivariantly with the group itself.

- *Any equivariant map $\varphi : (\mathcal{X}/G) \times G \to (\mathcal{X}/G) \times G$ is a right multiplication on each orbit i.e., for each orbit $O \in \mathcal{X}/G$ there is an $h_O \in G$ such that $\varphi(O, g) = (O', gh_O)$ for all $g \in G$. In particular, if φ induces a bijection on orbits then it is an isomorphism.*

We refer to the Appendix for a proof. The first part of the statement can be interpreted in plain words as a *decomposition of classes from poses* for any free group action. According to this terminology, a pose is abstractly an element of an arbitrary group G while a class is an orbit. The intuition behind the second part of the statement is that any equivariant map performs an orbit-dependent 'change of frame' in the sense that elements of an orbit O get composed on the right by a symmetry h_O depending on O. This will imply that our representations can differ from ground-truth ones only by such change of frames and will in fact guarantee isomorphic representations for our framework.

D.3 Method

D.3.1 General Equivariant Representation Learning

In the context of *representation learning* the goal of the model is to learn a map deemed 'representation' $\varphi\colon\ \mathcal{X} \to \mathcal{Z}$ from the data space $\mathcal{X}$ to a *latent space*. The learner optimizes a loss $\mathcal{L}\colon\ \mathcal{M} \to \mathbb{R}$ over parameters $\theta \in \mathcal{M}$ of the map $\varphi = \varphi_\theta$. The so-obtained representation can be deployed in downstream applications for an improved performance with respect to operating in the original data space.

The central assumption of *equivariant* representation learning is that data carries symmetries which the representation has to preserve. As discussed in Section D.2, this means that a group of symmetries G acts on both $\mathcal{X}$ and $\mathcal{Z}$ and that the

representation φ is encouraged to be equivariant via the loss. While the action on $\mathcal{Z}$ is designed as part of the model, the action on $\mathcal{X}$ is unknown in general and has to be conveyed by data. Concretely, the dataset consists of triples (x, g, y) with $x \in \mathcal{X}, g \in G$ and $y = g \cdot x$. The group element g carries symmetry information which is relative between x and y. Equivariance is then naturally encouraged via a loss in the form:

$$\mathcal{L}(\theta; x, g, y) = d\left(\varphi_\theta(y),\ g \cdot \varphi_\theta(x)\right). \tag{D.2}$$

Here $d\colon\ \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_{\geq 0}$ is a similarity function on $\mathcal{Z}$. Generally speaking, d does not necessarily need to satisfy the axioms for a distance but we at least require it to be *positive definite* i.e., $d(z, z') = 0$ iff $z = z'$. Note that we assume the group together with its algebraic structure to be known a priori and not inferred during the learning process. Its action over the latent space is defined in advance and constitutes the primary inductive bias for equivariant representation learning.

D.3.2 Learning to Decompose Class and Pose

Motivated by Proposition D.2.2, we propose to set the latent space as:

$$\mathcal{Z} = \underbrace{\mathcal{E}}_{Class} \times \underbrace{G}_{Pose} \tag{D.3}$$

with G acting trivially on $\mathcal{E}$ and via multiplication on itself. Here, $\mathcal{E}$ is any set which is meant to represent classes of the encoded data. Since there is in general no prior information about the action by symmetries on $\mathcal{X}$ and its orbits, $\mathcal{E}$ has to be set beforehand. Assuming $\mathcal{E}$ has enough capacity to contain $\mathcal{X}/G$, Proposition D.2.2 shows that an isomorphic equivariant data representation is possible in $\mathcal{Z}$. By fixing (positive definite) similarity functions $d_\mathcal{E}$ and d_G on $\mathcal{E}$ and G respectively, we obtain a joint latent similarity function $d(z, z') = d_\mathcal{E}(z_\mathcal{E}, z'_\mathcal{E}) + d_G(z_G, z'_G)$, where the subscripts denote the corresponding components. When $G \subseteq \mathrm{GL}(n)$ is a group of matrices, a typical choice for d_G is the (squared) Frobenius distance i.e., the Euclidean distance for matrices seen as flattened vectors. The equivariance loss $\mathcal{L}(\theta; x, g, y)$ in Equation D.2 then reads:

$$\underbrace{d_\mathcal{E}\left(\varphi^\mathcal{E}(y),\ \varphi^\mathcal{E}(x)\right)}_{Invariant} + \underbrace{d_G\left(\varphi^G(y),\ g\varphi^G(x)\right)}_{Multiplication-Equivariant}. \tag{D.4}$$

Here we denoted the components of the representation map by $\varphi = (\varphi^\mathcal{E}, \varphi^G)$ and omitted the parameter θ for simplicity. To spell things out, $\varphi^\mathcal{E}$ encourages data from the same orbit to lie close in $\mathcal{E}$ (i.e., $\varphi^\mathcal{E}$ is ideally invariant) while φ^G aims for equivariance with respect to multiplication on the pose component G.

If $\varphi_\mathcal{E}$ is injective then Proposition D.2.2 guarantees lossless (i.e., isomorphic) representations, which we summarize in the following corollary:

Corollary D.3.1. *Suppose that $\varphi_{\mathcal{E}}$ is injective. Then $\mathcal{L}(\theta; x, g, y) = 0$ for all $x, g, y = g \cdot x$ if and only if φ_θ is an equivariant isomorphism on its image.*

In order to force injectivity, we propose a typical solution from contrastive learning literature [4] encouraging latent points to spread apart. To this end, we opt for the standard *InfoNCE loss* ([18]), although other choices are possible. This means that we replace the term $d_{\mathcal{E}}\left(\varphi^{\mathcal{E}}(y),\ \varphi^{\mathcal{E}}(x)\right)$ in Equation D.4 with

$$\frac{1}{\tau} d_{\mathcal{E}}\left(\varphi^{\mathcal{E}}(y),\ \varphi^{\mathcal{E}}(x)\right) + \log \mathbb{E}_{x'}\left[e^{-\frac{1}{\tau} d_{\mathcal{E}}\left(\varphi^{\mathcal{E}}(x'),\ \varphi^{\mathcal{E}}(x)\right)}\right]. \tag{D.5}$$

The hyperparameter $\tau \in \mathbb{R}_{>0}$ ('temperature') controls the amount of latent entropy. Following [4,18], we set the class component as a sphere $\mathcal{E} = \mathbb{S}^m$ by normalizing the output of $\varphi_{\mathcal{E}}$. This allows to deploy the cosine dissimilarity $d_{\mathcal{E}}(z, z') = -\cos(\angle z z') = -z \cdot z'$ and is known to lead to improved performances due to the compactness of $\mathcal{E}$ [19].

D.3.3 Parametrizing via the Exponential Map

The output space of usual machine learning models such as deep neural networks is Euclidean. Our latent space (Equation D.3) contains G as a factor, which might be non-Euclidean as in the case of $G = \mathrm{SO}(n)$. In order to implement our representation learner φ it is thus necessary to parametrize the group G. To this end, we assume that G is a differentiable manifold (with differentiable composition and inversion maps) i.e., that G is a *Lie group*. One can then define the *Lie algebra* $\mathfrak{g}$ of G as the tangent space to G at 1.

We propose to rely on the *exponential map* $\mathfrak{g} \to G$, denoted by $v \mapsto e^v$, to parametrize G. This means that φ outputs an element v of $\mathfrak{g}$ that gets mapped into G as e^v. Although the exponential map can be defined for general Lie groups by solving an appropriate ordinary differential equation, we focus on the case $G \subseteq \mathrm{GL}(n)$. The Lie algebra $\mathfrak{g}$ is then contained in the space of $n \times n$ matrices and the exponential map amounts to the matrix Taylor expansion $e^v = \sum_{k \geq 0} v^k / k!$. For specific groups the latter can be simplified via simple closed formulas. For example, the exponential map of $\mathbb{R}^n$ is the identity while for $\mathrm{SO}(3)$ it can be efficiently computed via the Rodrigues' formula [20].

D.3.4 Relation to Disentanglement

Our equivariant representation learning framework is related to the popular notion of *disentanglement* [1,21]. Intuitively, in a disentangled representation a variation of a distinguished aspect in the data is reflected by a change of a single component in the latent space. Although there is no common agreement on a rigorous formulation of the notion [6], a proposal has been addressed in [9]. The presence of multiple

dynamic aspects in the data is formalized as an action on $\mathcal{X}$ by a decomposed group

$$G = G_1 \times \cdots \times G_n \tag{D.6}$$

where each of the factors G_i is responsible for the variation of a single aspect. A representation $\varphi\colon\ \mathcal{X} \to \mathcal{Z}$ is then defined to be disentangled if (i) there is a decomposition $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_n$ where each $\mathcal{Z}_i$ is acted upon trivially by the factors G_j with $j \neq i$ and (ii) φ is equivariant.

Our latent space (Equation D.3) automatically yields to disentanglement in this sense. Indeed, in the case of a group as in Equation D.6 we set $\mathcal{Z}_i = G_i$. In order to deal with the remaining factor $\mathcal{Z}_0 = \mathcal{E}$, a copy of the trivial group $G_0 = \{1\}$ can be added to G without altering it up to isomorphism. The group $G \simeq G_0 \times \cdots \times G_n$ acts on $\mathcal{Z} = \mathcal{E} \times G = \mathcal{Z}_0 \times \cdots \times \mathcal{Z}_n$ as required for a disentangled latent space. In conclusion, our work fits in the line of research aiming to infer disentangled representation via indirect and weak forms of supervision [22], of which symmetry structures are an example.

D.4 Related Work

Equivariant Representation Learning. Models relying on symmetry and equivariance have been studied in the context of representation learning. These models are typically trained on variations of the equivariance loss (Equation D.2) and are designed for specific groups G and actions on the latent space $\mathcal{Z}$. The pioneering *Transforming Autoencoders* [23] learn to represent image data translated by $G = \mathbb{R}^2$ in the pixel plane, with $\mathcal{Z}$ consisting of several copies of G ('capsules') acting on itself. Although such models are capable of learning isomorphic representations, the orbits are not explicitly modeled in the latent space. In contrast, our invariant component $\mathcal{E}$ is an interpretable alternative to multiple capsules making orbits recoverable from the representation. A series of other works represent data in a latent space modelled via the group of symmetries G [24–26]. These works however either do not reserve additional components dedicated to orbits, obtaining a representation that forgets the intrinsic classes of data, or address specific groups i.e., $\mathrm{SO}(3)$, the torus $\mathrm{SO}(2) \times \mathrm{SO}(2)$ and product of cyclic groups respectively. *Affine Equivariant Autoencoders* [10] deal with affine transformations of the pixel-plane (shearing an image, for example) and implement a latent action through a hand-crafted map $t\colon\ G \to \mathcal{Z} = \mathbb{R}^n$. Groups of rotations $\mathrm{SO}(n)$ linearly acting on a Euclidean latent space $\mathcal{Z} = \mathbb{R}^n$ are explored in [11, 12]. Since rotating a vector around itself has no effect, linear actions are not free (for $n \geq 3$), which makes isomorphic representations impossible. *Equivariant Neural Rendering* [27] proposes a latent voxel grid on which $\mathrm{SO}(3)$ acts approximately by rotating and interpolating values. In contrast, our latent group action is exact and thus induces no loss of information. We provide an empirical comparison to both linear Euclidean actions and Equivariant Neural Rendering in Section D.5. Lastly, [28] have recently proposed to learn equivariant

representations by splitting the latent space into an invariant component and an equivariant one, which bears similarity to our framework. Differently from us, however, the model is trained via an equivariance loss in the data space $\mathcal{X}$ and thus requires the group actions over data to be known a priori. As previously discussed, this is limiting and often unrealistic in practice.

Convolutional Networks. Convolutional layers in neural networks [13, 14] satisfy equivariance a priori with respect to transformations of the pixel plane. They were originally introduced for (discretized) translations and later extended to more general groups [29–31]. However, they require data and group actions in a specific form. Abstractly speaking, data need to consist of vector fields over a base space (images seen as RGB fields over the pixel plane, for example) acted upon by G, which does not hold in general. Examples of symmetries not in this form are changes in perspective of first-person images of a scene and rotations of rigid objects on an image. Our model is instead applicable to arbitrary (Lie) group actions and infers equivariance in a data-driven manner. Moreover, equivariance through G-convolutions alone is hardly suitable for representation learning as the output dimension coincides with the input one. Dimensionality reduction techniques deployed together with convolutions such as max-pooling or fully-connected layers disrupt equivariance completely. The latent space in our framework is instead compressed and is ideally isomorphic to the data space $\mathcal{X}$ (Proposition D.2.2).

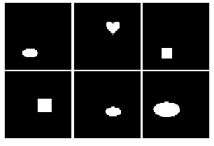
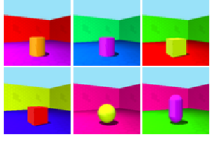
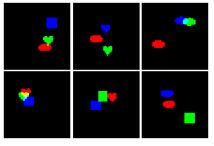
World Models. Analogously to group actions, Markov Decision Processes (MDPs) from reinforcement learning and control theory involve a possibly stochastic interaction $\mathcal{A} \times \mathcal{X} \to \mathcal{X}$ with an environment $\mathcal{X}$ via a set $\mathcal{A}$ of moves. In general, no algebraic structure (such as a group composition) is assumed on $\mathcal{A}$. In this context, a representation equivariant with respect to the action is referred to as *World Model* [15,32,33] or *Markov Decision Process Homomorphism* (MDPH) [16]. MDPHs are usually deployed as pre-trained representations for downstream tasks or trained jointly with the agent for exploration purposes [34]. However, the latent action $\mathcal{A} \times \mathcal{Z} \to \mathcal{Z}$ of an MDPH is learned since no prior knowledge is assumed around $\mathcal{A}$ or the environment. This implies that the resulting representation is unstructured and uninterpretable. We instead assume that $G = \mathcal{A}$ is a group acting (freely) on $\mathcal{X}$, which enables us to define a geometrically-structured and disentangled latent space that guarantees isomorphic equivariant representations. We provide an empirical comparison to MDPHs in Section D.5.

D.5 Experiments

D.5.1 Dataset Description

Our empirical investigation aims to assess our framework via both qualitative and quantitative analysis on datasets with a variety of symmetries. To this end we

Table D.1: Datasets involved in our experiments, with the corresponding group of symmetries and number of orbits.

	Sprites	Shapes	Multi-Sprites	Chairs	Apartments
$\mathcal{X}$					
G	$\mathbb{R}^3$	$\mathbb{R}^3$	$\mathbb{R}^6$	$\mathrm{SO}(3)$	$\mathbb{R}^2 \times \mathrm{SO}(2)$
$\lvert\mathcal{X}/G\rvert$	3	4	27	3	2

deploy five datasets summarized in Table D.1: three with translational symmetry extracted from dSprites and 3DShapes [35, 36], one with rotational symmetry extracted from ShapeNet [37] and one simulating a mobile agent exploring apartments and collecting first-person views. The latter is extracted from Gibson [38] and generated via the Habitat simulator [39]. Datapoints are triples (x, g, y) where x, y are 64×64 images, $g \in G$ and $y = g \cdot x$. We refer to the Appendix for a more detailed description of the datasets.

D.5.2 Baselines and Implementation Details

We compare our method with the following models designed for learning equivariant representations:

MDP Homomorphisms (MDPH) from [15, 16]: a framework where the representation $\varphi : \mathcal{X} \to \mathcal{Z}$ is learnt jointly with the latent action $T : G \times \mathcal{Z} \to \mathcal{Z}$. The two models are trained with the equivariance loss $\mathbb{E}_{x,g,y=g\cdot x}[d(\varphi(y),\ T(g, \varphi(x)))]$ (cf. Equation D.2). In order to avoid trivial solutions, an additional 'hinge' loss term $\mathbb{E}_{x,x'}[\max\{0,\ 1 - d(x, x')\}]$ is optimized that encourages encodings to spread apart in the latent space. This is analogous to (the denominator of) the InfoNCE loss (Equation D.5) which we rely upon to avoid orbit collapse in $\mathcal{E}$. Differently from us, an MDPH does not assume any prior knowledge on G nor any algebraic structure on the latter. However, this comes at the cost of training an additional model T and losing the structures and guarantees provided by our framework.

Linear: a model with $\mathcal{Z} = \mathbb{R}^3$ on which SO(3) acts by matrix multiplication. Such a latent space has been employed in previous works [11, 12]. The model is trained with the same loss as MDPH i.e., equivariance loss together with the additional hinge term avoiding collapses such as $\varphi = 0$. Note that the action on $\mathcal{Z}$ is no longer free (even away from 0) since rotating a vector around itself has no effect. Differently from our method, the model is thus forced to lose information in order to learn an equivariant representation.

Equivariant Neural Renderer (ENR) from [27]: a model with a tensorial latent space $\mathcal{Z} = \mathbb{R}^{C\times D\times H\times W}$, thought as a copy of $\mathbb{R}^C$ for each point in a $D \times H \times W$ grid in $\mathbb{R}^3$. The group SO(3) *approximately* acts on $\mathcal{Z}$ by rotating the grid and interpolating the obtained values in $\mathbb{R}^C$. The model is trained trained jointly with a decoder $\psi : \mathcal{Z} \to \mathcal{X}$ and optimizes variation of the equivariance loss incorporating reconstruction: $\mathbb{E}_{x,g,y=g\cdot x}[d_{\mathcal{X}}(y,\ \psi(g \cdot \varphi(x)))]$. We set $d_{\mathcal{X}}$ as the standard binary cross-entropy metric for (normalized) images. Although the action on $\mathcal{Z}$ is free, the latent discretisation and consequent interpolation make the model only approximately equivariant.

We implement the equivariant representation learner φ as a ResNet-18 [40], which is a deep convolutional neural network with residual connections. We train

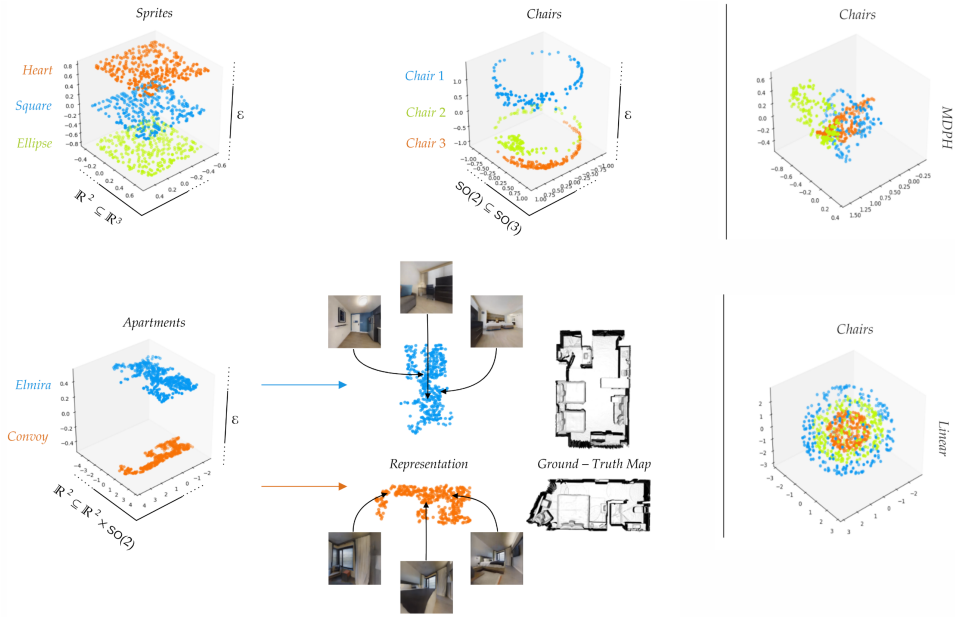


Figure D.4: Left: visualization of encodings through φ from the Sprites, Chairs and Apartments datasets. The images display the projection to the annotated components of $\mathcal{Z}$ and data are colored by their ground-truth class. Each latent orbit from Apartments is compared to the view from the top of the corresponding scene. **Right:** same visualization for the baseline models MDPH and Linear on the Chairs dataset.

our models for 100 epochs through stochastic gradient descent by means of the Adam optimizer with learning rate 10^{-3} and batch size 16. The distance d_G is set as the squared Euclidean one for $G = \mathbb{R}^n$ and for $G = \mathrm{SO}(2) \subseteq \mathbb{R}^2$, while it is set to the squared Frobenius one for $G = \mathrm{SO}(3)$. The invariant component consists of a sphere $\mathcal{E} = \mathbb{S}^7 \subseteq \mathbb{R}^8$ (see Section D.3) parametrized by the normalized output of 8 neurons in the last layer of φ. All the models from Section D.5.2 implement the same architecture (ResNet-18). ENR moreover implements 3D convolutional layers around the latent space as suggested in the original work [27]. The latent action model T for MDPH is implemented as a two-layer deep neural network (128 neurons per layer) with ReLU activation functions. For MDPH we set $\dim(\mathcal{Z}) = 8 + \dim(G)$, which coincides with the output dimensionality of our model.

D.5.3 Visualizations of the Representation

In this section we present visualizations of the latent space of our model (Equation D.3), showcasing its geometric benefits. The preservation of symmetries coming from equivariance enables indeed to transfer the intrinsic geometry of data explic-

itly to the representation. Moreover, the invariant component $\mathcal{E}$ separates the orbits of the group action, allowing to distinguish the intrinsic classes of data in the latent space. Finally, the representation from our model automatically disentangles factors of the group as discussed in Section D.3.4.

Figure D.4 (left) presents visualizations of encodings through φ for the datasets Sprites, Chairs and Apartments. For each dataset we display the projection to one component of $\mathcal{E}$ as well as a relevant component of the group G. Specifically, for Sprites we display the component $\mathbb{R}^2 \subseteq G = \mathbb{R}^3$ corresponding to translations in the pixel plane, for Chairs we display a circle $\mathrm{SO}(2) \subseteq G = \mathrm{SO}(3)$ corresponding to one Euler angle while for Apartments we display the component $\mathbb{R}^2 \subseteq G = \mathbb{R}^2 \times \mathrm{SO}(2)$ corresponding to translations in the physical world. For Apartments, we additionally compare representation of each of the two apartments with the ground-truth view from the top.

As can be seen, in all cases the model correctly separates the orbits in $\mathcal{E}$ through self-supervision alone. Since the orbits are isomorphic to the group G itself, the model moreover preserves the geometry of each orbit separately. For Sprites, this means that (the displayed component of) each orbit is an isometric copy of the pixel-plane, with disentangled horizontal and vertical translations. (Figure D.4, top-left). For Apartments, this similarly means that each orbit exhibits an isometric copy of the real-world scene. One can recover a map of each of the explored scenes by, for example, estimating the density of data in $\mathcal{Z}$ (Figure D.4, bottom-right) and further use the model φ to localize the agent within such map. Our equivariant representation thus solves a *localization and mapping task* in a self-supervised manner and of multiple scenes simultaneously.

As a qualitative comparison, Figure D.4 (right) includes visualizations for the models MDPH (trained with $\dim(\mathcal{Z}) = 3$) and Linear on the Chairs dataset. As can be seen, the latent space of MDPH is unstructured: the geometry of $\mathcal{X}$ is not preserved and classes are not separated. This is because the latent action of MDPH is learned end-to-end and is thus uninterpretable and unconstrained a priori. For Linear the classes are organized as spheres in $\mathcal{Z}$, which are are the orbits of the latent action by $G = \mathrm{SO}(n)$. Such orbits are not isomorphic to G (one Euler angle is missing) since the action is not free. This means that $\mathcal{Z}$ loses information and does not represent the dataset faithfully.

D.5.4 Performance Comparison

In this section we numerically compare our method to the equivariant representation learning frameworks described at the beginning of Section D.5. We evaluate the models through *hit-rate*, which is a standard score that allows to compare equivariant representations with different latent space geometries [15]. Given a test

Table D.2: Hit-rate (mean and std over 3 runs) on test trajectories of increasing length.

Dataset	Model	1 Step	10 Steps	20 Steps
Sprites	Ours	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
	MDPH	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.98_{\pm 0.02}$
Shapes	Ours	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
	MDPH	$1.00_{\pm 0.00}$	$0.99_{\pm 0.01}$	$0.96_{\pm 0.04}$
Multi-Sprites	Ours	$1.00_{\pm 0.00}$	$0.93_{\pm 0.03}$	$0.93_{\pm 0.03}$
	MDPH	$1.00_{\pm 0.00}$	$0.28_{\pm 0.06}$	$0.11_{\pm 0.01}$
Chairs	Ours	$0.98_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.94_{\pm 0.01}$
	Linear	$0.89_{\pm 0.08}$	$0.87_{\pm 0.10}$	$0.87_{\pm 0.10}$
	MDPH	$0.98_{\pm 0.00}$	$0.88_{\pm 0.07}$	$0.78_{\pm 0.13}$
	ENR	$0.98_{\pm 0.00}$	$0.91_{\pm 0.01}$	$0.82_{\pm 0.05}$
Apartments	Ours	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$
	MDPH	$0.98_{\pm 0.02}$	$0.94_{\pm 0.03}$	$0.86_{\pm 0.05}$

triple $(x, g, y = g \cdot x)$, we say that 'x hits y' if $\varphi(y)$ is the nearest neighbour in $\mathcal{Z}$ of $g \cdot \varphi(x)$ among a random batch of encodings $\{\varphi(x)\}_{x \in \mathcal{B}}$. For a test set, the hit-rate is then defined as the number of times x hits y divided by the test set size. We set the number of aforementioned random encodings to $|\mathcal{B}| = 32$. For each model, the nearest neighbour is computed with respect to the same latent metric d as the one used for training. In order to test the performance of the models when acted upon multiple times in a row, we generate test sets where g is a trajectory i.e., it is factorized as $g = g_1 g_2 \cdots g_T$ for $T \in \{1, 10, 20\}$. Hit-rate is then computed after sequentially acting by the g_i's in the latent space. This captures the accumulation of errors in the equivariant representation and thus evaluates the performance for long-term predictions. All the test sets consist of 10% of the corresponding dataset.

The results are presented in Table D.2. As can be seen, all the models perform nearly perfectly on single-step predictions with the exception of Linear (89% hit-rate). For the latter the latent group action is not free, which prevents learning a lossless equivariant representation and thus degrades the quality of predictions. On longer trajectories, however, our model outperforms the baselines by an increasing margin. MDPH accumulates errors due to the lack of structure in its latent space: its latent action is learned, which does not guarantee stability with respect to composition of multiple symmetries. The degradation of performance for MDPH is particularly evident in the case of Multi-Sprites (11% hit-rate on 20 steps), which

is probably due to the large number of orbits (27) and the consequent complexity of the prediction task. Our model is instead robust even in presence of many orbits (93% hit-rate on Multi-Sprites) due to the dedicated invariant component $\mathcal{E}$ in its latent space.

When the latent space is equipped with a group action, stability on long trajectories follows from *associativity* of the group composition and the action (see Definition D.2.1 and D.2.2). This is evident from the results for the Chairs dataset, where our model and Linear outperform MDPH on longer trajectories (94% and 87% against 78% hit-rate on 20 steps) and exhibit a stable hit-rate as the number of steps increases. Even though ENR carries a latent group action, it still accumulates errors (82% hit-rate on 20 steps) due to the discretization of the its latent space i.e., the latent grid acted upon by SO(3). The consequent interpolation makes the latent action only approximately associative, causing errors to accumulate on long trajectories.

D.6 Conclusions, Limitations and Future Work

In this work we addressed the problem of learning equivariant representations by decomposing the latent space into a group component and an invariant one. We theoretically showed that our representations are lossless, disentangled and preserve the geometry of data. We empirically validated our approach on a variety of groups, compared it to other equivariant representation learning frameworks and discussed applications to the problem of scene mapping.

Our formalism builds on the assumption that the group of symmetries is known a priori and not inferred from data. This is viable in applications to robotics, but can be problematic in other domains. If a data feature is not taken into account among symmetries, it will formally define distinct orbits. For example, the eventual change in texture for images of rigid objects has to be part of the symmetries in order to still maintain shapes as the only intrinsic classes. A framework where the group structure is learned might be a feasible, although less interpretable alternative to prior symmetry knowledge that constitutes an interesting line of future investigation.

D.7 Acknowledgements

This work was supported by the Swedish Research Council, Knut and Alice Wallenberg Foundation and the European Research Council (ERC-BIRD-884807).

D.8 Appendix

D.8.1 Proofs of Theoretical Results

Proposition D.8.1. *The following holds:*

- *There is an equivariant isomorphism*

$$\mathcal{X} \simeq (\mathcal{X}/G) \times G \tag{D.7}$$

  where G acts trivially on the orbits and via multiplication on itself, i.e., $g \cdot (e, h) = (e, gh)$ for $g, h \in G$, $e \in \mathcal{X}/G$. In other words, each orbit can be identified equivariantly with the group itself.

- *Any equivariant map $\varphi\colon\ (\mathcal{X}/G) \times G \to (\mathcal{X}/G) \times G$ is a right multiplication on each orbit i.e., for each orbit $O \in \mathcal{X}/G$ there is an $h_O \in G$ such that $\varphi(O, g) = (O', gh_O)$ for all $g \in G$. In particular, if φ induces a bijection on orbits then it is an isomorphism.*

Proof. We start by proving the first statement. Choose a system of representatives $\mathcal{S} \subseteq \mathcal{X}$ for orbits, that is $\mathcal{S}$ contains exactly one element for each class. Consider the map $f\colon\ (\mathcal{X}/G) \times G \to \mathcal{X}$ given, for $s \in \mathcal{S}$ and $g \in G$, by $f([s], g) = g \cdot s$, where $[s]$ denotes the orbit of s. It is straightforward to check that f is indeed equivariant. Now if $f([s], g) = f([t], h)$ for $s, t \in \mathcal{S}$ and $g, h \in G$ then $g \cdot s = h \cdot t$ and s, t are thus in the same orbit, which implies $s = t$ because of uniqueness of representatives. But then $h \cdot s = g \cdot s$ and, equivalently, $g^{-1}h \cdot s = s$, from which we deduce $g = h$ since the action is free. This shows that f is injective. Finally, for $x \in \mathcal{X}$, one can write $x = g \cdot s$ for the representative s of the orbit of x, which means that $x = f([s], g)$. That is, f is surjective and thus also bijective, which concludes the proof of the first statement.

We now prove the second statement. Consider an equivariant map $\varphi\colon\ (\mathcal{X}/G) \times G \to (\mathcal{X}/G) \times G$. For each orbit $O \in (\mathcal{X}/G)$ denote by $h_O \in G$ the element such that $\varphi(O, 1) = (O', h_O)$. Then by equivariance $\varphi(O, g) = \varphi(O, g1) = (O', gh_O)$, as desired. □

D.8.2 Description of Datasets

In our experiments we deploy the following datasets, which are also summarized in Table D.1:

Sprites: extracted from *dSprites* [35]. It consists of grayscale images depicting three sprites (heart, square, ellipse) translating and dilating in the pixel plane. The group of symmetries is $G = \mathbb{R}^3$: a factor $\mathbb{R}^2$ translates the sprites in the pixel plane while the last copy of $\mathbb{R}$, which is isomorphic via exponentiation to $\mathbb{R}_{>0}$ equipped

with multiplication, acts through dilations. The dataset size is 3×10^4 and there are three orbits, each corresponding to a sprite.

Shapes: extracted from *3DShapes* [36]. It consists of colored images depicting four objects (cube, cylinder, sphere, pill) on a background divided into wall, floor and sky. Again, $G = \mathbb{R}^3$ but with the action given by color shifting: each of the factors $\mathbb{R}$ acts by changing the color of the corresponding scene component among object, wall and floor. The dataset size is 4×10^4 and there are four orbits, each corresponding to a shape.

Multi-Sprites: obtained from Sprites by overlapping images of three colored sprites (with fixed scale). The group of symmetries is $G = \mathbb{R}^6 = \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2$, each of whose factors $\mathbb{R}^2$ translates one of the three sprites in the pixel plane. The added colors endow the sprites with an implicit ordering, which is necessary for the action to be well-defined. The dataset size is 3×10^4. Since the scene is composed by three possibly repeating sprites, there are $3^3 = 27$ orbits corresponding to the different configurations.

Chairs: extracted from *ShapeNet* [37]. It consists of colored images depicting three types of chair from different angles. The group of symmetries is $G = \mathrm{SO}(3)$, which rotates the depicted chair. The dataset size is 3×10^4 and there are three orbits, each corresponding to a type of chair.

Apartments: extracted from *Gibson* [38] and generated via the *Habitat* simulator [39]. It consists of colored images of first-person renderings of two apartments ('Elmira' and 'Convoy'). The data simulate the visual perception of an agent such as a mobile robot exploring the two apartments and collecting images and symmetries. The latter belong to the group of two-dimensional orientation-preserving Euclidean isometries $G = \mathbb{R}^2 \times \mathrm{SO}(2)$ and coincide with the possible moves (translations and rotations) by part of the agent. One can realistically imagine the agent perceiving the symmetries through some form of odometry i.e., measurement of movement. Note that the action by G on $\mathcal{X}$ is *partially* defined since $g \cdot x$ is not always possible because of obstacles. As long as the agent is able to reach any part of each of the apartments, the latter still coincide with the two orbits of the group action. The dataset size 2×10^4.

All the datasets consist of triples (x, g, y) where x, y are 64×64 images, $g \in G$ and $y = g \cdot x$.

References

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[3] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *ICML workshop on unsupervised and transfer learning*, JMLR, 2012.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, PMLR, 2020.

[5] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[6] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *ICML*, PMLR, 2019.

[7] I. Higgins, S. Racanière, and D. Rezende, "Symmetry-based representations for artificial and biological general intelligence," *Frontiers in Computational Neuroscience*, p. 28, 2022.

[8] Y. Bengio, T. Deleu, N. Rahaman, R. Ke, S. Lachapelle, O. Bilaniuk, A. Goyal, and C. Pal, "A meta-transfer objective for learning to disentangle causal mechanisms," *arXiv preprint arXiv:1901.10912*, 2019.

[9] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *arXiv preprint arXiv:1812.02230*, 2018.

[10] X. Guo, E. Zhu, X. Liu, and J. Yin, "Affine equivariant autoencoder.," in *IJCAI*, 2019.

[11] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Interpretable transformations with encoder-decoder networks," in *ICCV*, IEEE, 2017.

[12] R. Quessard, T. Barrett, and W. Clements, "Learning group structure and disentangled representations of dynamical environments," in *NeurIPS*, 2020.

[13] T. S. Cohen and M. Welling, "Group equivariant convolutional networks," in *ICML*, PMLR, 2016.

[14] T. S. Cohen, M. Geiger, and M. Weiler, "A general theory of equivariant cnns on homogeneous spaces," in *NeurIPS*, 2019.

[15] T. Kipf, E. van der Pol, and M. Welling, "Contrastive learning of structured world models," in *ICLR*, 2020.

[16] E. van der Pol, T. Kipf, F. A. Oliehoek, and M. Welling, "Plannable approximations to mdp homomorphisms: Equivariance under actions," in *AAMAS*, 2020.

[17] J. J. Rotman, *An introduction to the theory of groups*, vol. 148. Springer Science & Business Media, 2012.

[18] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[19] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *ICML*, PMLR, 2020.

[20] K. K. Liang, "Efficient conversion from rotating matrix to rotation axis and angle by extending rodrigues' formula," *arXiv preprint arXiv:1810.02999*, 2018.

[21] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *ICLR*, 2017.

[22] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen, "Weakly-supervised disentanglement without compromises," in *International Conference on Machine Learning*, pp. 6348–6359, PMLR, 2020.

[23] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *International conference on artificial neural networks*, pp. 44–51, Springer, 2011.

[24] L. Falorsi, P. de Haan, T. R. Davidson, N. De Cao, M. Weiler, P. Forré, and T. S. Cohen, "Explorations in homeomorphic variational auto-encoding," *arXiv preprint arXiv:1807.04689*, 2018.

[25] L. Tonnaer, L. A. P. Rey, V. Menkovski, M. Holenderski, and J. W. Portegies, "Quantifying and learning linear symmetry-based disentanglement," *ICML*, 2020.

[26] T. Yang, X. Ren, Y. Wang, W. Zeng, and N. Zheng, "Towards building a group-based unsupervised representation disentanglement framework," *arXiv preprint arXiv:2102.10303*, 2021.

[27] E. Dupont, M. B. Martin, A. Colburn, A. Sankar, J. Susskind, and Q. Shan, "Equivariant neural rendering," in *ICML*, PMLR, 2020.

[28] R. Winter, M. Bertolini, T. Le, F. Noé, and D.-A. Clevert, "Unsupervised learning of group invariant and equivariant representations," *NeurIPS*, 2022.

[29] T. S. Cohen, M. Geiger, M. Köhler, and M. Welling, "Spherical CNNs," in *ICLR*, 2018.

[30] H. Maron, O. Litany, G. Chechik, and E. Fetaya, "On learning sets of symmetric elements," in *ICML*, PMLR, 2020.

[31] T. S. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, "Gauge equivariant convolutional networks and the icosahedral cnn," in *ICML*, PMLR, 2019.

[32] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[33] J. Y. Park, O. Biza, L. Zhao, J. W. van de Meent, and R. Walters, "Learning symmetric embeddings for equivariant world models," *ICML*, 2022.

[34] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *ICML*, PMLR, 2017.

[35] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner, "dsprites: Disentanglement testing sprites dataset." https://github.com/deepmind/dsprites-dataset/, 2017.

[36] C. Burgess and H. Kim, "3d shapes dataset." https://github.com/deepmind/3d-shapes/, 2018.

[37] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[38] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson Env: real-world perception for embodied agents," in *CVPR*, IEEE, 2018.

[39] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *ICCV*, IEEE, 2019.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, IEEE, 2016.

Paper E

Equivariant Representation Learning in the Presence of Stabilizers

Luis Armando Pérez Rey*, Giovanni Luca Marchetti*, Danica Kragic, Dmitri Jarnikov, Mike Holenderski

Abstract

We introduce Equivariant Isomorphic Networks (EquIN) – a method for learning representations that are equivariant with respect to general group actions over data. Differently from existing equivariant representation learners, EquIN is suitable for group actions that are not free, i.e., that stabilize data via nontrivial symmetries. EquIN is theoretically grounded in the orbit-stabilizer theorem from group theory. This guarantees that an ideal learner infers isomorphic representations while trained on equivariance alone and thus fully extracts the geometric structure of data. We provide an empirical investigation on image datasets with rotational symmetries and show that taking stabilizers into account improves the quality of the representations.

E.1 Introduction

Incorporating data symmetries into deep neural representations defines a fundamental challenge and has been addressed in several recent works [1–5]. The overall aim is to design representations that preserve symmetries and operate coherently with respect to them – a functional property known as *equivariance*. This is because the preservation of symmetries leads to the extraction of geometric and semantic structures in data, which can be exploited for data efficiency and generalization [6]. As an example, the problem of *disentangling* semantic factors of variation in data has

been rephrased in terms of equivariant representations [7,8]. As disentanglement is known to be unfeasible with no inductive biases or supervision [9], symmetries of data arise as a geometric structure that can provide weak supervision and thus be leveraged in order to disentangle semantic factors.

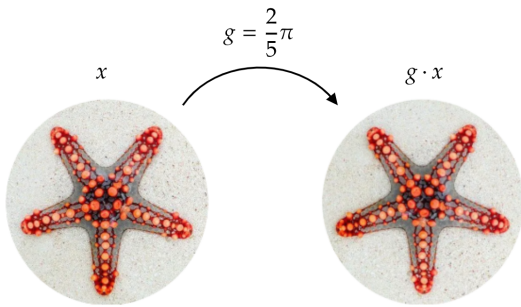


Figure E.1: An example of an action on data that is not free. The datapoint x is stabilized by the symmetry $g \in G$.

The majority of models from the literature rely on the assumption that the group of symmetries acts *freely* on data [10] i.e., that no datapoint is stabilized by nontrivial symmetries. This avoids the need to model *stabilizers* of datapoints, which are unknown subgroups of the given symmetry group. However, non-free group actions arise in several practical scenarios. This happens, for example, when considering images of objects acted upon by the rotation group via a change of orientation. Such objects may be symmetrical, resulting in rotations leaving the image almost identical and consequently ambiguous in its orientation, see Figure E.1. Discerning the correct orientations of an object is important for applications such as pose estimation [11] and reinforcement learning [12]. This motivates the need to design equivariant representation learning frameworks that are capable of modeling stabilizers and therefore suit non-free group actions.

In this work, we propose a method for learning equivariant representation for general and potentially non-free group actions. Based on the Orbit-Stabilizer Theorem from group theory, we design a model that outputs subsets of the group, which represent the stabilizer subgroup up to a symmetry – a group theoretical construction known as *coset*. The representation learner optimizes an equivariance loss relying on supervision from symmetries alone. This means that we train our model on a dataset consisting of relative symmetries between pairs of datapoints, avoiding the need to know the whole group action over data a priori. From a theoretical perspective, the above-mentioned results from group theory guarantee that an ideal learner infers representations that are isomorphic to the original dataset. This implies that our representations completely preserve the symmetry structure

while preventing any loss of information. We name our framework Equivariant Isomorphic Networks – EquIN for short. In summary, our contributions include:

- A novel equivariant representation learning framework suitable for non-free group actions.
- A discussion grounded on group theory with theoretical guarantees for isomorphism representations.
- An empirical investigation with comparisons to competing equivariant representation learners on image datasets.

We provide Python code implementing our framework together with all the experiments at the following repository: `luis-armando-perez-rey/non-free`.

E.2 Related Work

In this section, we first briefly survey representation learning methods from the literature leveraging on equivariance. We then draw connections between equivariant representations and world models from reinforcement learning and discuss the role of equivariance in terms of disentangling semantic factors of data.

Equivariant Representation Learning. Several works in the literature have proposed and studied representation learning models that are equivariant with respect to a group of data symmetries. These models are typically trained via a loss encouraging equivariance on a dataset of relative symmetries between datapoints. What distinguishes the models is the choice of the latent space and of the group action over the latter. Euclidean latent spaces with linear or affine actions have been explored in [1,13,14]. However, the intrinsic data manifold is non-Euclidean in general, leading to representations that are non-isomorphic and that do not preserve the geometric structure of the data. To amend this, a number of works have proposed to design latent spaces that are isomorphic to disjoint copies of the symmetry group [4, 10, 15, 16]. When the group action is free, this leads to isomorphic representations and thus completely recovers the geometric structure of the data [10]. However, the proposed latent spaces are unsuitable for non-free actions. Since they do not admit stabilizers, no equivariant map exists, and the model is thus unable to learn a suitable representation. In the present work, we extend this line of research by designing a latent space that enables learning equivariant representations in the presence of stabilizers. Our model implicitly represents stabilizer subgroups and leads to isomorphic representations for arbitrary group actions.

Latent World Models. Analogously to group actions, Markov Decision Processes (MDPs) from reinforcement learning and control theory involve a, possibly stochastic, interaction with an environment. This draws connections between MDPs and

symmetries since the latter can be thought of as transformations and, thus, as a form of interaction. The core difference is that in an MDP, no algebraic structure, such as a group composition, is assumed on the set of interactions. In the context of MDPs, a representation that is equivariant with respect to the agent's actions is referred to as latent *World Model* [12, 17, 18] or *Markov Decision Process Homomorphism* (MDPH) [19]. In an MDPH the latent action is learned together with the representation by an additional model operating on the latent space. Although this makes MDPHs more general than group-equivariant models, the resulting representation is unstructured and uninterpretable. The additional assumptions of equivariant representations translate instead into the preservation of the geometric structure of data.

Disentanglement. As outlined in [6], a desirable property for representations is disentanglement, i.e., the ability to decompose in the representations the semantic factors of variations that explain the data. Although a number of methods have been proposed for this purpose [20, 21], it has been shown that disentanglement is mathematically unachievable in an unbiased and unsupervised way [9]. As an alternative, the notion has been rephrased in terms of symmetry and equivariance [7]. It follows that isomorphic equivariant representations are guaranteed to be disentangled in this sense [4, 10]. Since we aim for general equivariant representations that are isomorphic, our proposed method achieves disentanglement as a by-product.

E.3 Group Theory Background

We review the fundamental group theory concepts necessary to formalize our representation learning framework. For a complete treatment, we refer to [22].

Definition E.3.1. A group is a set G equipped with a *composition map* $G \times G \to G$ denoted by $(g, h) \mapsto gh$, an *inversion map* $G \to G$ denoted by $g \mapsto g^{-1}$, and a distinguished *identity element* $1 \in G$ such that for all $g, h, k \in G$:

Associativity	*Inversion*	*Identity*
$g(hk) = (gh)k$	$g^{-1}g = gg^{-1} = 1$	$g1 = 1g = g$

Elements of a group represent abstract symmetries. Spaces with a group of symmetries G are said to be acted upon by G in the following sense.

Definition E.3.2. An action by a group G on a set $\mathcal{X}$ is a map $G \times \mathcal{X} \to \mathcal{X}$ denoted by $(g, x) \mapsto g \cdot x$, satisfying for all $g, h \in G,\ x \in \mathcal{X}$:

Associativity	*Identity*
$g \cdot (h \cdot x) = (gh) \cdot x$	$1 \cdot x = x$

Suppose that G acts on a set $\mathcal{X}$. The action defines a set of *orbits* $\mathcal{X}/G$ given by the equivalence classes of the relation $x \sim y$ iff $y = g \cdot x$ for some $g \in G$. For each $x \in \mathcal{X}$, the *stabilizer* subgroup is defined as

$$G_x = \{g \in G \mid g \cdot x = x\}. \tag{E.1}$$

Stabilizers of elements in the same orbit are conjugate, meaning that for each x, y belonging to the same orbit O there exists $h \in G$ such that $G_y = hG_xh^{-1}$. By abuse of notation, we refer to the conjugacy class G_O of stabilizers for $O \in \mathcal{X}/G$. The action is said to be *free* if all the stabilizers are trivial, i.e., $G_O = \{1\}$ for every O.

We now recall the central notion for our representation learning framework.

Definition E.3.3. A map $\varphi : \mathcal{X} \to \mathcal{Z}$ between sets acted upon by G is *equivariant* if $\varphi(g \cdot x) = g \cdot \varphi(x)$ for every $x \in \mathcal{X}$ and $g \in G$. An equivariant bijection is referred to as *isomorphism*.

Intuitively, an equivariant map between $\mathcal{X}$ and $\mathcal{Z}$ preserves their corresponding symmetries. The following is the fundamental result on group actions [22].

Theorem E.3.1 (Orbit-Stabilizer)**.** *The following holds:*

- *Each orbit O is isomorphic to the set of (left)* cosets $G/G_O = \{gG_O \mid g \in G\}$. *In other words, there is an isomorphism:*

$$\mathcal{X} \simeq \coprod_{O \in \mathcal{X}/G} G/G_O \quad \subseteq 2^G \times \mathcal{X}/G \tag{E.2}$$

  where 2^G denotes the power-set of G on which G acts by left multiplication i.e., $g \cdot A = \{ga \mid a \in A\}$.

- *Any equivariant map*

$$\varphi : \mathcal{X} \to \coprod_{O \in \mathcal{X}/G} G/G_O \tag{E.3}$$

  that induces a bijection on orbits is an isomorphism.

Theorem E.3.1 describes arbitrary group actions completely and asserts that orbit-preserving equivariant maps are isomorphisms. Our central idea is to leverage on this in order to design a representation learner that is guaranteed to be isomorphic when trained on equivariance alone.

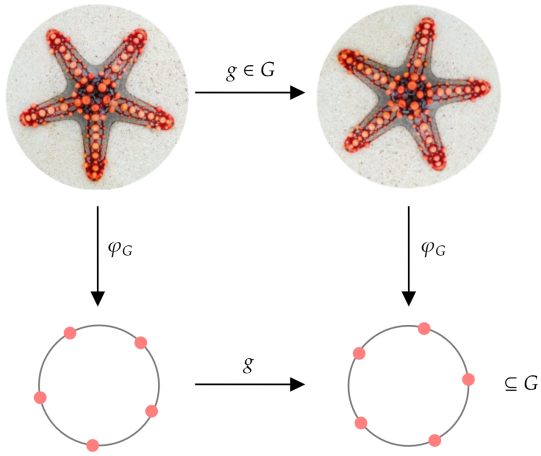


Figure E.2: An illustration of EquIN encoding data equivariantly as subsets of the symmetry group G. This results in representations that are suitable even when the action by G on data is not free.

E.4 Equivariant Isomorphic Networks (EquIN)

Our goal is to design an equivariant representation learner based on Theorem E.3.1. We aim to train a model

$$\varphi : \mathcal{X} \to \mathcal{Z} \tag{E.4}$$

with a latent space $\mathcal{Z}$ on a loss encouraging equivariance. The ideal choice for $\mathcal{Z}$ is given by $\coprod_{O \in \mathcal{X}/G} G/G_O$ since the latter is isomorphic to $\mathcal{X}$ (Theorem E.3.1). In other words, φ ideally outputs cosets of stabilizers of the input datapoints. However, while we assume that G is known a priori, its action on $\mathcal{X}$ is not and has to be inferred from data. Since the stabilizers depend on the group action, they are unknown a priori as well. In order to circumvent the modeling of stabilizers and their cosets, we rely on the following simple result:

Proposition E.4.1. *Let $\varphi : \mathcal{X} \to 2^G$ be an equivariant map. Then for each $x \in \mathcal{X}$ belonging to an orbit O, $\varphi(x)$ contains a coset of (a conjugate of) G_O.*

Proof. Pick $x \in \mathcal{X}$. Then for every $g \in G_x$ it holds that $\varphi(x) = \varphi(g \cdot x) = g \cdot \varphi(x)$. In other words, $G_x h = hh^{-1} G_x h \subseteq \varphi(x)$ for each $h \in \varphi(x)$. Since $h^{-1} G_x h$ is conjugate to G_x the thesis follows. □

Proposition E.4.1 enables φ to output arbitrary subsets of G instead of cosets of stabilizers. As long as those subsets are *minimal* w.r.t. to inclusion, they will coincide with the desired cosets.

Based on this, we define the latent space of EquIN as $\mathcal{Z} = \mathcal{Z}_G \times \mathcal{Z}_O$ and implement the map φ as a pair of neural networks $\varphi_G : \mathcal{X} \to \mathcal{Z}_G$ and $\varphi_O : \mathcal{X} \to \mathcal{Z}_O$. The component $\mathcal{Z}_G$ represents cosets of stabilizers while $\mathcal{Z}_O$ represents orbits. Since the output space of a neural network is finite-dimensional, we assume that the stabilizers of the action are finite. The model φ_G then outputs N elements

$$\varphi_G(x) = \{\varphi_G^1(x), \cdots, \varphi_G^N(x)\} \subseteq G \tag{E.5}$$

where $\varphi_G^i(x) \in G$ for all i. The hyperparameter N should be ideally chosen larger than the cardinality of the stabilizers. On the other hand, the output of φ_O consists of a vector of arbitrary dimensionality. The only requirement is that the output space of φ_O should have enough capacity to contain the space of orbits $\mathcal{X}/G$.

E.4.1 Parametrizing G via the Exponential Map

The output space of usual machine learning models such as deep neural networks is Euclidean. However, φ_G needs to output elements of the group G (see Equation E.5), which may be non-Euclidean as in the case of $G = \mathrm{SO}(n)$. Therefore, in order to implement φ_G, it is necessary to parametrize G. To this end, we assume that G is a differentiable manifold, with differentiable composition and inversion maps, i.e., that G is a *Lie group*. One can then define the *Lie algebra* $\mathfrak{g}$ of G as the tangent space to G at 1.

We propose to rely on the *exponential map* $\mathfrak{g} \to G$, denoted by $v \mapsto e^v$, to parametrize G. This means that φ_G first outputs N elements

$$\varphi_G(x) = \{v^1, \cdots, v^N\} \subseteq \mathfrak{g} \tag{E.6}$$

that get subsequently mapped into G as $\{e^{v^1}, \cdots, e^{v^N}\}$. Although the exponential map can be defined for general Lie groups by solving an appropriate ordinary differential equation, we focus on the case $G \subseteq \mathrm{GL}(n)$. The Lie algebra $\mathfrak{g}$ is then contained in the space of $n \times n$ matrices and the exponential map amounts to the matrix Taylor expansion

$$e^v = \sum_{k \geq 0} \frac{v^k}{k!} \tag{E.7}$$

where v^k denotes the power of v as a matrix. For specific groups, the latter can be simplified via simple closed formulas. For example, the exponential map of $\mathbb{R}^n$ is the identity while for SO(3) it can be efficiently computed via the Rodrigues' formula [23].

E.4.2 Training Objective

As mentioned, our dataset $\mathcal{D}$ consists of samples from the unknown group action. This means that datapoints are triplets $(x, g, y) \in \mathcal{X} \times G \times \mathcal{X}$ with $y = g \cdot x$. Given

a datapoint $(x, g, y) \in \mathcal{D}$ the learner φ_G optimizes the equivariance loss over its parameters:

$$\mathcal{L}_G(x, g, y) = d(g \cdot \varphi_G(x),\ \varphi_G(y)) \tag{E.8}$$

where d is a semi-metric for sets. We opt for the asymmetric *Chamfer distance*

$$d(A, B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} d_G(a, b) \tag{E.9}$$

because of its differentiability properties. Any other differentiable distance between sets of points can be deployed as an alternative. Here d_G is a metric on G and is typically set as the squared Euclidean for $G = \mathbb{R}^n$ and as the squared Frobenius for $G = \mathrm{SO}(n)$. As previously discussed, we wish $\varphi_G(x)$, when seen as a set, to be minimal in cardinality. To this end, we add the following regularization term measuring the discrete entropy:

$$\widetilde{\mathcal{L}}_G(x) = \frac{\lambda}{N^2} \sum_{1 \leq i,j \leq N} d_G(\varphi_G^i(x),\ \varphi_G^j(x)) \tag{E.10}$$

where λ is a weighting hyperparameter. On the other hand, since orbits are invariant to the group action φ_O optimizes a *contrastive loss*. We opt for the popular InfoNCE loss from the literature [24]:

$$\mathcal{L}_O(x, y) = d_O(\varphi_O(x),\ \varphi_O(y)) + \log \mathbb{E}_{x'} \left[e^{-d_O(\varphi_O(x'),\ \varphi_O(x))} \right] \tag{E.11}$$

where x' is marginalized from $\mathcal{D}$. As customary for the InfoNCE loss, we normalize the output of φ_O and set $d_O(a, b) = -\cos(\angle ab) = -a \cdot b$. The second summand of $\mathcal{L}_O$ encourages injectivity of φ_O and as such prevents orbits from overlapping in the representation.

The Orbit-Stabilizer Theorem (Theorem E.3.1) guarantees that if EquIN is implemented with ideal learners φ_G, φ_O then it infers isomorphic representations in the following sense. If the $\mathcal{L}_G(x, g, y)$ and the first summand of $\mathcal{L}_O(x, y)$ vanish for every (x, g, y) then φ is equivariant. If moreover the regularizations, $\widetilde{\mathcal{L}}_G$ and the second summand of $\mathcal{L}_O$, are at a minimum then $\varphi_G(x)$ coincides with a coset of G_O for every $x \in O$ (Proposition E.4.1) and φ_O is injective. The second claim of Theorem E.3.1 implies then that the representation is isomorphic on its image, as desired.

E.5 Experiments

We empirically investigate EquIN on image data acted upon by a variety Lie groups. Our aim is to show both qualitatively and quantitatively that EquIN reliably infers isomorphic equivariant representations for non-free group actions.

We implement the neural networks φ_G and φ_O as a ResNet18 [25]. For a datapoint $x \in \mathcal{X}$, the network implements multiple heads to produce embeddings $\{\varphi_G^1(x), \cdots, \varphi_G^N(x)\} \subseteq G$. The output dimension of φ_O is set to 3. We train the model for 50 epochs using the AdamW optimizer [26] with a learning rate of 10^{-4} and batches of 16 triplets $(x, g, y) \in \mathcal{D}$.

E.5.1 Datasets

We consider the following datasets consisting of 64×64 images subject to non-free group actions. Samples from these datasets are shown in Figure E.4.

Rotating Arrows: images of radial configurations of $\nu \in \{1, 2, 3, 4, 5\}$ arrows rotated by $G = \mathrm{SO}(2)$. The number of arrows ν determines the orbit and the corresponding stabilizer is (isomorphic to) the cyclic group C_ν of cardinality ν. The dataset contains 2500 triplets (x, g, y) per orbit.

Colored Arrows: images similar to Rotating Arrows but with the arrows of five different colors. This extra factor produces additional orbits with the same stabilizer subgroups. The number of orbits is therefore 25. The dataset contains 2000 triplets per orbit.

Double Arrows: images of two radial configurations of $2, 3$ and $3, 5$ arrows respectively rotated by the torus $G = \mathrm{SO}(2) \times \mathrm{SO}(2)$. The action produces two orbits with stabilizers given by products of cyclic groups: $C_2 \times C_3$ and $C_3 \times C_5$ respectively. The dataset contains 2000 triplets per orbit.

ModelNet: images of monochromatic objects from ModelNet40 [27] rotated by $G = \mathrm{SO}(2)$ along an axis. We consider five objects: an airplane, a chair, a lamp, a bathtub and a stool. Each object corresponds to a single orbit. The lamp, the stool and the chair have the cyclic group C_4 as stabilizer while the action over the airplane and the bathub is free. The dataset contains 2500 triplets per orbit.

Solids: images of a monochromatic tetrahedron, cube and icosahedron [28] rotated by $G = \mathrm{SO}(3)$. Each solid defines an orbit, and the stabilizers of the tetrahedron, the cube, and the icosahedron are subgroups of order 12, 24 and 60 respectively. The dataset contains 7500 triplets per orbit.

E.5.2 Comparisons

We compare EquIN with the following two equivariant representation learning models.

Baseline: a model corresponding to EquIN with $N = 1$ where φ_G outputs a single element of G. The latent space is $\mathcal{Z} = G \times \mathcal{Z}_O$, on which G acts freely. We deploy

this as the baseline since it has been proposed with minor variations in a number of previous works [4, 8, 10, 29] assuming free group actions.

Equivariant Neural Renderer (ENR): a model from [30] implementing a tensorial latent space $\mathcal{Z} = \mathbb{R}^{S^3}$, thought as a scalar signal space on a $S \times S \times S$ grid in $\mathbb{R}^3$. The group SO(3) act *approximately* on $\mathcal{Z}$ by rotating the grid and interpolating the obtained values. The model is trained jointly with a decoder $\psi : \mathcal{Z} \to \mathcal{X}$ and optimizes a variation of the equivariance loss that incorporates reconstruction: $\mathbb{E}_{x,g,y=g\cdot x}[d_{\mathcal{X}}(y,\ \psi(g \cdot \varphi(x)))]$ where $d_{\mathcal{X}}$ is the binary cross-entropy for normalized images. Although the action on $\mathcal{Z}$ is free, the latent discretization and consequent interpolation make the model only approximately equivariant. Similarly to EquIN, we implement ENR as ResNet18. As suggested in the original work [30] we deploy 3D convolutional layers around the latent and set to zero the latent dimensions outside a ball. We set $S = 8$ with 160 non-zero latent dimensions since this value is comparable to the latent dimensionality of EquIN, between 7 and 250 dimensions depending on N, making the comparison fair. Note that ENR is inapplicable to Double Arrows since its symmetry group is not naturally embedded into SO(3).

E.5.3 Quantitative Results

In order to quantitatively compare the models, we rely on the following evaluation metrics computed on a test dataset $\mathcal{D}_{\text{test}}$ consisting of 10% of the corresponding training data:

Hit-Rate: a standard score comparing equivariant representations with different latent space geometries [17]. Given a test triple $(x, g, y = g \cdot x) \in \mathcal{D}_{\text{test}}$, we say that '$x$ hits y' if $\varphi(y)$ is the nearest neighbor in $\mathcal{Z}$ of $g \cdot \varphi(x)$ among a random batch of encodings $\{\varphi(x)\}_{x \in \mathcal{B}}$ with $|\mathcal{B}| = 20$. The hit-rate is then defined as the number of times x hits y divided by the test set size. For each model, the nearest neighbor is computed with respect to the same latent metric d as the one used for training.

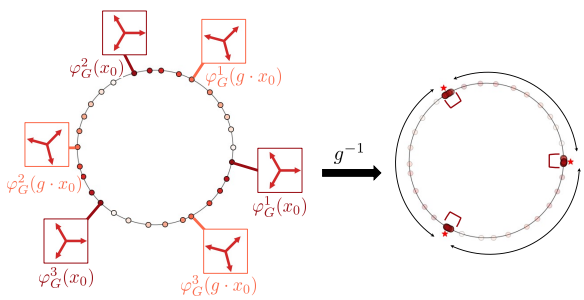


Figure E.3: Diagram explaining the estimation of the disentanglement metric for EquIN. This example assumes that $G = \text{SO}(2)$ and that A is the identity.

Higher values of the metric are better.

Disentanglement: an evaluation metric proposed in [4] to measure disentanglement according to the symmetry-based definition of [7]. This metric is designed for groups in the form $G = \mathrm{SO}(2)^T$ and therefore is inapplicable to the SOLIDS dataset. Per orbit, the test set is organized into datapoints of the form $y = g \cdot x_0$ where x_0 is an arbitrary point in the given orbit. In order to compute the metric, the test dataset is encoded into $\mathcal{Z}$ via the given representation and then projected to $\mathbb{R}^{2T}$ via principal component analysis. Then for each independent copy of $\mathrm{SO}(2) \subseteq G$, a group action on the corresponding copy of $\mathbb{R}^2$ is inferred by fitting parameters via a grid search. Finally, the metric computes the average dispersion of the transformed embeddings as the variance of $g^{-1} \cdot A\varphi_G(y)$. For EquIN, we propose a modified version accounting for the fact that φ_G produces multiple points in G using the Chamfer distance d and averaging the dispersion with respect to each transformed embedding, see Figure E.3. The formula for computing the metric is given by:

$$\mathbb{E}_{y,y'}[d(h^{-1} \cdot A\varphi_G(y'),\ g^{-1} \cdot A\varphi_G(y))] \tag{E.12}$$

where $y = g \cdot x_0$ and $y' = h \cdot x_0$. Lower values of the metric are better.

The results are summarized in Table E.1. EquIN achieves significantly better scores than the baseline. The latter is unable to model the stabilizers in its latent space, leading to representations of poor quality and loss of information. ENR is instead competitive with EquIN. Its latent space suits non-free group actions since stabilizers can be modelled as signals over the latent three-dimensional grid. ENR achieves similar values of hit-rate compared to EquIN. The latter generally outperforms ENR, especially on the MODELNET dataset, while is outpermformed on ROTATING ARROWS. According to the disentanglement metric, EquIN achieves significantly lower scores than ENR. This is probably due to the fact the latent group action in ENR is approximate, making the model unable to infer representations that are equivariant at a granular scale.

E.5.4 Qualitative Results

We provide a number of visualizations as a qualitative evaluation of EquIN. Figure E.4 illustrates the output of φ_G on the various datasets. As can be seen, EquIN correctly infers the stabilizers i.e., the cyclic subgroups of SO(2) and the subgroup of SO(3) of order 12. When N is larger than the ground-truth cardinalities of stabilizers, the points φ_G^i are overlapped and collapse to the number of stabilizers as expected. Figure E.5 displays the output of φ_O for data from COLORED ARROWS. The orbits are correctly separated in $\mathcal{Z}_O$. Therefore, the model is able to distinguish data due to variability in the number ν of arrows as well as in their color.

Table E.1: Mean and standard deviation of the metrics across five repetitions. The number juxtaposed to the name of EquIN indicates the cardinality N of the output of φ_G.

Dataset	Model	Disentanglement (↓)	Hit-Rate (↑)
Rotating Arrows	Baseline	$1.582_{\pm 0.013}$	$0.368_{\pm 0.004}$
	EquIN5	$0.009_{\pm 0.005}$	$0.880_{\pm 0.021}$
	EquIN10	$0.092_{\pm 0.063}$	$0.857_{\pm 0.050}$
	ENR	$0.077_{\pm 0.028}$	$0.918_{\pm 0.009}$
Colored Arrows	Baseline	$1.574_{\pm 0.007}$	$0.430_{\pm 0.004}$
	EquIN5	$0.021_{\pm 0.015}$	$0.930_{\pm 0.055}$
	EquIN10	$0.001_{\pm 0.001}$	$0.976_{\pm 0.005}$
	ENR	$0.106_{\pm 0.032}$	$0.949_{\pm 0.018}$
Double Arrows	Baseline	$1.926_{\pm 0.019}$	$0.023_{\pm 0.004}$
	EquIN6	$0.028_{\pm 0.006}$	$0.512_{\pm 0.011}$
	EquIN15	$0.004_{\pm 0.001}$	$0.820_{\pm 0.104}$
	EquIN20	$0.002_{\pm 0.001}$	$0.934_{\pm 0.020}$
ModelNet	Baseline	$1.003_{\pm 0.228}$	$0.538_{\pm 0.086}$
	EquIN4	$0.012_{\pm 0.022}$	$0.917_{\pm 0.074}$
	EquIN10	$0.003_{\pm 0.001}$	$0.910_{\pm 0.011}$
	ENR	$0.037_{\pm 0.038}$	$0.817_{\pm 0.085}$
Solids	Baseline	-	$0.123_{\pm 0.007}$
	EquIN12	-	$0.126_{\pm 0.004}$
	EquIN24	-	$0.139_{\pm 0.056}$
	EquIN60	-	$0.596_{\pm 0.106}$
	EquIN80	-	$\mathbf{0.795}_{\pm 0.230}$
	ENR	-	$0.772_{\pm 0.095}$

E.5.5 Hyperparameter Analysis

For our last experiment, we investigate the effects of the hyperparameters N and λ when training EquIN on datasets with different numbers of stabilizers.

First, we show that a value of N larger than the cardinality of the stabilizers is necessary to achieve good values of disentanglement, and hit-rate for datasets with non-free group action, see Figure E.6. However, large values of N can result in

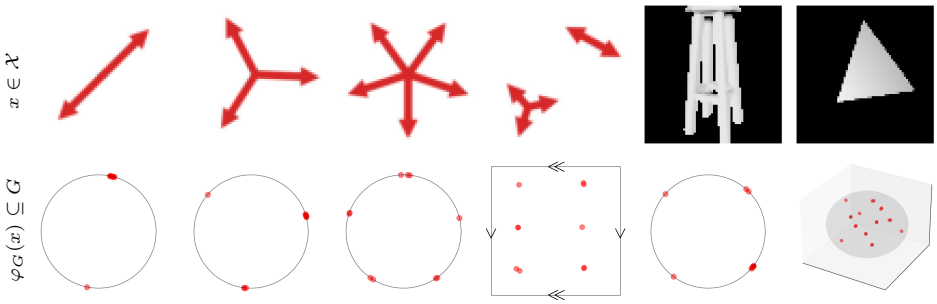


Figure E.4: Visualization of datapoints x and the corresponding predicted (coset of the) stabilizer $\varphi_G(x)$. For Double Arrows, the torus $G = \mathrm{SO}(2) \times \mathrm{SO}(2)$ is visualized as an identified square. For the tetrahedron from Solids, G is visualized as a projective space $\mathbb{RP}^3 \simeq \mathrm{SO}(3)$.

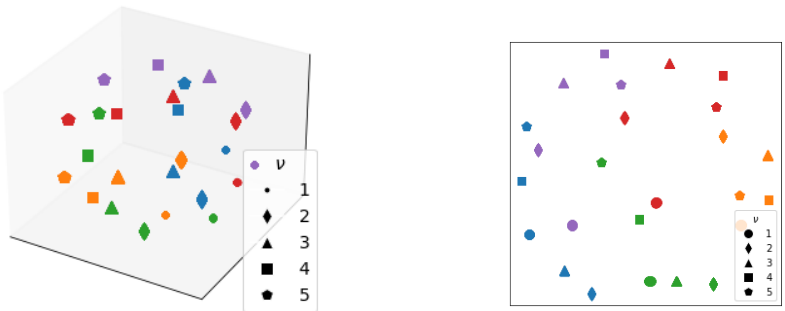


Figure E.5: Embeddings $\varphi_O(x) \in \mathcal{Z}_O \subseteq \mathbb{R}^3$ for x in Colored Arrows. Each symbol represents the ground-truth cardinality $\nu = |G_x|$ of the stabilizer while the color of the symbol represents the corresponding color of the arrow (left). The same embeddings are projected onto $\mathbb{R}^2$ via principal component analysis (right).

non-collapsing embeddings φ_G corresponding to non-minimal cosets of the stabilizers. In these cases, the regularization term of Equation E.10 and its corresponding weight λ plays an important role.

The bottom row of Figure E.7 shows the embeddings $\varphi_G(x)$ learnt for a datapoint $x \in \mathcal{X}$ with stabilizer $G_x \simeq C_2$ of cardinality two. The plots show how for low values of λ, the network converges to a non-minimal set. When an optimal value is chosen, such as $\lambda = 1$, the embeddings obtained with φ_G collapse to a set with the same cardinality as the stabilizers. If λ is too large, the embeddings tend to degenerate and collapse to a single point.

If the value of λ is too small, the discrete entropy of the learnt embeddings is

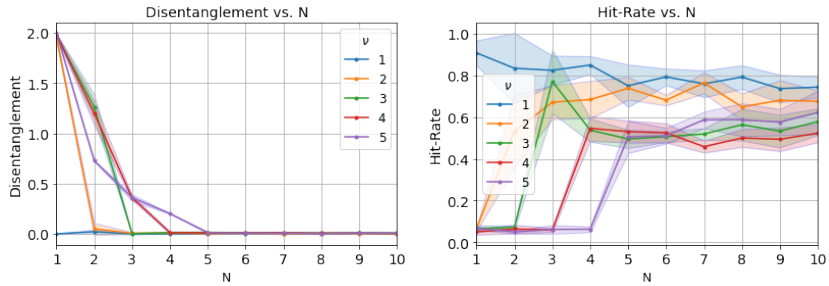


Figure E.6: Disentanglement and hit-rate for models trained with different values of N. Each line in the plot represents the results of a model trained on a dataset with a single orbit whose stabilizer has cardinality ν. The plots show the mean and standard deviation across five repetitions.

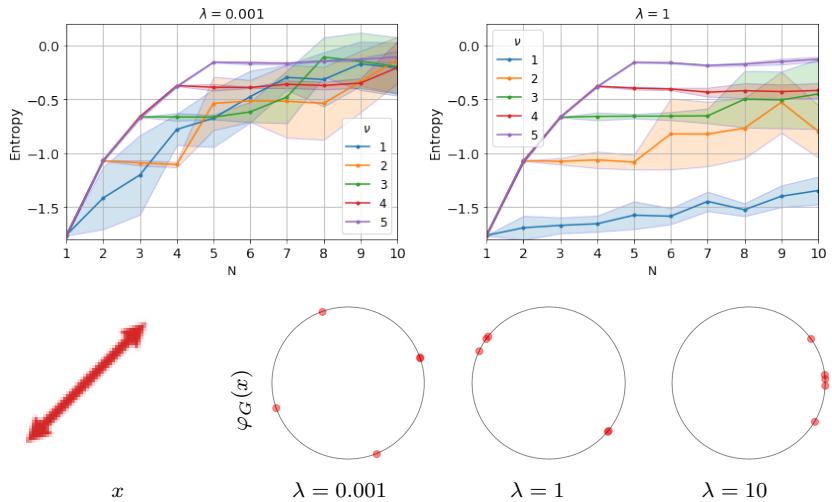


Figure E.7: Discrete entropy for models trained on the arrows dataset with different cardinalities of stabilizer ν and two distinct values of λ (top row). Example embeddings $\varphi_G(x)$ obtained for a datapoint x with two stabilizers obtained with models using $\lambda \in \{0.001, 1, 10\}$ (bottom row).

not restricted. It continues to increase even if the number of embeddings matches the correct number of stabilizers. When an appropriate value of λ is chosen, the entropy becomes more stable as the embeddings have converged to the correct cardinality.

The plots in Figure E.8 show the inverse relationship between λ and the entropy of the encoder φ_G that describes the collapse of the embeddings. The collapse of

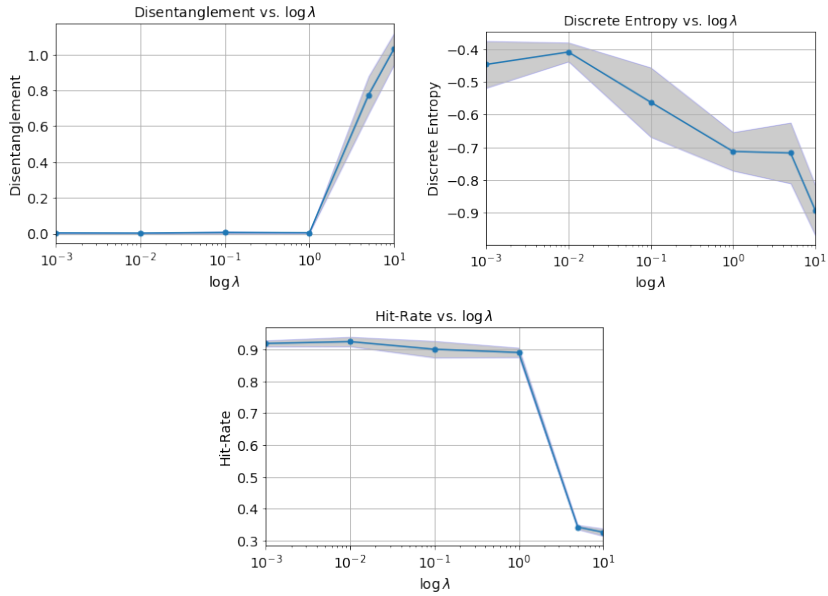


Figure E.8: Disentanglement, discrete entropy and hit-rate for models trained with different values of λ and fixed $N = 5$. The training dataset corresponds to the rotating arrows with $\nu \in \{1, 2, 3, 4, 5\}$. Each line shows the mean and standard deviation across five repetitions.

the embeddings also results in a lower performance of disentanglement and hit-rate by the models as seen for higher values of $\lambda > 1$. Throughout the experiments, we fix the value of $\lambda = 1$ except for SOLIDS where a value of $\lambda = 10$ was chosen since the number N used is larger.

E.6 Conclusions and Future Work

In this work, we introduced EquIN, a method for learning equivariant representations for possibly non-free group actions. We discussed the theoretical foundations and empirically investigated the method on images with rotational symmetries. We showed that our model can capture the cosets of the group stabilizers and separate the information characterizing multiple orbits.

EquIN relies on the assumption that the stabilizers of the group action are finite. However, non-discrete stabilizer subgroups sometimes occur in practice, e.g., in continuous symmetrical objects such as cones, cylinders or spheres. Therefore, an interesting future direction is designing an equivariant representation learner suitable for group actions with non-discrete stabilizers.

E.7 Acknowledgements

This work was supported by the Swedish Research Council, the Knut and Alice Wallenberg Foundation and the European Research Council (ERC-BIRD-884807). This work has also received funding from the NWO-TTW Programme "Efficient Deep Learning" (EDL) P16-25.

References

[1] R. Quessard, T. D. Barrett, and W. R. Clements, "Learning disentangled representations and group structure of dynamical environments," in *Advances in Neural Information Processing Systems*, 2020.

[2] I. Higgins, S. Racanière, and D. Rezende, "Symmetry-based representations for artificial and biological general intelligence," *Frontiers in Computational Neuroscience*, 2022.

[3] T. Cohen and M. Welling, "Learning the irreducible representations of commutative Lie groups," in *International Conference on Machine Learning*, 2014.

[4] L. Tonnaer, L. A. Perez Rey, V. Menkovski, M. Holenderski, and J. Portegies, "Quantifying and Learning Linear Symmetry-Based Disentanglement," in *International Conference on Machine Learning*, 2022.

[5] K. Ahuja, J. Hartford, and Y. Bengio, "Properties from nechanisms: An equivariance perspective on identifiable representation learning," in *International Conference on Learning Representations*, 2022.

[6] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[7] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *arXiv preprint*, 2018.

[8] H. Caselles-Dupré, M. Garcia-Ortiz, and D. Filliat, "Symmetry-based disentangled representation learning requires interaction with environments," in *Advances in Neural Information Processing Systems*, 2019.

[9] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning*, 2019.

[10] G. L. Marchetti, G. Tegnér, A. Varava, and D. Kragic, "Equivariant Representation Learning via Class-Pose Decomposition," *arXiv preprint*, 2022.

[11] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Transactions on Visualization and Computer Graphics*, 2016.

[12] D. Ha and J. Schmidhuber, "World models," *arXiv preprint*, 2018.

[13] X. Guo, E. Zhu, X. Liu, and J. Yin, "Affine equivariant autoencoder.," in *International Joint Conference on Artificial Intelligence*, 2019.

[14] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Interpretable transformations with encoder-decoder networks," in *International Conference on Computer Vision*, 2017.

[15] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning*, Springer, 2011.

[16] L. Falorsi, P. de Haan, T. R. Davidson, N. D. Cao, M. Weiler, P. Forré, and T. S. Cohen, "Explorations in homeomorphic variational auto-encoding," in *ICML18 Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[17] T. Kipf, E. van der Pol, and M. Welling, "Contrastive learning of structured world models," in *International Conference on Learning Representations*, 2020.

[18] J. Y. Park, O. Biza, L. Zhao, J. W. van de Meent, and R. Walters, "Learning symmetric embeddings for equivariant world models," *International Conference on Machine Learning*, 2022.

[19] E. van der Pol, T. Kipf, F. A. Oliehoek, and M. Welling, "Plannable approximations to mdp homomorphisms: Equivariance under actions," in *International Conference on Autonomous Agents and Multi-Agent Systems*, 2020.

[20] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "Beta-vae: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*, 2017.

[21] R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Advances in Neural Information Processing Systems*, 2018.

[22] J. J. Rotman, *An introduction to the theory of groups*, vol. 148. Springer Science & Business Media, 2012.

[23] K. K. Liang, "Efficient conversion from rotating matrix to rotation axis and angle by extending rodrigues' formula," *arXiv preprint*, 2018.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*, 2020.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2 2019.

[27] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[28] K. Murphy, C. Esteves, V. Jampani, S. Ramalingam, and A. Makadia, "Implicit representation of probability distributions on the rotation manifold," in *International Conference on Machine Learning*, 2021.

[29] M. Painter, J. Hare, and A. Prügel-Bennett, "Linear disentangled representations and unsupervised action estimation," in *Advances in Neural Information Processing Systems*, 2020.

[30] E. Dupont, M. B. Martin, A. Colburn, A. Sankar, J. Susskind, and Q. Shan, "Equivariant neural rendering," in *International Conference on Machine Learning*, 2020.

Paper F

Back to the Manifold: Recovering from Out-of-Distribution States

Alfredo Reichlin, Giovanni Luca Marchetti, Hang Yin, Ali Ghadirzadeh, Danica Kragic

Abstract

Learning from previously collected datasets of expert data offers the promise of acquiring robotic policies without unsafe and costly online explorations. However, a major challenge is a distributional shift between the states in the training dataset and the ones visited by the learned policy at the test time. While prior works mainly studied the distribution shift caused by the policy during the offline training, the problem of recovering from out-of-distribution states at the deployment time is not very well studied yet. We alleviate the distributional shift at the deployment time by introducing a recovery policy that brings the agent back to the training manifold whenever it steps out of the in-distribution states, e.g., due to an external perturbation. The recovery policy relies on an approximation of the training data density and a learned equivariant mapping that maps visual observations into a latent space in which translations correspond to the robot actions. We demonstrate the effectiveness of the proposed method through several manipulation experiments on a real robotic platform. Our results show that the recovery policy enables the agent to complete tasks while the behavioral cloning alone fails because of the distributional shift problem.

F.1 Introduction

Data-driven methods in robotics, including reinforcement learning (RL), are often challenged by expensive, slow and unsafe data collection on real systems [1]. Offline solutions, such as *Offline RL* [2] and *Behavior Cloning* (BC) [3], learn a control

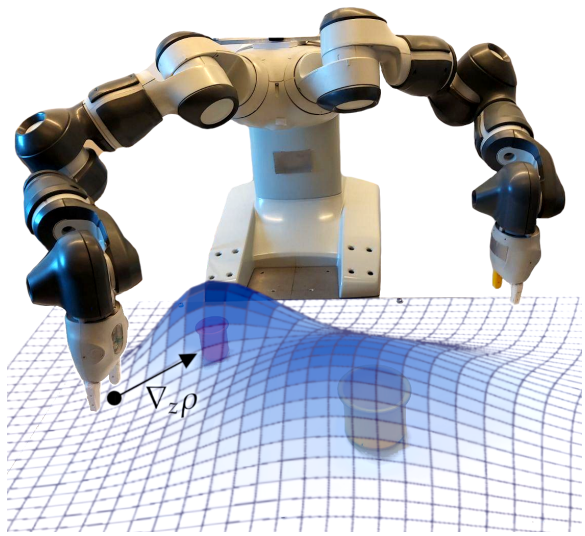


Figure F.1: The proposed recovery policy performs gradient ascent on the estimated density ρ of the demonstrations to get the agent back in-distribution.

policy from a pre-collected dataset hence avoiding the problems of interacting with a physical robot. However, learning from a fixed offline dataset may compromise the capacity of the learner on dealing with novel situations not contained in the training dataset at the test time. Querying the trained policy on such out-of-distribution (OOD) inputs can exacerbate the compounding of the errors when the policy is subsequently applied to states evolved according to the previous state and action [4]. Offline RL avoids this problem by constraining the learned policy to deviate minimally from the policy that collected the data [5–7]. However, there is no mechanism for offline RL to recover from OOD conditions, for example, when starting at a random unseen initial state or being exposed to external perturbations during execution.

One way to improve the performance of the agent in the deployment phase is to optimize an extra objective, e.g., by minimizing the uncertainty-level of the agent in predicting the next state [8, 9] which implicitly helps the agent to stay in-distribution. However, designing such objectives is challenging and they may require learning probabilistic visual forward dynamic models that output a measure of uncertainty.

In this work, we propose a method to augment a policy trained by offline behavior cloning with a *recovery policy* whose actions are computed by a gradient

ascent on the estimated density of the training distribution. This is illustrated in Figure F.1, in which the robot is guided by the recovery policy providing action directions to stay in-distribution while approaching the task object. We achieve this by training a model that encodes input visual observations into a Euclidean latent space, where translations in this space correspond to robot actions, such as Cartesian displacements of the robot end-effector. The encoding has a property known as translational *equivariance* that allows for the conversion of the aforementioned gradient of the estimated training data density into an action. Therefore, the recovery design benefits from a latent representation that (1) is low-dimensional, thus amenable to density estimation, and (2) is task-agnostic, i.e., it can be shared among other tasks.

We empirically demonstrate the feasibility of the proposed method on real-robotic visuomotor policy training tasks. Compared to a behavioral cloning policy, we show that the augmented policy improves the success rate on manipulation tasks. Additionally, when the robot is externally pushed OOD, it allows to recover and successfully complete the task. We also demonstrate how the trained latent equivariant representation can be shared among several tasks, making it task-agnostic. Our main contributions are:

- A method to augment policies trained with offline data to recover from OOD conditions through the gradient of a conditional density estimator.
- An empirical evaluation of the performances of the proposed method on real-robotic manipulation tasks.

F.2 Related Work

Offline policy learning mainly falls into behavior cloning [10] and offline RL [2]. The classic formulation of BC has a number of limitations when learning on real-world data. The most prominent of which, is called the compounding error [11] and occurs due to the sequential nature of the learning framework. This problem is even more profound when learning from small-sized datasets. There have been a number of works targeting this, however, they generally break the fully offline formulation [4, 12, 13].

Offline RL, on the other hand, formulates the problem of learning a policy using offline data from an RL perspective. A naive application of RL on previously collected data results in either an high variance of the learning process [14] or wrong estimates of the expected return [15]. Possible solutions to this involve either constraining the learned policy to minimize the deviation from the one that collected the data (behavioral policy) [5,16,17] or incentivizing the policy to avoid actions on the boundary of the training distribution by changing the reward function [6,7,18]. Moreover, in case the agent happens to step OOD, or it is forcefully brought there,

there is no explicit way to recover. Which is what we address in this work.

Safety measures in the context of a learned controller have been addressed in different forms [19]. One common thread is the identification of *safe* regions where the agent can operate and the use of a recovery policy. In [20], unknown regions are defined by the uncertainty estimate of a perception module. When the agent steps there, a model-based reset policy is triggered. Differently from our method, they require a model of the objects and they assume the transition function is known in a subset of the state space. Other works instead assume to directly have access to a *constraint function* from the environment or approximate it from data, which quantifies the safeness of states. A policy can then be learned to actively avoid such states [21, 22] or plan a trajectory that remains in the safe region of the state space [23–25]. Having access to the constraint function or data-points in unsafe states can, however, be problematic for robotic applications. A similar work to ours proposes to learn an approximation of the tangent space of the task manifold at any point [26]. This can, in turn, be used to plan the overall trajectory. If some kind of perturbation occurs, the agent can project its current position in the manifold and plan a path to get back in. The projection is learned explicitly using a dataset of perturbed points. On the contrary, the way we learn the encoder allows us to automatically get the projection direction even from high-dimensional states like images.

State representation for control has been widely studied in prior works [27–29]. Dividing the optimization of the representation from the policy has the advantage of easing the learning process and enables to constrain the policy formulation [30, 31]. Moreover, the learned representation can be re-used for different tasks assuming the underlying dynamics remain the same. In [32] and [33] an encoder is trained to compress the state representation while retaining all of the information. A transition model is then inferred on top of the representation to predict the dynamics of the environment. This formulation, however, produces a generic representation with no particular properties. To this end, more rigid constraints have been proposed. In [34] the representation is forced to evolve linearly in time, while in [30] the model outputs spatial features representing the observations. Moreover, by imposing a linear dynamic on the representation, the learned controller can be simplified and, under some assumptions, learned optimally [35, 36]. Unlike our method, this dynamic cannot be directly converted into an action. [37] train a variational autoencoder with the additional constraint of making the latent representation evolve according to Newtonian physics. This allows for classical controllers, like a PID, to be applied directly. In [38], they propose to learn an encoder and a transition model at the same time. Both models are learned such that the latent representation is equivariant to the transition model. Differently from all of these methods, we require our representation to be globally translational-equivariant in order to convert the gradient of the density estimator into a viable action.

F.3 Background

We study the problem of learning a policy using a Markov decision process (MDP). We assume the MDP to be fully observable and specified by the tuple (S, A, T). Here S is the set of observations representing the state of the environment, A is the set of actions the agent can take, and $T: S \times A \to S$ is the transition function governing the change in observations when the agent performs an action. We can then formulate the problem of policy training as learning a map $\pi: S \to A$ by minimizing a cost function. Throughout this paper we define the set of states S to be images and the set of actions $A \subseteq \mathbb{R}^n$ to be continuous translations of the agent's end-effector in the Euclidean space. A dataset $\mathcal{D}$ of experts' demonstrations is a collection of trajectories representing the robot completing a task in different conditions. This dataset can be considered as a collection of tuples $\mathcal{D} = \{(s, a, s')\}$ with $s' = T(s, a)$.

F.3.1 Behavioral Cloning

Behavioral Cloning is one of the most widely used imitation learning (IL) approaches due to its ease of implementation. It defines the cost function of the policy π as a supervised learning loss. As such, the policy's parameters θ can be inferred by minimizing the Mean Squared Error (MSE) between its estimated actions and the ones in the dataset $\mathcal{D}$:

$$\pi = \pi_{\theta^*}, \qquad \theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{(s,a)\in\mathcal{D}} \|\pi_\theta(s) - a\|^2. \tag{F.1}$$

This simple IL strategy offers a number of advantages. First, it is easy to implement and stable to train. Second, it can be learned completely from offline data requiring neither access to the environment nor any knowledge of the transition model or a reward function.

F.3.2 Equivariant Mapping

In this section, we explain *equivariance* as a property of a mapping E that in our case maps an input observation s into a latent representation $z \in Z$, i.e. $z = E(s)$. Here, we consider two transition functions, $T: S \times A \to S$ and $T': Z \times A \to Z$. The map $E: S \to Z$ is defined to be equivariant with respect to the transitions T, T' if [39, 40]:

$$E(T(s, a)) = T'(E(s), a). \tag{F.2}$$

Figure F.2 visually illustrates the equivariance property for the mapping E and the transitions T and T'. Intuitively, mapping the observation into the latent space followed by the transition T' must result in the same latent value z as applying the transition T followed by mapping the resulting observation into the latent space.

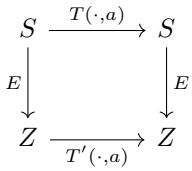


Figure F.2: Commutative diagram illustrating equivariance as a property of the mapping E.

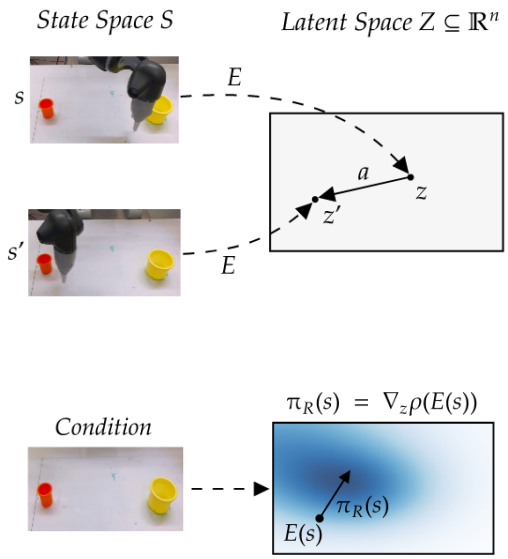


Figure F.3: Overview of the proposed method. **Top**: the equivariant model E maps the observation space (images) to the Euclidean latent space Z contained in the action space A. Actions of the agent correspond to translations in Z. **Bottom**: the latent density ρ is estimated conditioned on task-specific information. The recovery policy π_R follows the gradient of ρ and thus redirects the agent back to the training manifold.

F.4 Method

In this section, we introduce our method that augments a policy trained with BC with a recovery strategy which can bring the agent back *in-distribution*, where the BC policy performs optimally. We achieve this by introducing a *recovery policy* $\pi_R : S \to A$ whose actions move the agent to the training manifold, i.e., within the support of the observations in the training dataset. The intuition is that the agent should follow the recovery policy to recover from OOD states, and the BC policy when in-distribution to complete the task. Therefore, we propose to compute the

actions given by each policy, and find the weighted sum of the actions to obtain the final policy output. The weights are computed according to a normalized estimation of the density of the training states. However, as we describe in section F.4.2, we estimate the density by first mapping the observation into a latent space $z = E(s)$, and then computing the density given the latent variable z as the input $\rho(z)$. To ensure values in $(0, 1)$, the estimated density is normalized using a sigmoid function $\bar{\rho}(z) = 1/(1 + \exp(-(\rho(z) + \epsilon)/\tau))$ with an appropriate offset ϵ and temperature τ parameters, and then used as the following to compute the output of the augmented policy $\widetilde{\pi}$:

$$\widetilde{\pi}(s) = \bar{\rho}(z)\pi(s) + (1 - \bar{\rho}(z))\, \pi_R(s). \tag{F.3}$$

In the following sections, we first introduce our proposed equivariant mapping which maps raw visual observations into a latent space in which translation corresponds to the robot actions. Then, we introduce our method to estimate the density of the training observations in the latent space. Finally, we describe how the recovery policy is constructed based on the learned equivariant mapping.

F.4.1 Learning an Equivariant Mapping

We propose to learn a low-dimensional representation of the input visual observations by explicitly learning an equivariant mapping $E : S \to Z$. As we describe in section F.4.3, we exploit the equivariant property of the mapping to construct the recovery policy. Besides, learning a low-dimensional representation of visual inputs can also help in estimating the density of the training data.

Given the transition of the MDP $T(s, a)$, we consider the following transition in the latent space: $T'(z, a) = z + a$. This is because the robot actions *translate* the end-effector in the Euclidean space. We refer to Section F.6 for a discussion on how this can be generalized to actions beyond translations of an end-effector of a manipulator. We want to learn an equivariant mapping with respect to T, T' i.e., E has to satisfy the following version of Equation F.2:

$$E(T(s, a)) = E(s) + a. \tag{F.4}$$

As shown in Figure F.3 (left), E implements translational equivariance since it converts transitions into translations. Note that Equation F.4 assumes that A and Z share the same ambient space $\mathbb{R}^n$. The equivariant model E is trained on states and actions well-distributed within the environment i.e., a dataset $\mathcal{D}'$ is pre-collected independently from the specific task. As long as the scene composed of the extrinsic objects in the robot's environment remains constant, E can be deployed in different tasks. In order to ensure a correct representation, $\mathcal{D}'$ needs to be distributed as uniformly as possible. The mapping E is parameterized by a neural network $E = E_\varphi$ with output space $\mathbb{R}^n$ and optimized by minimizing the following objective function on the dataset $\mathcal{D}'$:

$$\varphi^* = \underset{\varphi}{\operatorname{argmin}} \sum_{(s,a,s')\in\mathcal{D}'} \|E_\varphi(s') - E_\varphi(s) - a\|^2. \tag{F.5}$$

F.4.2 Estimating the Density of the Training States

We estimate the probability density of the agent being within the support of the training data by learning a parametric *density estimator*. The density estimator needs to be conditioned on the position of the manipulation objects for the task. It is thus conditioned on the image observation for the initial configuration of the manipulation task (Figure F.3, right). We use *Mixture Density Networks* (MDN) [41], which estimate a conditional Gaussian mixture density. The MDN outputs the density in the form of means μ_i, (diagonal) co-variances σ_i and weights w_i of a mixture of Gaussians $\mathcal{N}(z;\ \mu_i, \sigma_i)$. The density in the point z can then be computed as follows:

$$\rho(z) = \sum_{i=1}^{N} w_i \mathcal{N}(z;\ \mu_i, \sigma_i). \tag{F.6}$$

The MDN is trained by minimizing the average negative log-likelihood of ρ over the observations in the training dataset $\mathcal{D}$.

F.4.3 Recovery Policy

The recovery policy is responsible to output actions that bring the agent closer to states within the support of the training data. This is done by finding an action that brings the agent into a state with higher estimated density. This is equivalent to performing gradient ascent on the estimated density function in the latent space. Because of the translational-equivariant property of our mapping, i.e., $z' = E(s') = E(s) + a = z + a$, the gradient $\nabla_z \rho(z)$ is equal to a robot action that moves the agent to higher density states. Therefore, we can simply define the output of the recovery policy for an observation s as:

$$\pi_R(s) = \eta \nabla_z\ \rho(E(s)) \tag{F.7}$$

where η is a scale parameter. Once the density of states has been estimated in the latent space, the recovery policy can be implemented accordingly without any further training phase.

F.5 Experiments

In order to assess the effectiveness of the recovery of the proposed model, we present the results of a number of experiments. First, the experimental setup is described, then details on each of the experiments are presented.

- The first experiment compares the performances of a BC agent with and without recovery on a robotic manipulation task.
- The second experiment tests the ability of a BC agent learned from noisy data, with and without recovery, in performing the same task.
- The third experiment compares the ability of BC, with and without recovery, in resuming a task if brought forcefully OOD.
- The fourth experiment involves solving a different task. The goal of this experiment is to show that the representation is agnostic to the task.

F.5.1 Experimental Setup

All the experiments are performed in the real world using a YuMi-IRB 14000 collaborative robot by ABB. We record all the data through teleoperation of the robot by a human. To implement the teleoperation system we used a virtual reality (VR) system connected to the robot's controllers. Teleoperation through VR has, in fact, proved to be a viable option for robotics applications thanks mainly to its ease in use and speed of data collection [42–45]. In particular, for our experiments, we interface an Oculus Quest 2 device to the robot's operating system.

To record the data, the human operator stands in front of the robot and operates the VR hand controller to command desired velocities to the end-effector of the robotic arm. Commands are thus mirrored with respect to the human perspective. Velocity commands are sent with a frequency of 10Hz to the robot that then translates them to the equivalent joint velocities. In all of the experiments, the velocities are just translations in space as no angular velocities are considered, meaning that $A \subseteq \mathbb{R}^3$. Images, representing the state of the system by a static camera placed in front of the robot in coordination with the VR commands.

The setup is the same across all experiments and its shown in Figure F.4. The robot is placed in front of a table where two objects are placed, a small plastic orange cylinder, the *manipulated object*, and a bigger plastic yellow cylinder, the *target object*. Throughout all the experiments the target object is never moved while the robot needs to interact with the manipulated object.

F.5.2 Networks' Architectures

The equivariant encoder E is parameterized by 5 convolutional layers with 64 channels except the last one with 8 followed by a hidden fully-connected layer with 64 units, every hidden layer is followed by a ReLU activation function. The network is trained with a learning rate of 10^{-3} using the Adam optimizer. Input images are cropped, resized and normalized before being fed to the network.

The policy is parameterized by a neural network with a ResNet18 backbone pre-trained on ImageNet and a randomly initialized fully connected head with 2 hidden layers of 64 units each. The MDN density estimator is parameterized by a CNN model of 4 layers with 64 channels and one with 8 channels followed by 3 output layers for the mean, diagonal variance and weights of the mixture of Gaussians. The MDN is trained, as stated in Section F.4.2, to minimize the negative log-likelihood of the latent representation of the equivariant encoder. The MDN is conditioned on the initial image of each trajectory and the gripper state. The reason being that the model needs to be aware of the position of the object to be picked up to output a conditional density but it should not have access to the position of the gripper. There is, in fact, a functional dependency between images where the gripper is shown and the density estimate itself. By conditioning the MDN on the current image during the roll-out, the density would collapse. Conditioning the model on the gripper state is also needed as the position of the gripper should be considered in-distribution or not based on the object being grasped or not, respectively. Training an MDN is notoriously unstable [46] and we found that adding a deconvolution decoder that maps a middle representation of the model into a reconstruction of the original image can stabilize the training. Both the policy and the MDN are trained using the Adam optimizer with a learning rate of 10^{-4}. The other hyper-parameters used for this experiment are the following: $\eta = 0.05$, $\epsilon = 2.0$ and $\tau = 0.5$.

F.5.3 Equivariant Encoder

The equivariant representation used for the density estimator is agnostic to the task and can be learned a priori. For the dataset $\mathcal{D}'$, we collect 6 trajectories of the robotic arm moving uniformly in the space and interacting with the manipulated object for a total of 1741 steps. Interaction with the objects is needed in order to make the learned encoder invariant to its position. Equivariance is defined with respect to the robot's actions only and the learned representation should not be sensitive to the object's position.

Because the actions are translations in the Euclidean 3D space the equivariant representation will correspond to points in 3D where the gripper of the robot is located (the central point of the actual movement). The representation preserves distances so the scale is one to one with the actions' magnitude. This can be seen in Figure F.4 where the expert's trajectories have been mapped into the latent space of the encoder and projected in 2D. In fact, the shape of the table and the relative position of the objects are maintained. For interpretability, we force the initial configuration of the robot to be the origin of the representation by including a second term in the loss of Equation F.5.

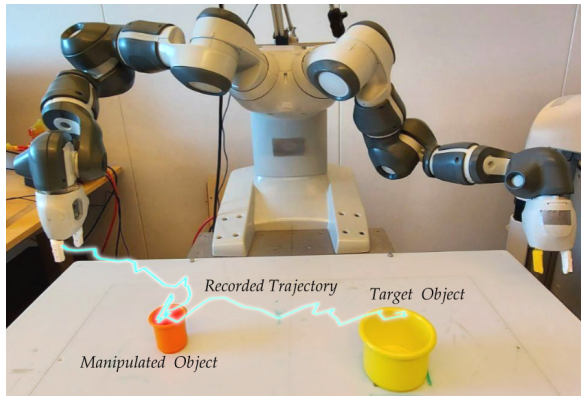


Figure F.4: Experimental setup for the pick-and-drop task. The small bin (*manipulated object*) has to be dropped inside the larger one (*target object*). A trajectory recorded by the expert via teleoperation is also displayed.

F.5.4 Pick-and-Drop Experiment

Experiment description

In the first experiment, the goal is for the robot to pick the manipulated object, move it on top of the target object and drop it inside. Here the actions are the combination of the gripper's velocity and a binary value representing the state of the gripper (either open or close). The initial position of the manipulated object is initialized randomly in the first half of the table while the target object is kept fixed on the other half of the table. A dataset of 120 trajectories of a human demonstrating the task is used to train the model. The dataset accounts for 5526 steps in total and is used to train both the behavioral cloning policy and the density estimator. The demonstrations are collected using the VR teleoperation system described above. However, these demonstrations cannot be assumed optimal due to the noise in the teleoperation system and the non-Markovianity of the environment. In fact, there are two elements here that break the Markovian assumption. The first is the current velocity and acceleration the robot has before giving it a command. The next state will vary depending on these properties that are not inferable from one single image. The second element is the inverse kinematic module of the robot. The resulting actual displacement of the end effector will also depend on the current state of the joints that is not fully observable from the images.

Results

In the first set of experiments, we test the proposed method against the imitation learning policy without the recovery term. The models are evaluated on 20 trials with the manipulated object in different positions. Performances are based on their ability to successfully grasp the manipulated object and their ability to then drop it

Table F.1: Results of the pick-and-drop task for a standard behavioral cloning policy with and without the proposed recovery. The models are compared in their ability of picking the manipulated object correctly and dropping it inside the target object. The table shows results on models learned on correct demonstrations as well as demonstrations with actions that are shifted by one time step with respect to the corresponding images. Models are also tested on their ability to overcome perturbations while performing the task.

Model	Grasp	Drop
Pick-and-Drop		
BC	25%	25%
BC with Recovery	70%	55%
Shifted Actions Pick-and-Drop		
BC	0%	0%
BC with Recovery	70%	50%
Perturbed Pick-and-Drop		
BC	30%	10%
BC with Recovery	100%	70%

inside the target bin. Table F.1 shows the results of this experiment. The standard BC model manages to complete the task only one-fourth of the time. On the other hand, coupling the same policy with the proposed recovery lets the agent adjust its position every time it makes a mistake that would bring it OOD. This results in a much higher success rate.

Further, to assess the robustness of the recovery policy, we test it in noisy conditions. We train a second BC policy on the same dataset but with all the actions shifted by one step with respect to the images. We effectively simulate a data gathering scenario where there is a delay between the camera sensor and the actual movement of the robot. Two subsequent states are not connected by the saved action. However, because of the uniformity of the task, the real action does not differ considerably and the overall motion of the agent keeps a similar behavior. Nonetheless, a policy trained on this sub-optimal dataset does not manage to solve the task even once. On the other hand, by coupling the same policy with the proposed recovery module the performances are quite unchanged with respect to the correct dataset case, see Table F.1.

Lastly, we test the BC policy with and without recovery by applying a random displacement to the robot. At the beginning of the task, we move the gripper in a random direction and let it continue the task from there. The new position could

Table F.2: Results of the pushing task for standard behavioral cloning policy with and without the proposed recovery. The models are compared in their ability of inserting the tip of the gripper within the manipulated object correctly and pushing it towards the target.

Push	
Model	Complete
BC	20%
BC with Recovery	25%

be outside of the training distribution, making the imitation learning policy behave randomly. The recovery policy instead can climb back to the training manifold and continue with the task normally. The object is initialized in a position where the imitation learning agent is able to complete the task. As shown in Table F.1, the learned policy suffers considerably from this kind of perturbations. On the other hand, when coupled with the proposed recovery it can always get back in and most of the time complete the task.

F.5.5 Pushing Experiment

The second set of experiments involves the same environment with unchanged dynamics. The goal is for the agent to insert the gripper's tip into the manipulated object and push it all the way towards the target object. In this experiment, the gripper is always closed. Because the robotic arm moves in the same way and only the manipulated object is moved throughout the roll-outs, the encoder can be used without retraining. Both the imitation learning policy and the density estimator have to be retrained on the new demonstrations. For this experiment, a new dataset of 60 demonstrations is collected for a total of 3895 steps. Results in Table F.2 show that the encoder can be used without retraining and that the recovery policy increases the performances of the learned policy.

F.6 Conclusions and Future Work

We proposed to couple an agent learned on experts' demonstrations with a recovery policy to keep it within the training data. This is achieved by explicitly modeling the training distribution with a density estimator and bypassing the agent's action whenever the current state is detected to be OOD. By training an encoder to be equivariant to the agent's actions, the recovery policy can be formulated as a form of gradient ascent on the density estimate.

We applied the proposed methodology to a robotic manipulator whose actions correspond to Euclidean translations. As a possible extension, more complex actions could be considered such as rotations of joins and end-effectors. This would involve *Lie groups* beyond the Euclidean space such as the group of rotations $\mathrm{SO}(n)$, $n = 2, 3$. A further extension of the framework lies in designing more complex recovery strategies than pure gradient ascent in order to smooth the resulting movement.

F.7 Acknowledgements

This work has been supported by the Swedish Research Council, Knut and Alice Wallenberg Foundation, European Research Council (BIRD-884807) and H2020 CANOPIES.

References

[1] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[2] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[3] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[4] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, JMLR Workshop and Conference Proceedings, 2011.

[5] W. Zhou, S. Bajracharya, and D. Held, "Plas: Latent action space for offline reinforcement learning," *arXiv preprint arXiv:2011.07213*, 2020.

[6] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.

[7] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, "Combo: Conservative offline model-based policy optimization," *arXiv preprint arXiv:2102.08363*, 2021.

[8] A. Ghadirzadeh, J. Bütepage, A. Maki, D. Kragic, and M. Björkman, "A sensorimotor reinforcement learning framework for physical human-robot interaction," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2682–2688, IEEE, 2016.

[9] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine, "Uncertainty-aware reinforcement learning for collision avoidance," *arXiv preprint arXiv:1702.01182*, 2017.

[10] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *arXiv preprint arXiv:1811.06711*, 2018.

[11] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 661–668, 2010.

[12] P. Abbeel and A. Y. Ng, "Inverse reinforcement learning.," 2010.

[13] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[14] D. Precup, "Eligibility traces for off-policy policy evaluation," *Computer Science Department Faculty Publication Series*, p. 80, 2000.

[15] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing off-policy q-learning via bootstrapping error reduction," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *International Conference on Machine Learning*, pp. 2052–2062, PMLR, 2019.

[17] X. Chen, A. Ghadirzadeh, T. Yu, Y. Gao, J. Wang, W. Li, B. Liang, C. Finn, and C. Zhang, "Latent-variable advantage-weighted policy optimization for offline rl," *arXiv preprint arXiv:2203.08949*, 2022.

[18] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma, "Mopo: Model-based offline policy optimization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14129–14142, 2020.

[19] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, 2021.

[20] M. A. Lee, C. Florensa, J. Tremblay, N. Ratliff, A. Garg, F. Ramos, and D. Fox, "Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7505–7512, IEEE, 2020.

[21] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, "Recovery rl: Safe reinforcement learning with learned recovery zones," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4915–4922, 2021.

[22] K. Srinivasan, B. Eysenbach, S. Ha, J. Tan, and C. Finn, "Learning to be safe: Deep rl with a safety critic," *arXiv preprint arXiv:2010.14603*, 2020.

[23] B. Thananjeyan, A. Balakrishna, U. Rosolia, F. Li, R. McAllister, J. E. Gonzalez, S. Levine, F. Borrelli, and K. Goldberg, "Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3612–3619, 2020.

[24] A. Wilcox, A. Balakrishna, B. Thananjeyan, J. E. Gonzalez, and K. Goldberg, "Ls3: Latent space safe sets for long-horizon visuomotor control of sparse reward iterative tasks," in *Conference on Robot Learning*, pp. 959–969, PMLR, 2022.

[25] I. Mitsioni, P. Tajvar, D. Kragic, J. Tumova, and C. Pek, "Safe data-driven contact-rich manipulation," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pp. 120–127, IEEE, 2021.

[26] M. Li, K. Tahara, and A. Billard, "Learning task manifolds for constrained object manipulation," *Autonomous Robots*, vol. 42, no. 1, pp. 159–174, 2018.

[27] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat, "State representation learning for control: An overview," *Neural Networks*, vol. 108, pp. 379–392, 2018.

[28] X. Chen, A. Ghadirzadeh, M. Björkman, and P. Jensfelt, "Adversarial feature training for generalizable robotic visuomotor control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1142–1148, IEEE, 2020.

[29] A. Hämäläinen, K. Arndt, A. Ghadirzadeh, and V. Kyrki, "Affordance learning for end-to-end visuomotor robot control," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1781–1788, IEEE, 2019.

[30] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 512–519, IEEE, 2016.

[31] A. Ghadirzadeh, A. Maki, D. Kragic, and M. Björkman, "Deep predictive policy training using reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2351–2358, IEEE, 2017.

[32] J.-A. M. Assael, N. Wahlström, T. B. Schön, and M. P. Deisenroth, "Data-efficient learning of feedback policies from image pixels using deep dynamical models," *arXiv preprint arXiv:1510.02173*, 2015.

[33] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[34] R. Goroshin, M. F. Mathieu, and Y. LeCun, "Learning to linearize under uncertainty," *Advances in neural information processing systems*, vol. 28, 2015.

[35] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," *Advances in neural information processing systems*, vol. 28, 2015.

[36] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. Johnson, and S. Levine, "Solar: Deep structured representations for model-based reinforcement learning," in *International Conference on Machine Learning*, pp. 7444–7453, PMLR, 2019.

[37] M. Jaques, M. Burke, and T. M. Hospedales, "Newtonianvae: Proportional control and goal identification from pixels via physical latent spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4454–4463, 2021.

[38] T. Kipf, E. van der Pol, and M. Welling, "Contrastive learning of structured world models," *arXiv preprint arXiv:1911.12247*, 2019.

[39] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *International conference on machine learning*, pp. 2990–2999, PMLR, 2016.

[40] G. L. Marchetti, G. Tegnér, A. Varava, and D. Kragic, "Equivariant representation learning via class-pose decomposition," *arXiv preprint arXiv:2207.03116*, 2022.

[41] C. M. Bishop, "Mixture density networks," 1994.

[42] N. Koganti, A. Rahman HAG, Y. Iwasawa, K. Nakayama, and Y. Matsuo, "Virtual reality as a user-friendly interface for learning from demonstrations," in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–4, 2018.

[43] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

[44] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5628–5635, IEEE, 2018.

[45] D. Whitney, E. Rosen, D. Ullman, E. Phillips, and S. Tellex, "Ros reality: A virtual reality framework using consumer-grade hardware for ros-enabled robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, IEEE, 2018.

[46] O. Makansi, E. Ilg, O. Cicek, and T. Brox, "Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7144–7153, 2019.

Paper G

Harmonics of Learning: Universal Fourier Features Emerge in Invariant Networks

Giovanni Luca Marchetti, Christopher Hillar, Danica Kragic, Sophia Sanborn

Abstract

In this work, we formally prove that, under certain conditions, if a neural network is invariant to a finite group then its weights recover the Fourier transform on that group. This provides a mathematical explanation for the emergence of Fourier features – a ubiquitous phenomenon in both biological and artificial learning systems. The results hold even for non-commutative groups, in which case the Fourier transform encodes all the irreducible unitary group representations. Our findings have consequences for the problem of symmetry discovery. Specifically, we demonstrate that the algebraic structure of an unknown group can be recovered from the weights of a network that is at least approximately invariant within certain bounds. Overall, this work contributes to a foundation for an algebraic learning theory of invariant neural network representations.

G.1 Introduction and Related Work

Artificial neural networks trained on natural data exhibit a striking phenomenon: regardless of exact initialization, dataset, or training objective, models trained on the same data domain frequently converge to similar learned representations [2]. For example, the early layer weights of diverse image models tend to converge to

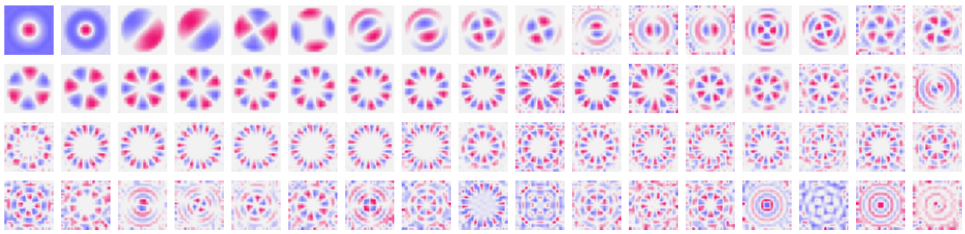


Figure G.1: Emergent Circular Harmonics. Weights learned by a neural network trained for invariance to planar rotations resemble circular harmonics. Data from [1].

Gabor filters and color-contrast detectors [3]. Remarkably, many of these same features are observed in the visual cortex [4–6], suggesting a form of representational *universality* that transcends biological and artificial substrates. While such findings are empirically well-established in the *mechanistic interpretability* literature [7], the field lacks theoretical explanations.

Spatially localized versions of canonical 2D Fourier basis functions, such as Gabor filters or wavelets, are perhaps the most frequently observed universal features in image models. They commonly arise in the early layers of vision models – trained with efficient coding [8, 9], classification [3], temporal coherence [10], and next-step prediction [11] objectives – as well as in the primary visual cortices of diverse mammals – including cats [12], monkeys [13], and mice [14]. Non-localized Fourier features have been observed in networks trained to solve tasks that permit cyclic wraparound – for example, modular arithmetic [15], more general group compositions [16], or invariance to the group of cyclic translations [1]. In the domain of spatial navigation, the so-called *grid cells* of the entorhinal cortex [17] display periodic firing patterns at different spatial frequencies as they build a map of space. Their response properties are naturally modeled with the harmonics of the twisted torus [18–20]. Similar features also emerge in artificial neural networks trained to solve spatial navigation tasks [21, 22]. The ubiquity of these features across diverse learning systems is both striking and unexplained.

In this work, we provide a mathematical explanation for the emergence of Fourier features in learning systems such as neural networks. We argue that the mechanism responsible for this emergence is the downstream *invariance* of the learner to the action of a *group of symmetries* (e.g. planar translation or rotation). Since natural data typically possess symmetries, invariance is a fundamental bias that is injected both implicitly and sometimes explicitly into learning systems [23]. Motivated by this, we derive theoretical guarantees for the presence of Fourier features in invariant learners that apply to a broad class of machine learning models.

Our results rely on the inextricable link between harmonic analysis and group theory [24]. The standard discrete Fourier transform is a special case of more general Fourier transforms on groups, which can be defined by replacing the standard basis of harmonics by irreducible unitary group representations. The latter are equivalent to the familiar definition for cyclic or, more generally, commutative groups, but are more involved for non-commutative ones. In order to accommodate both the scenarios, we develop a general theory that applies, in principle, to arbitrary finite groups.

This work represents an attempt to provide mathematical grounding for a general algebraic theory of representation learning, while addressing the *universality hypothesis* for neural networks [3, 25]. A suite of earlier theoretical works [26–28] established such universality for sparse coding models [8], deriving the conditions under which a network will recover the original bases that generate data through sparse linear combinations. In this case, the statistics of the data determine the uniqueness of the representation. Our findings, on the other hand, are purely *algebraic*, since they rely exclusively on the invariance properties of the learner. Given the centrality of invariance to many machine learning tasks, our theory encompasses a broad class of scenarios and neural network architectures, while providing a new perspective on classical neuroscience [12, 29]. As such, it sets a foundation for a learning theory of representations in artificial and biological neural systems, grounded in the mathematics of symmetry.

G.1.1 Overview of Results

In this section, we provide a non-technical overview of the theoretical results presented in this work. Our main result can be summarized as follows.

Informal Theorem G.1.1 (Theorem G.3.1 and Corollary G.3.2)**.** *If $\varphi(W, x)$ is a parametric function of a certain kind that is invariant in the input variable x to the action of a finite group G, then each component of its weights W coincides with a harmonic of G up to a linear transformation. In particular, when the weights are orthonormal, W coincides with the Fourier transform of G up to linear transformations.*

In the above, the term 'harmonic' refers to an irreducible unitary representation of G. Indeed, one-dimensional unitary representations correspond to homomorphisms with the unit circle $\mathrm{U}(1) \subseteq \mathbb{C}$, which is reminiscent of the classical definition via the imaginary exponential. However, *non-commutative* groups can have higher-dimensional irreducible representations, intuitively meaning that harmonics are valued in unitary matrices. In this case, the components of W can be interpreted as *capsules* in the sense of [30] i.e., neural units processing matrix-valued signals.

We show that the hypothesis on φ in Theorem G.1.1 is satisfied by several machine learning models from the literature. In particular, the theorem applies to the recently-introduced (Bi)Spectral Networks [1], to single fully-connected layers of McCulloch-Pitts neurons, and, to an extent, to traditional deep networks. Since harmonic analysis is naturally formalized over the complex numbers, we consider models with complex weights W, which fits into the larger program of complex-valued machine learning [31–33].

The group-theoretical Fourier transform encodes the entire group structure of G. Therefore, as a consequence of Theorem G.1.1 the multiplication table of G can be recovered from the weights W of an invariant parametric function φ – a fact empirically demonstrated in [1]. This addresses the question of *symmetry discovery* – an established machine learning problem aiming to recover the unknown group of symmetries of data with minimal supervision and prior knowledge [34–36].

Since the multiplication table is a discrete object, it is expected that the invariance constraint on φ can be loosened while still recovering the group correctly. To this end, we prove the following.

Informal Theorem G.1.2 (Theorem G.3.6)**.** *If $\varphi(W, x)$ is 'almost invariant' to G according to certain functional bounds and the weights are 'almost orthonormal', then the multiplicative table of G can be recovered from W.*

Lastly, we implement a model satisfying the requirements of our theory and demonstrate its symmetry discovery capabilities. To this end, we train it via contrastive learning on an objective encouraging invariance and extract the multiplicative table of G from its weights. Our Python implementation is available at a public repository [1].

G.2 Mathematical Background

We begin by introducing the fundamental concepts from harmonic analysis and group theory used in this paper. For a complete treatment, we refer the reader to [24].

G.2.1 Groups and Actions

A *group* is an algebraic object whose elements represent abstract symmetries, which can be composed and inverted.

Definition G.2.1. A group is a set G equipped with a *composition map* $G \times G \to G$ denoted by $(g, h) \mapsto gh$, an *inversion map* $G \to G$ denoted by $g \mapsto g^{-1}$, and a distinguished *identity element* $1 \in G$ such that for all $g, h, k \in G$:

[1] `https://github.com/sophiaas/spectral-universality`

Associativity	*Inversion*	*Identity*
$g(hk) = (gh)k$	$g^{-1}g = gg^{-1} = 1$	$g1 = 1g = g$

A map $\rho : G \to G'$ between groups is called a *homomorphism* if $\rho(gh) = \rho(g)\rho(h)$ for all $g, h \in G$.

Examples of groups include the permutations of a set and the general linear group $\mathrm{GL}(V)$ of invertible operators over a vector space V, both equipped with the usual composition and inversion of functions. A further example that will be relevant in this work is the *unitary* group $\mathrm{U}(V) \subseteq \mathrm{GL}(V)$ associated to a Hilbert space V, consisting of operators U satisfying $UU^\dagger = I$, where $\dagger$ denotes the conjugate transpose and I is the identity matrix. Groups satisfying $gh = hg$ for all $g, h \in G$ are deemed *commutative*.

The idea of a space $\mathcal{X}$ having G as a group of symmetries is abstracted by the notion of group *action*.

Definition G.2.2. An action by a group G on a set $\mathcal{X}$ is a map $G \times \mathcal{X} \to \mathcal{X}$ denoted by $(g, x) \mapsto g \cdot x$, satisfying for all $g, h \in G,\ x \in \mathcal{X}$:

Associativity	*Identity*
$g \cdot (h \cdot x) = (gh) \cdot x$	$1 \cdot x = x$

A map $\varphi\colon\ \mathcal{X} \to \mathcal{Z}$ between sets acted upon by G is called *equivariant* if $\varphi(g \cdot x) = g \cdot \varphi(x)$ for all $g \in G, x \in \mathcal{X}$. It is called *invariant* if moreover G acts trivially on $\mathcal{Z}$ or, explicitly, if $\varphi(g \cdot x) = \varphi(x)$.

In general, the following actions can be defined for arbitrary groups: G acts on any set *trivially* by $g \cdot x = x$, and G acts on itself seen as a set via (left) *multiplication* by $g \cdot h = gh$. Further examples are $\mathrm{GL}(V)$ and $\mathrm{U}(V)$ acting on V by evaluating operators.

G.2.2 Harmonic Analysis on Groups

Harmonic analysis on groups [37] generalizes standard harmonic analysis. We focus here on finite groups for simplicity, which are sufficient for practical applications. This avoids technicalities such as integrability conditions and continuity issues arising for infinite groups. We start by considering commutative groups and cover non-commutative ones in Section G.2.3.

Let G be a finite commutative group of order $|G|$. Denote by $\langle G \rangle = \mathbb{C}^G$ the free complex vector space generated by G. Intuitively, an element $x = (x_g)_{g \in G} \in \langle G \rangle$ represents a complex-valued signal over G. The space $\langle G \rangle$ is endowed with the *convolution* product,

$$(x \star y)_g = \sum_{h \in G} x_h y_{h^{-1}g}, \tag{G.1}$$

and is acted upon by G via $g \cdot x = \delta_g \star x = (x_{g^{-1}h})_{h \in G}$, where δ_g is the canonical basis vector.

Definition G.2.3. The *dual* $G^\vee$ of G is the set of homomorphisms $\rho : G \to \mathrm{U}(1)$, where $\mathrm{U}(1) \subseteq \mathbb{C}$ is the group of unitary complex numbers equipped with multiplication. It is itself a group when equipped with pointwise composition $(\rho\mu)(g) = \rho(g)\mu(g)$.

A homomorphism $\rho \in G^\vee$ intuitively represents a *harmonic* over G, generalizing the familiar notion from signal processing. If we endow $\langle G \rangle$ with the canonical scalar product $\langle x, y \rangle = \sum_{g \in G} \overline{x_g} y_g$, then $G^\vee \subseteq \langle G \rangle$ forms an orthogonal basis with all the norms equal to $|G|$. The linear base-change is, by definition, the Fourier transform over $\langle G \rangle$:

Definition G.2.4. The *Fourier transform* is the map $\langle G \rangle \to \langle G^\vee \rangle$, $x \mapsto \hat{x}$, defined for $\rho \in G^\vee$ as:

$$\hat{x}_\rho = \langle \rho, x \rangle. \tag{G.2}$$

The Fourier transform is a linear isometry or, equivalently, a unitary operator, up to a multiplicative constant of $|G|$. Moreover, it exchanges the convolution product $\star$ over $\langle G \rangle$ with the Hadamard product $\odot$ over $\langle G^\vee \rangle$. Definition G.2.4 generalizes the usual discrete Fourier transform in the following sense. For an integer $d > 0$, consider the cyclc group C_d with d elements. Concretely $G = \mathbb{Z}/d\mathbb{Z} \simeq C_d$ is the group of integers modulo d equipped with addition as composition. The dual $G^\vee$ consists of homomorphisms of the form:

$$\mathbb{Z}/d\mathbb{Z} \ni g \mapsto e^{2\pi\sqrt{-1}gk/d}, \tag{G.3}$$

for $k \in \{0, \cdots, d-1\}$. Equation G.2 specializes then to the familiar definition of the Fourier transform.

G.2.3 Non-Commutative Harmonic Analysis

So far, we have assumed that G is commutative. In this section we briefly discuss the extension of Fourier theory to non-commutative groups. This however requires more elaborate theoretical tools, which we now introduce. To begin with, in order to perform harmonic analysis on general groups it is necessary to discuss unitary representations. The latter will play the role of *matrix-valued harmonics*.

Definition G.2.5. A *unitary representation* of G is an action by G on a finite-dimensional Hilbert space V via unitary operators or, in other words, a homomorphism $\rho_V : G \to \mathrm{U}(V)$. A unitary representation is said to be *irreducible* if V does not contain any non-trivial[2] sub-representations.

[2]The trivial sub-representations of V are 0 and V.

We denote by $\mathrm{Irr}(G)$ the set of all irreducible representations of G up to isomorphism. Moreover, for a vector space V we denote by $\mathrm{End}(V)$ the space of its linear operators.

Definition G.2.6. The *Fourier transform* is the map $\langle G\rangle \to \bigoplus_{\rho_V \in \mathrm{Irr}(G)} \mathrm{End}(V)$, $x \mapsto \hat{x}$, defined for $\rho_V \in \mathrm{Irr}(G)$ as:

$$\hat{x}_{\rho_V} = \sum_{g\in G} \rho_V(g)^\dagger x_g \in \mathrm{End}(V). \tag{G.4}$$

This generalizes Definition G.2.4 since for a commutative group, ρ_V is irreducible if, and only if, $\dim(V) = 1$. Analogously to the commutative setting, the Fourier transform exchanges the convolution product $\star$ with the point-wise operator composition, which we still denote by $\odot$. Moreover, the Fourier transform is a unitary operator up to a multiplicative constant of $|G|$ with respect to the normalized Hilbert-Schmidt scalar product on $\mathrm{End}(V)$, given by:

$$\langle A, B\rangle = \dim(V)\ \mathrm{tr}(A^\dagger B). \tag{G.5}$$

The norm associated to the Hilbert-Schmidt scalar product is the Frobenius norm. The relations between irreducible unitary representations coming from the unitarity of the Fourier transform are known as *Schur orthogonality* relations.

G.3 Theoretical Results

We now present the primary theoretical contributions of this work. Concretely, we demonstrate that if certain parametric functions are invariant to a group then their weights must almost coincide with harmonics, i.e. irreducible unitary group representations. We start by introducing general algebraic notions and principles, and then proceed to specialize them to machine learning scenarios.

Let G be a finite group, $\mathcal{H}$ be a set, and $V_1, \cdots, V_k$ be complex finite-dimensional Hilbert spaces. In what follows, we will consider the space,

$$\mathcal{W} = \langle G\rangle \otimes \bigoplus_i \mathrm{End}(V_i) \simeq \bigoplus_i \mathrm{End}(V_i)^{\oplus G}. \tag{G.6}$$

$\mathcal{W}$ is a Hilbert space when endowed with the scalar product given by the product of the canonical scalar product over $\langle G\rangle$ and the normalized Hilbert-Schmidt scalar products over $\mathrm{End}(V_i)$ (see Equation G.5). For $W \in \mathcal{W}$, we will denote each of its components as $W_i = (W_i(g))_g \in \mathrm{End}(V_i)^{\oplus G}$. Moreover, we will often interpret elements $W \in \mathcal{W}$ as linear maps $\langle G\rangle \to \bigoplus_i \mathrm{End}(V_i)$ via $W(x) = \sum_{g\in G} W(g)x_g$ for $x \in \langle G\rangle$, where $W(g) = (W_i(g))_i$. Note that G acts on the left tensor factor of $\mathcal{W}$ while for each i, $\mathrm{U}(V_i)$ acts on the right tensor factor of $\langle G\rangle \otimes \mathrm{End}(V_i)$ by composition of operators.

Definition G.3.1. We say that a map $\varphi\colon \mathcal{W} \to \mathcal{H}$ has *unitary symmetries* if for $W, W' \in \mathcal{W}$ of the same norm, $\varphi(W) = \varphi(W')$ implies that for each i there exists a unitary operator $U_i \in \mathrm{U}(V_i)$ such that $W_i = U_i \cdot W_i'$.

In the context of machine learning, $\mathcal{H}$ will represent the *hypothesis space*, consisting of functions the model can learn. On the other hand, φ will represent the parametrization of such hypotheses, with its domain $\mathcal{W}$ being the space of weights. The component $\langle G\rangle$ of $\mathcal{W}$ will be responsible for parametrizing the input space, while each component $\mathrm{End}(V_i)$ will represent a computational unit, i.e. a complex-valued *neuron* in the language of neural networks. For commutative groups, we simply have $V_i = \mathbb{C} \simeq \mathrm{End}(V_i)$. In general, $\mathrm{End}(V_i)$ can be thought of as parametrizing matrix-valued signals, which, as mentioned in Section G.1.1, are computed by neural units sometimes referred to as *capsules* [30].

The following is an abstract algebraic principle at the core of this work.

Theorem G.3.1. *Suppose that $\varphi\colon \mathcal{W} \to \mathcal{H}$ has unitary symmetries and that for some $W \in \mathcal{W}$ the following holds:*

- *$\varphi(g \cdot W) = \varphi(W)$ for all $g \in G$.*
- *W, seen as a linear map $\langle G\rangle \to \bigoplus_i \mathrm{End}(V_i)$, is surjective.*

Then for every i there exist $W_i' \in \mathrm{End}(V_i)$ and an irreducible unitary representation $\rho_i : G \to \mathrm{U}(V_i)$ such that for all $g \in G$,

$$W_i(g) = W_i' \rho_i(g)^\dagger. \tag{G.7}$$

Proof. Since φ has unitary symmetries and $\|g \cdot W\| = \|W\|$, it follows that for every $g \in G$ and every i there exists $\rho_i(g) \in \mathrm{U}(V_i)$ such that

$$g \cdot W_i = \rho_i(g) \cdot W_i. \tag{G.8}$$

In particular, by considering the component with index $1 \in G$ on both sides of Equation G.8, we see that $W_i(g^{-1}) = W_i(1)\ \rho_i(g)$. We wish to show that ρ_i is a homomorphism. To this end, for all $g, h \in G$ it holds that

$$\rho_i(gh) \cdot W_i = (gh) \cdot W_i = g \cdot (\rho_i(h) \cdot W_i) = \rho_i(h) \cdot (g \cdot W_i) = (\rho(g)\rho(h)) \cdot W_i. \tag{G.9}$$

By the surjectivity hypothesis, the set $\{W_i(g)\}_{g\in G}$ generates $\mathrm{End}(V_i)$ as a vector space. Therefore, Equation G.9 implies that $\rho_i(gh) = \rho_i(g)\rho_i(h)$, as desired. Lastly, note that ρ_i is irreducible by the sujectivity assumption. Indeed, a nontrivial linear subspace of V_i fixed by $\rho_i(g)$ for all g would be sent by $W_i(g)$ into a fixed proper subspace due to Equation G.8. This contradicts the surjectivity of W_i. □

Note that the surjectivity assumption implies the constraint $\sum_i \dim(V_i)^2 \leq |G|$. As a consequence of the result above, the full Fourier transform arises with an additional orthogonality assumption.

Corollary G.3.2. *Suppose that $\varphi\colon \mathcal{W} \to \mathcal{H}$ has unitary symmetries and that for some $W \in \mathcal{W}$ the following holds:*

- *$\varphi(g \cdot W) = \varphi(W)$ for all $g \in G$.*
- *W is unitary up to a multiplicative constant, i.e. $W^\dagger W = |G| I$.*

Then W is the Fourier transform up to composing each of the components W_i by an operator with Frobenius norm equal to 1.

Proof. Note that the unitarity assumption above implies the surjectivity assumption from Theorem G.3.1. Therefore, it follows that for every i, there exists an irreducible unitary representation $\rho_i : G \to \mathrm{U}(V_i)$ such that $W_i(g) = W_i(1)\ \rho_i(g)^\dagger$. We wish to show that if $i \neq j$ then ρ_i and ρ_j are non-isomorphic representations. If not, the orthogonality assumption implies that

$$0 = \sum_{g\in G} \overline{W_i(g)} \otimes W_j(g) = \sum_{g\in G} \left(\overline{W_i(1)} \otimes W_j(1)\right) \left(\rho_i(g)^\top \otimes \rho_j(g)^\dagger\right) = \tag{G.10}$$

$$= |G|\ \overline{W_i(1)} \otimes W_j(1), \tag{G.11}$$

where $\top$ denotes the transpose and where the last identity follows from the Schur orthogonality relations. But then $W_i(1)$ or $W_j(1)$ is vanishing, which contraddicts the unitarity assumption.

Lastly, in order to compute the Frobenius norm of $W_i(1)$, note that

$$|G|\ \|W_i(1)\|^2 = \sum_{g\in G} \|W_i(g)\ \rho_i(g)\|^2 = \sum_{g\in G} \|W_i(g)\|^2 = |G|, \tag{G.12}$$

from which $\|W_i(1)\| = 1$ follows. □

Again, the orthogonality assumption implies that $V_1, \cdots, V_k$ are the ambient Hilbert spaces of all the irreducible unitary representations of G up to isomorphism, and in particular $\sum_i \dim(V_i)^2 = |G|$.

We now wish to discuss the other crucial assumption of Theorem G.3.1 requiring that $\varphi(g \cdot W) = \varphi(W)$ for all $g \in G$, which is reminiscent of invariance. However, when $\mathcal{H}$ is a space of functions, we are typically interested in models that are invariant in the input variable rather than the weight variable. Therefore, we introduce the following condition, aimed at reconciling inputs and weights. To this end, suppose that $\mathcal{H}$ is a set of functions $\mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is a set acted upon by G and $\mathcal{Y}$ is a set. We adhere to the notation $\varphi(W, x) = \varphi(W)(x)$, $x \in \langle G\rangle$, $W \in \mathcal{W}$, for simplicity.

Definition G.3.2. We say that φ satisfies the *adjunction property* if

$$\varphi(W, g \cdot x) = \varphi(g^{-1} \cdot W, x) \tag{G.13}$$

for all $x \in \mathcal{X}, g \in G$.

The adjunction property implies that if $\varphi(W, x)$ is invariant in x, then $\varphi(g \cdot W, x) = \varphi(W, x)$ for all x, g, recovering the assumption of Theorem G.3.1.

Remark G.3.1. As explained above, the tensor component $\langle G \rangle$ of $\mathcal{W}$ typically represents the input space of a given machine learning model. One can consider the more general scenario when data consist of complex signals over a finite set $\mathcal{S}$ acted upon by G, therefore replacing $\langle G \rangle = \mathbb{C}^G$ by $\mathbb{C}^{\mathcal{S}}$. This is the case, for example, for data consisting of images acted by the cyclic group via rotations, since the input space cannot be identified with signals over G. Assuming the action over $\mathcal{S}$ is *free*, meaning that $g \cdot s = s$ implies $g = 1$, $\mathcal{S}$ can be decomposed into copies of G deemed *orbits*. Specifically, there is an equivariant isomorphism $\mathcal{S} \simeq G \sqcup \cdots \sqcup G$, which in turn induces a linear isomorphism $\mathbb{C}^{\mathcal{S}} \simeq \langle G \rangle^{\oplus p}$, where p is the number of orbits. The results from this section can be extended to this scenario by applying all the arguments to the copies of $\langle G \rangle$ separately, each of which will serve as a domain for its set of irreducible unitary representations. When the action is not free, it is necessary to take into account *stabilizers*, i.e. $g \in G$ such that $g \cdot s = s$ for some $s \in \mathcal{S}$. Roughly speaking, we expect that the results from this section can be adapted to an extent, obtaining unitary representations 'up to stabilizers'. However, the precise meaning of the latter has yet to be clarified, and goes beyond the scope of this work.

G.3.1 Examples

In this section, we provide examples of machine learning models with unitary symmetries. As anticipated, in the context of machine learning $\mathcal{H}$ and $\mathcal{W}$ represent the hypothesis space and the parameter space, respectively. Indeed, in what follows $\mathcal{H}$ will consist of functions of the form $\langle G \rangle \to \mathcal{Y}$ for some codomain $\mathcal{Y}$, and we will adhere to the notation from Definition G.3.2 accordingly. All the models considered in this section satisfy the adjunction property.

Spectral Networks

We start by considering *Spectral Networks* – a class of polynomial machine learning models that inspired this work—and to which our theory applies naturally. These models were introduced by [1] in cubic form based on the invariant theory of $\langle G \rangle$ (see Section G.6.2 in the Appendix for an overview of the latter). The overall idea behind Spectral Networks is to approximate the n-order polynomial invariants (deemed spectral invariants) of $\langle G \rangle$ for an *unknown* group G. Specifically, suppose that $V_1, \cdots, V_k$ are the ambient Hilbert spaces of the irreducible unitary representations of G. Given a multi-index $\underline{i} = (i_1, \cdots, i_n) \in \{1, \cdots, k\}^n$, The Spectral Network of order n is defined as the collection of parametric maps $\varphi_{\underline{i}}(W, \cdot) : \langle G \rangle \to \mathrm{End}(V_{i_1} \otimes \cdots \otimes V_{i_n})$:

$$\varphi_{\underline{i}}(W, x) = W_{i_1}(x) \otimes \cdots \otimes W_{i_n}(x) \ \left(W_{i_1}^{\dagger} \odot \cdots \odot W_{i_n}^{\dagger} \right)(\overline{x}), \tag{G.14}$$

where $W = \oplus_i W_i \in \mathcal{W} = \langle G \rangle \oplus_i \otimes \mathrm{End}(V_i)$, $\odot$ denotes the G-wise tensor product of operators, and $\overline{x}$ denotes the component-wise conjugate of x. For a commutative G, since $V_i = \mathbb{C}$ for all i, the above expression reduces to:

$$\varphi_{\underline{i}}(W, x) = W_{i_1} x \cdots \; W_{i_n} x \; \overline{W_{i_1} \odot \cdots \odot W_{i_n} \; x}. \tag{G.15}$$

For $n = 1$ Spectral Networks are deemed Power-Spectral Networks, and for a commutative G they take the form $\varphi_i(W, x) = |W_i x|^2$. The latter can be simply interpreted as a linear model followed by an activation function. Even though the squared absolute value is uncommon as an activation function in machine learning, it has appeared in models of biological neural networks [38].

For simplicity, we will consider only the Spectral Networks involving a single unitary representation; that is, we will focus on constant multi-indices $\underline{i} = (i, \cdots, i)$ in Equation G.14. To this end, let V be a finite-dimensional Hilbert space and $\mathcal{H}$ be the set of functions $\langle G \rangle \to \mathrm{End}(V^{\otimes n})$ for some $n \in \mathbb{N}$. We set $\mathcal{W} = \langle G \rangle \otimes \mathrm{End}(V)$.

Proposition G.3.3. *Consider the Spectral Network, given by:*

$$\varphi(W, x) = W(x)^{\otimes n} \; W^{\dagger \odot n}(\overline{x}). \tag{G.16}$$

Then φ has unitary symmetries.

We refer to the Appendix for a proof.

McCulloch-Pitts Neurons and Deep Networks

While Spectral Networks provide the most direct application of our theory, in this section we discuss the most common and fundamental neural network primitives in deep learning: the fully-connected McCulloch-Pitts neuron [39] and the deep neural network. We consider models with complex coefficients and focus on commutative groups, i.e. all the Hilbert spaces V_i from Definition G.3.1 will be equal to $\mathbb{C}$.

A McCulloch-Pitts neuron has the form $\varphi(W, x) = \sigma(Wx)$, where $\sigma \colon \mathbb{C} \to \mathcal{Y}$ is a map playing the role of an activation function and $W \in \mathcal{W} = \langle G \rangle$ is the weight vector. For $\sigma(z) = |z|^2$, the McCulloch-Pitts neuron reduces to a commutative Power-Spectral Network i.e., a Spectral Network with $n = 1$. The hypothesis space $\mathcal{H}$ consists of functions $\langle G \rangle \to \mathcal{Y}$.

Proposition G.3.4. *Consider a map $\sigma \colon \mathbb{C} \to \mathcal{Y}$ and let $\varphi(W, x) = \sigma(Wx)$. Suppose that $0 \in \mathbb{C}$ is isolated in its fiber of σ, i.e. there exists an open subspace $O \subseteq \mathbb{C}$ such that $\sigma^{-1}(\sigma(0)) \cap O = \{0\}$. Then φ has unitary symmetries.*

We refer to the Appendix for a proof. The above assumption on σ is satisfied by popular activations functions from neuroscience and machine learning, such as the sigmoid and the leaky Rectified Linear Unit (ReLU), applied after taking complex

absolute value.

Next, we discuss the case of classical deep networks. We model the latter as $\varphi(W, x) = \chi(|W(x)|^2)$, where $W \in \mathcal{W} = \langle G \rangle \otimes \mathbb{C}^k$ and $|\cdot|$ denotes the component-wise absolute value. Here, k is the number of neurons in the first hidden layer of the network, while $\chi \colon \mathbb{R}^k_{\geq 0} \to \mathbb{R}$ is the head of the network, encompassing all the layers after the first one. Note that since typical neural networks are real-valued, we combine real and complex models. Namely, the first layer of φ is complex, and its output is fed into the real head χ by taking squared absolute values.

Differently from Section G.3, for the next result we will restrict $\mathcal{W}$ to the subspace of $\langle G \rangle \otimes \mathbb{C}^k$ consisting of W such that the components W_i are orthonormal. This implies, in particular, the constrain $k \leq |G|$. Note that $\mathcal{W}$ is closed by the actions of G and $\mathrm{U}(\mathbb{C})$. The orthonormality condition is anyway necessary in order to recover the full Fourier transform, as stated in Corollary G.3.2.

Proposition G.3.5. *Suppose that there is an open subspace $O \subseteq \mathbb{R}^k_{\geq 0}$ containing 0 where χ is affine with distinct non-vanishing coefficients i.e., $\chi(z) = \sum_i a_i z_i + b$ for $z \in O$ with $0 \neq a_i \neq a_j$ for $i \neq j$. Then $\varphi(W, x) = \chi(|W(x)|^2)$ has unitary symmetries.*

We refer to the Appendix for a proof. Since typical (real-valued) deep neural networks have piece-wise linear activations functions such as (leaky) ReLU, they define piece-wise affine maps and therefore are affine when restricted to appropriate open subspaces. Moreover, the hypothesis on the coefficients a_i in Proposition G.3.5 is *generic*, meaning that it defines an open dense subset of $(a_1, \cdots, a_k) \in \mathbb{R}^k$.

G.3.2 Group Recovery

Corollary G.3.2 allows us to recover the group structure of G up to isomorphism from the weights of a map with unitary symmetries. In other words, this enables the recovery of an unknown group in a data-driven manner from the weights of an invariant machine learning model, addressing the problem of symmetry discovery discussed in Section G.1.1. The procedure was originally suggested and validated empirically in [1].

To this end, assume that φ_i satisfies the requirements of Corollary G.3.2. Moreover, we introduce the additional assumption that $W_i(1) = I \in \mathrm{End}(V_i)$ for all i. If that is the case, W coincides exactly with the Fourier transform by Corollary G.3.2. This implies that the multiplication table of G can be recovered by:

$$gh = \operatorname*{argmin}_{l \in G} \|W(g) \odot W(h) - W(l)\|, \tag{G.17}$$

where $\odot$ denotes the Hadamard product i.e., component-wise operator composition. Note that this notation has a different meaning here than in Section G.3.1. Since

the $W(g)$'s are orthogonal, the only possible values for the norms over which the minimum is performed are 0 and $\sqrt{2|G|}$.

The multiplication table of G is a discrete object, while the weights $W \in \mathcal{W}$ can vary continuously. Therefore, it is natural to expect that the invariance condition ($g \cdot W = W$ for all $g \in G$) can be relaxed, while still recovering the multiplication table correctly. In what follows, we analyze relaxations of invariance and give bounds in which the group recovery algorithm holds. We start by introducing a quantity measuring how close a map is to having unitary symmetries. To this end, we assume that $\mathcal{H}$ is a metric space with distance function $\Delta : \mathcal{H} \times \mathcal{H} \to \mathbb{R}_{\geq 0}$.

Definition G.3.3. Given a map $\varphi\colon \mathcal{W} \to \mathcal{H}$, its *unitarity defect* is defined for $\delta \in \mathbb{R}_{>0}$ as:

$$\omega_\varphi(\delta) = \sup_{\substack{W,W' \in \mathcal{W} \\ \|W\| = \|W'\| \\ \Delta(\varphi(W),\varphi(W')) \leq \delta}} \max_i \inf_{U \in \mathrm{U}(V_i)} \|W_i - U \cdot W'_i\|, \tag{G.18}$$

Note that φ has unitary symmetry if, and only if, $\omega_\varphi(0) = 0$. The following is our main relaxation result.

Theorem G.3.6. *Suppose that $\varphi\colon \mathcal{W} \to \mathcal{H}$ is a map and fix $W \in \mathcal{W}$. Denote*

$$L = \|W^\dagger W - |G| I\|_\infty, \tag{G.19}$$

where $\|\cdot\|_\infty$ is the uniform Frobenius norm for $G \times G$ matrices. Suppose that the following holds:

- *For all $g \in G$:*

$$\omega_{\varphi_i}\left(\Delta(\varphi(g \cdot W), \varphi(W))\right) < \frac{\sqrt{\frac{1}{2} - \frac{L}{|G|}}}{\sqrt{|G| + L + 1}} \tag{G.20}$$

- $L \leq \frac{|G|}{2}$.
- $W_i(1) = I \in \mathrm{End}(V_i)$ *for all i.*

Then Equation G.17 holds, i.e. the group recovery algorithm is correct.

Proof. Firstly, the definition of L implies the inequalities $|G| - L \leq \|W(g)\|^2 \leq |G| + L$ and $|\langle W(g), W(h)\rangle| \leq L$ for all $g, h \in G$. In particular,

$$\|W(g) - W(h)\|^2 = \|W(g)\|^2 + \|W(h)\|^2 - 2\mathrm{Re}(\langle W(g), W(g)\rangle) \geq 2|G| - 4L. \tag{G.21}$$

By hypothesis, for each i and $g \in G$ there exists $\rho_i(g) \in \mathrm{U}(V_i)$ such that

$$\|g^{-1} \cdot W_i - \rho_i(g) \cdot W_i\| < \frac{\sqrt{\frac{1}{2} - \frac{L}{|G|}}}{\sqrt{|G| + L + 1}}. \tag{G.22}$$

In particular, $\|W_i(gh) - \rho_i(g)W_i(h)\|$ is bounded by the same quantity for all $g, h \in G$. Therefore, via the triangle inequality we see that:

$$\begin{aligned} &\|W_i(g)W_i(h) - W_i(gh)\| \leq && \text{(G.23)}\\ \leq &\|W_i(h)W_i(g) - \rho_i(g)W_i(h)\| + \|W_i(gh) - \rho_i(g)W_i(h)\| = && \text{(G.24)}\\ = &\underbrace{\|W_i(h)\|}_{\leq\sqrt{|G|+L}}\ \|W_i(g) - \rho_i(g)W_i(1)\| + \|W_i(gh) - \rho_i(g)W_i(h)\| < && \text{(G.25)}\\ < &\sqrt{\frac{1}{2} - \frac{L}{|G|}}. && \text{(G.26)} \end{aligned}$$

Equation G.23 implies that

$$\|W(g) \odot W(h) - W(gh)\| < \sqrt{|G|}\sqrt{\frac{1}{2} - \frac{L}{|G|}} = \frac{\sqrt{2|G| - 4L}}{2}. \tag{G.27}$$

Since $\|W(p) - W(q)\| \geq \sqrt{2|G| - 4L}$ for all $p \neq q \in G$, $W(g) \odot W(h)$ is closer to $W(gh)$ than to any other $W(q)$ for $q \in G$, which immediately implies the desired result. □

The assumption on L in the above result is a relaxation of the unitarity assumption in Corollary G.3.2 since $L = 0$ if, and only if, W is unitary up to a multiplicative constant.

We provide an explicit bound for the unitarity defect of the McCulloch-Pitts neurons discussed in Section G.3.1. To this end, let $\mathcal{H}$ be the space of continuous functions defined on the unit sphere in $\langle G \rangle$ equipped with the uniform metric (i.e., the L_∞ distance) as Δ.

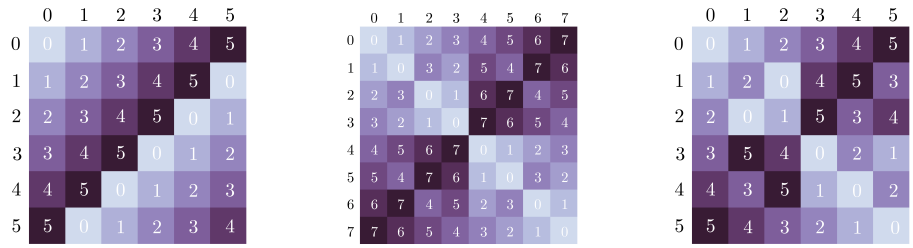
Proposition G.3.7. *Let $W \in \langle G \rangle$ and consider $\varphi(W, x) = \sigma(Wx)$. Suppose that the activation function $\sigma \colon \mathbb{C} \to \mathbb{C}$ is continuous and satisfies the following coercivity condition: there exist constants $C \in \mathbb{R}_{>0}$, $n \in \mathbb{N}$ such that for every $x \in \mathbb{C}$:*

$$|\sigma(0) - \sigma(x)| \geq C\ |x|^n. \tag{G.28}$$

Then the unitarity defect of φ satisfies for $\delta < C$:

$$\omega_\varphi(\delta) \leq \sqrt{2\left(1 - \sqrt{1 - \left(\frac{\delta}{C}\right)^{\frac{2}{n}}}\right)}. \tag{G.29}$$

We refer to the Appendix for a proof. Note that the coercivity condition from above plays the role of the assumption on σ from Proposition G.3.4.



$C_6 \simeq C_2 \times C_3$ $C_2 \times C_2 \times C_2$ $D_3 \simeq S_3$

Figure G.2: Learned Group Multiplication Tables. Tables inferred by Power-Spectral Networks for the groups C_6 (commutative), $C_2 \times C_2 \times C_2$ (commutative), and D_3 (non-commutative). Rows and columns are labeled with integers that index group elements. Each cell of the table contains the index of the group element obtained by composing the group elements indexed in the row and column. Note that the table is a symmetric matrix if, and only if, the group is commutative.

Implementation

We now empirically explore the theory developed in this paper and demonstrate that Spectral Networks are able to recover the group structure in practice. To this end, we implement a non-commutative Power-Spectral Network $\varphi_i(W, x) = W_i x \, W_i^\dagger \overline{x}$ with weights $W \in \mathbb{C}^{d_i \times d_i \times d}$, where $d = |G|$ is the cardinality of the given group and $d_1, \cdots, d_k$ are the dimensions of its irreducible unitary representations. As discussed at the beginning of Section G.3.2, we force $W_i(1) = I \in \mathbb{C}^{d_i \times d_i}$, where the index 1 is arbitrarily chosen.

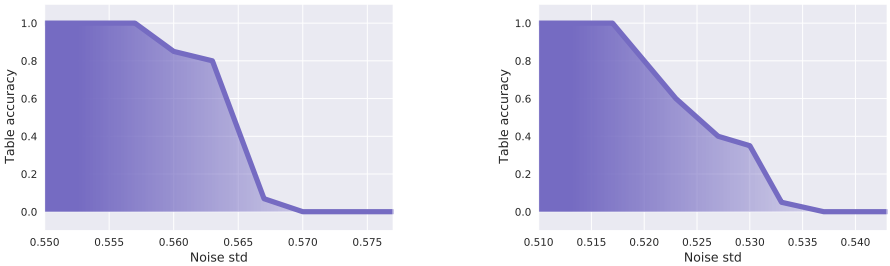
Following [1], we train the model via *contrastive learning* [40]. Namely, given a finite dataset $\mathcal{D}$ of pairs (x, y), where $x, y \in \langle G \rangle \simeq \mathbb{C}^d$ and $x = g \cdot y$ for an unknown $g \in G$, the objective optimized by the model is:

$$\mathcal{L}(W) = \sum_{(x,y)\in\mathcal{D}} \sum_i \|\varphi_i(W, x) - \varphi_i(W, y)\|^2 + \eta \, \|dI - WW^\dagger\|^2, \tag{G.30}$$

where $\eta > 0$ is a hyper-parameter and $\|\cdot\|$ is the Frobenius norm. The first term in Equation G.30 encourages invariance with respect to G while the second one encourages W to be unitary.

The model is trained via the Adam optimizer [41], which interprets the complex weights as real tensors of doubled dimensionality. The dataset $\mathcal{D} \subseteq \langle G \rangle \oplus \langle G \rangle$ is generated procedurally by first sampling x from a standard Gaussian distribution over $\langle G \rangle \simeq \mathbb{R}^{2d}$, then sampling $g \in G$ uniformly, and finally producing the datapoint $(x, y = g \cdot x) \in \mathcal{D}$. We provide a Python implementation of both the model

and the experiments at a public repository (see Section G.1.1). The code is available in both the PyTorch [42] and the JAX [43] frameworks.



C_5 C_6

Figure G.3: Group Recovery Accuracy. Accuracy for the recovery of the group multiplication tables across 20 training runs as the amount of noise injected into data increases. The transition from 1.0 to 0.0 accuracy is sharp, and here we visualize only the noise regions where the values are non-trivial.

Once trained, we evaluate the model by checking whether the multiplication table $M \in \{1, \cdots, d\}^{d \times d}$ obtained via the group recovery algorithm described by Equation G.17 coincides with the one of G. Since there is no canonical ordering on G, the table is recovered up to a permutation π of $\{1, \cdots, d\}$ acting as $(\pi \cdot M)_{i,j} = \pi(M_{\pi^{-1}(i),\pi^{-1}(j)})$. Therefore, we check whether $\pi \cdot M$ coincides with the table of G for all the permutations π. Figure G.2 reports the (correct) multiplication tables obtained at convergence for both commutative and non-commutative groups. Specifically, we consider the cyclic group C_6, the product of cyclic groups $C_2 \times C_2 \times C_2$ and the dihedral group D_3, which is isomorphic to the symmetric group S_3.

In order to validate empirically the robustness of the group recovery procedure, we additionally train the model on data corrupted by white noise, i.e. $\mathcal{D}$ consists of pairs $(x, y = g \cdot x + \epsilon)$, where ϵ is sampled from an isotropic Gaussian distribution. We vary the standard deviation of the latter and report in Figure G.3 the number of times the multiplication table is recovered correctly across 20 training runs – a metric referred to as 'table accuracy'.

As can be seen, the group structure is recovered most of the times even with large amounts of noise – up to ~ 0.5 standard deviation. The performance quickly degrades as the noise reaches a critical threshold. This demonstrates empirically that the group recovery procedure is robust to noise, which is in line with the theoretical bounds from Theorem G.3.6.

G.4 Conclusions, Limitations, and Future Work

In this work, we proved that if a machine learning model of a certain kind is invariant to a finite group, then its weights are closely related to the Fourier transform on that group. We discussed how, as a consequence, the algebraic structure of an unknown group can be recovered from a model that is invariant. We established these results for both commutative and non-commutative groups, and discussed relaxed conditions under which the group recovery procedure holds. Our results represent a first step towards a mathematical explanation of universal features inferred by both biological and artificial neural networks.

Due to its open-ended nature, this work is subject to a number of limitations and leaves directions open for future investigation. First, our theory encompasses models with complex-valued weights, which are non-canonical in machine learning. Thus, exploring analogues of the theory over real numbers is an interesting direction that would fit more directly with current practices in the field.

In addition, our theoretical framework encompasses learning models with unitary symmetries. The latter is a technical property satisfied by Spectral Networks and, to an extent, by traditional deep networks. However, it is not clear what other models fit into our framework, and whether the notion is general enough to accommodate other computational primitives fundamental to machine learning, such as the attention mechanisms. This is an open question that is worthy of investigation.

Lastly, in this work we focused on groups and their associated harmonics. However, the representations within neural networks or biological systems often resemble imperfect, or *localized*, versions of harmonics, i.e. wavelets, such as Gabors. Since wavelets do not describe group homomorphisms, our theory would need to be extended to accommodate this kind of locality. We suspect that this may be achieved by generalizing the framework to *groupoids* – an algebraic group-like structure that formalizes a locally-defined composition. This, however, goes beyond the scope of our work, and we leave it for future research.

G.5 Acknowledgements

This work was supported by the Swedish Research Council, the Knut and Alice Wallenberg Foundation and the European Research Council (ERC-BIRD-884807).

G.6 Appendix

G.6.1 Proofs of Theoretical Results

Proof of Proposition G.3.3

In order to prove this proposition, we will need some technical facts from matrix algebra. We start by showing the following uniqueness result.

Lemma G.6.1. *Let $d' \geq d$ and $A, B \in \mathbb{C}^{d' \times d}$. If $AA^\dagger = BB^\dagger$, then there exists a unitary matrix $U \in \mathbb{C}^{d \times d}$ such that $A = BU$.*

Proof. From the polar decomposition for matrices (see [44, Theorem 3.1.9]), we know that:

$$A = PV, \; B = QW, \tag{G.31}$$

where $P, Q \in \mathbb{C}^{d' \times d'}$ are Hermitian positive semidefinite and $V, W \in \mathbb{C}^{d' \times d}$ have orthonormal rows. Also, $P^2 = AA^\dagger = BB^\dagger = Q^2$ from which it follows that $P = Q$ by uniqueness of square roots of positive semidefinite Hermitian matrices (see [45, Theorem 7.2.6]). In particular, we have:

$$A = PV = QV = QWW^\dagger V = BU, \tag{G.32}$$

with $U = W^\dagger V$ unitary. □

Remark G.6.1. We note that if A and B are real matrices, then the conclusion holds with U being a real orthogonal matrix.

Next, we show that positive semidefinite Hermitian matrices possess unique tensor roots.

Lemma G.6.2. *Let $A, B \in \mathbb{C}^{d \times d}$ be Hermitian and positive semidefinite. If $A^{\otimes n} = B^{\otimes n}$ for some $n > 0$, then $A = B$.*

Proof. From the Spectral Theorem we know that:

$$A = UDU^\dagger, \tag{G.33}$$

where U is unitary and D is diagonal. From $A^{\otimes n} = B^{\otimes n}$ it follows that $D^{\otimes n} = C^{\otimes n}$, where $C = U^\dagger BU$. Note that the (point-wise) Hadamard product of matrices is a submatrix of the tensor (Kronecker) product. In particular, the off-diagonal entries of C must vanish. On the diagonal we have $(D_{i,i})^n = (C_{i,i})^n$ for every i, and therefore $D_{i,i} = C_{i,i}$ since they are non-negative. This shows that $C = D$, implying that $B = UDU^\dagger = A$. □

By putting together the above lemmas, we obtain the following.

Lemma G.6.3. *Let $A_1, \cdots, A_k, B_1, \cdots, B_k \in \mathbb{C}^{d\times d}$. Suppose that for some $n > 0$, for all $x \in \mathbb{C}^k$:*

$$\left(\sum_i x_i A_i\right)^{\otimes n} \left(\sum_i \overline{x_i} A_i^{\dagger\otimes n}\right) = \left(\sum_i x_i B_i\right)^{\otimes n} \left(\sum_i \overline{x_i} B_i^{\dagger\otimes n}\right). \tag{G.34}$$

Then there exists a unitary matrix $U \in \mathbb{C}^{d\times d}$ such that $A_i = B_i U$ for every i.

Proof. By multilinearity of the tensor product we see that for all $x \in \mathbb{C}^k$:

$$\sum_{i_1,\cdots,i_{n+1}} x_{i_1}\cdots x_{i_n}\overline{x}_{i_{n+1}}(A_{i_1}\otimes\cdots\otimes A_{i_n})A_{i_{n+1}}^{\dagger\otimes n} = \tag{G.35}$$

$$= \sum_{i_1,\cdots,i_{n+1}} x_{i_1}\cdots x_{i_n}\overline{x}_{i_{n+1}}(A_{i_1}A_{i_{n+1}}^\dagger)\otimes\cdots\otimes(A_{i_n}A_{i_{n+1}}^\dagger) = \tag{G.36}$$

$$= \sum_{i_1,\cdots,i_{n+1}} x_{i_1}\cdots x_{1_n}\overline{x}_{i_{n+1}}(B_{i_1}B_{i_{n+1}}^\dagger)\otimes\cdots\otimes(B_{i_n}B_{i_{n+1}}^\dagger). \tag{G.37}$$

Since a polynomial vanishes as function if and only if it is the zero polynomial, it follows that $(A_{i_1}A_{i_{n+1}}^\dagger)\otimes\cdots\otimes(A_{i_n}A_{i_{n+1}}^\dagger) = (B_{i_1}B_{i_{n+1}}^\dagger)\otimes\cdots\otimes(B_{i_n}B_{i_{n+1}}^\dagger)$ for all $i_1, \cdots, i_{n+1}$, and in particular $(A_iA_j^\dagger)^{\otimes n} = (B_iB_j^\dagger)^{\otimes n}$ for all i, j. From Lemma G.6.2 we conclude that $A_iA_j = B_iB_j$ for all i, j, which can be rephrased as $AA^\dagger = BB^\dagger$, where A, B are the $(dk)\times d$ matrices obtained by concatenating the A_i's and B_i's respectively. From Lemma G.6.1 we conclude that $A = BU$ for a unitary $d\times d$ matrix U, as desired. □

Proposition G.3.3 now follows immediately from Lemma G.6.3 by setting $A_i = W(g_i)$ and $B_i = W'(g_i)$ for $g_i \in G$ and $W, W' \in \langle G\rangle \otimes \mathrm{End}(V)$ of the same norm.

Proof of Proposition G.3.4

Proof. Pick $W, W' \in \langle G\rangle$ of the same norm such that $\varphi(W, x) = \varphi(W', x)$ for all $x \in \langle G\rangle$. Given the open set $O \subseteq \mathbb{C}$ from the hypothesis on σ, consider $O' = \{x \in \langle G\rangle \mid Wx, W'x \in O\}$, which is open and non-empty since $0 \in O'$. For $x \in O'$, $Wx = 0$ implies $\varphi(W, x) = \varphi(W', x) = 0$, from which it follows that $W'x = 0$ by definition of O. Therefore, W and W' share the same orthogonal complement, implying that $W' = \rho W$ for some $\rho \in \mathbb{C}$. Since W and W' have the same norm, we conclude that $\rho \in \mathrm{U}(\mathbb{C})$. □

Proof of Proposition G.3.5

Proof. Consider $W, W' \in \mathcal{W}$ such that $\varphi(W, x) = \varphi(W', x)$ for all $x \in \langle G\rangle$. Given the open set $O \subseteq \mathbb{R}^k_{\geq 0}$ from the hypothesis on χ, consider $O' = \{x \in$

$\langle G\rangle \mid |Wx|^2, |W'x|^2 \in O\}$, which is open and non-empty since $0 \in O'$. For $x \in O'$, $\varphi(W, x)$ can be written as:

$$\varphi(W, x) = \sum_i a_i |W_i\ x|^2 + b, \tag{G.38}$$

with the a_i's being distinct, and similarly for $\varphi(W', x)$. Since Hermitian forms are determined by their restriction on an open set, we deduce the following identity of operators:

$$\sum_i a_i W_i \otimes \overline{W_i} = \sum_i a_i W_i' \otimes \overline{W_i'}. \tag{G.39}$$

Since both the sets $\{W_i\}_i$ and $\{W_i'\}_i$ are orthonormal by hypothesis on $\mathcal{W}$, both sides of Equation G.39 define a spectral decomposition, i.e. a decomposition into projections over orthonormal vectors. Since the eigenvalues a_i are distinct and non-vanishing, it follows that $W_i = \rho_i W_i'$ for some $\rho_i \in \mathrm{U}(\mathbb{C})$, as desired. □

Proof of Proposition G.3.7

In order to prove this proposition, we will need the following technical fact from linear algebra.

Lemma G.6.4. *Let H be a finite-dimensional complex Hilbert space, $v, w \in H$ normal and $\varepsilon \in \mathbb{R}$ such that $0 < \varepsilon < 1$. Suppose that for every normal x orthogonal to w, it holds that $|\langle x, v\rangle| \leq \varepsilon$. Then there exists $\rho \in \mathrm{U}(\mathbb{C})$ such that*

$$\|v - \rho w\| \leq \sqrt{2\left(1 - \sqrt{1 - \varepsilon^2}\right)}. \tag{G.40}$$

Proof. Consider an orthogonal decomposition $v = \langle w_1, v\rangle w_1 + \langle w_2, v\rangle w_2$, where $w_1 \in w^\perp$ is normal and $w_2 = \rho w$ for some $\rho \in \mathrm{U}(\mathbb{C})$ such that $\langle w_2, v\rangle \in \mathbb{R}_{\geq 0}$. It follows that:

$$1 = \|v\|^2 = |\langle w_1, v\rangle|^2 + \langle w_2, v\rangle^2. \tag{G.41}$$

The hypothesis implies then that $\langle w_2, v\rangle \geq \sqrt{1 - \varepsilon^2}$. Therefore,

$$\|v - w_2\|^2 = 2 - 2\langle w_2, v\rangle \leq 2\left(1 - \sqrt{1 - \varepsilon^2}\right), \tag{G.42}$$

as desired. □

Remark G.6.2. Note that the right-hand side of Equation G.40 is bounded by the concise quantity $\sqrt{2\varepsilon}$.

We are now ready to prove Proposition G.3.7.

Proof. Consider $\delta \in \mathbb{R}_{>0}$ and $W, W' \in \langle G \rangle$ of the same norm such that $\Delta(\varphi(W), \varphi(W')) \leq \delta$. The latter and the hypotheses together imply that if $x \in \langle G \rangle$ is normal such that $W'x = 0$, then:

$$C \ |Wx|^n \leq |\sigma(0) - \sigma(Wx)| \leq \delta. \tag{G.43}$$

By Lemma G.6.4 there exists $\rho \in \mathrm{U}(\mathbb{C})$ such that:

$$\|W - \rho W'\| \leq \sqrt{2\left(1 - \sqrt{1 - \left(\frac{\delta}{C}\right)^{\frac{2}{n}}}\right)}, \tag{G.44}$$

from which the claim follows. □

G.6.2 Spectral Invariants

In this section, we overview the theory of *invariants* over $\langle G \rangle$, i.e. (polynomial) maps $\langle G \rangle \to \mathbb{C}$ that are invariant with respect to the action by G. To this end, we recall the following notion.

Definition G.6.1. Fix $n > 0$ and $\underline{\rho} = (\rho_1, \ldots, \rho_n) \in (G^\vee)^n$. The *spectrum of order* n associated to $\underline{\rho}$ is defined for $x \in \langle G \rangle$ as:

$$\beta_{\underline{\rho}}(x) = \hat{x}_{\rho_1} \cdots \hat{x}_{\rho_n} \overline{\hat{x}}_{\rho_1 \cdots \rho_n} \tag{G.45}$$

The spectra of order n are invariant polynomials of degree $n+1$ containing one conjugate variable. The presence of the latter is necessary for invariance. For $n = 1, 2$ they are alternatively referred to as *power spectra* and *bispectra* respectively. Note that the power spectra reduce simply to $\beta_\rho(x) = |\hat{x}_\rho|^2$, $\rho \in G^\vee$, and constitute a standard tool in signal processing. Bispectra, together with higher-order spectra, were first introduced in [46]. It is immediate to see that spectra generate all the polynomial invariants of $\langle G \rangle$ (see also [47, Theorem 2.1.4]).

Proposition G.6.5. *The space of polynomial invariants of degree $n+1$ over $\langle G \rangle$ (with one conjugate variable) is generated as a complex vector space by the spectra of order n.*

Proof. This follows from interpreting the invariance condition via the Fourier transform. Namely, given $\underline{\rho} \in (G^\vee)^{n+1}$ consider the monomial over $\langle G^\vee \rangle$ defined by $\hat{x}_{\rho_1} \cdots \hat{x}_{\rho_n} \overline{\hat{x}}_{\rho_{n+1}}$. Since $g \cdot \hat{x} = (\overline{\rho(g)}\ \hat{x}_\lambda)_{\lambda \in G^\vee}$, the monomial is invariant if and only if $\rho_1(g) \cdots \rho_n(g) \overline{\rho}_{n+1}(g) = 1$ for all $g \in G$, i.e. $\rho_{n+1} = \rho_1 \cdots \rho_n$. Since monomials linearly generate polynomials, the claim follows. □

A remarkable aspect of spectra of even order is the fact that they jointly determine real (generic) elements of $\langle G \rangle$ up to the action by G – a property known as *completeness*. This was first shown in [48]. For convenience, we report below a simple proof for finite commutative groups.

Proposition G.6.6. *Fix n even. Suppose that $x, y \in \mathbb{R}^G \subseteq \langle G \rangle$ are such that $\hat{x}_\rho, \hat{y}_\rho \neq 0$ for all $\rho \in G^\vee$. If $\beta_{\underline{\rho}}(x) = \beta_{\underline{\rho}}(y)$ for all $\underline{\rho} \in (G^\vee)^n$ then $x = g \cdot y$ for some $g \in G$.*

Proof. By setting $\underline{\rho} = (1, \cdots, 1)$ we see that $\langle 1, x\rangle^{n+1} = \langle 1, y\rangle^{n+1} \in \mathbb{R} \setminus \{0\}$ and therefore $\langle 1, x\rangle = \langle 1, y\rangle$ since n is even. For $\rho \in G^\vee$, by setting $\underline{\rho} = (\rho, \overline{\rho}, 1, \cdots, 1)$, we see that $\langle \rho, x\rangle\langle \overline{\rho}, x\rangle\langle 1, x\rangle^{n-1} = |\langle \rho, x\rangle|^2\langle 1, x\rangle^{n-1} = |\langle \rho, y\rangle|^2\langle 1, y\rangle^{n-1}$ and therefore $|\langle \rho, x\rangle| = |\langle \rho, y\rangle|$. Note that here we relied on the fact that x and y are real. This implies that the following map $\eta : G^\vee \to \mathbb{C}$ takes values in $\mathrm{U}(\mathbb{C})$:

$$\eta(\rho) = \frac{\langle \rho, x\rangle}{\langle \rho, y\rangle}. \tag{G.46}$$

Now, $\eta(1) = 1$ since $\langle 1, y\rangle = \langle 1, y\rangle$. By setting $\underline{\rho} = (\rho, \mu, \overline{\rho\mu}, 1, \cdots, 1)$ we see that $\eta(\rho)\eta(\mu) = \eta(\overline{\rho\mu})$ for all $\rho, \mu \in G^\vee$ and therefore $\eta \in (G^\vee)^\vee$. Since the Fourier transform sends $G^\vee \subseteq \langle G\rangle$ to $(G^\vee)^\vee \subseteq \langle G^\vee\rangle$, there exists $g \in G$ such that $\eta(\rho) = \overline{\rho(g)}$. This means that $\langle \rho, x\rangle = \overline{\rho(g)}\langle \rho, y\rangle$, which implies $x = g \cdot y$ by the equivariance properties of the Fourier transform. □

Spectral invariants can be defined in the non-commutative case, but arise subtleties. First, we replace the group structure of $G^\vee$ with the tensor product $\otimes$ of unitary representations. However, $\mathrm{Irr}(G)$ is not closed with respect to $\otimes$. This is circumvented by considering *Clebsch-Gordan coefficients*, i.e. irreducible unitary sub-representations of tensor products. This leads to the following definition of operator-valued spectra.

Definition G.6.2. Fix $n > 0$ and $\underline{\rho} = (\rho_{V_1}, \cdots, \rho_{V_n}) \in \mathrm{Irr}(G)^n$. The *spectrum of order n* associated to $\underline{\rho}$ is defined for $x \in \langle G\rangle$ as:

$$\beta_{\underline{\rho}}(x) = \hat{x}_{\rho_{V_1}} \otimes \cdots \otimes \hat{x}_{\rho_{V_n}} \left(\hat{x}^\dagger_{\rho_{T_1}} \oplus \cdots \oplus \hat{x}^\dagger_{\rho_{T_k}}\right) \in \mathrm{End}(V_1 \otimes \cdots \otimes V_n) \tag{G.47}$$

where the direct sum runs over the k irreducible unitary representations appearing in an orthogonal decomposition $V_1 \otimes \cdots \otimes V_n = T_1 \oplus \cdots \oplus T_k$.

The completeness of spectra of order $n \geq 2$ (Proposition G.6.6) extends to the non-commutative case [49].

References

[1] S. Sanborn, C. Shewmake, B. Olshausen, and C. Hillar, "Bispectral neural networks," *International Conference on Learning Representations (ICLR)*, 2023.

[2] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft, "Convergent learning: Do different neural networks learn the same representations?," *arXiv preprint arXiv:1511.07543*, 2015.

[3] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter, "An overview of early vision in inceptionv1," *Distill*, vol. 5, no. 4, pp. e00024–002, 2020.

[4] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, p. 574, 1959.

[5] C. A. Hass and G. D. Horwitz, "V1 mechanisms underlying chromatic contrast detection," *Journal of Neurophysiology*, vol. 109, no. 10, pp. 2483–2494, 2013.

[6] K. F. Willeke, K. Restivo, K. Franke, A. F. Nix, S. A. Cadena, T. Shinn, C. Nealley, G. Rodriguez, S. Patel, A. S. Ecker, *et al.*, "Deep learning-driven characterization of single cell tuning in primate visual area v4 unveils topological organization," *bioRxiv*, pp. 2023–05, 2023.

[7] T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell, "Toward transparent ai: A survey on interpreting the inner structures of deep neural networks," in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 464–483, IEEE, 2023.

[8] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[9] A. J. Bell and T. J. Sejnowski, "The "independent components" of natural scenes are edge filters," *Vision research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[10] J. Hurri and A. Hyvärinen, "Simple-cell-like receptive fields maximize temporal coherence in natural video," *Neural Computation*, vol. 15, no. 3, pp. 663–691, 2003.

[11] P.-É. H. Fiquet and E. P. Simoncelli, "Polar prediction of natural videos," *arXiv preprint arXiv:2303.03432*, 2023.

[12] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962.

[13] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[14] U. C. Dräger, "Receptive fields of single cells and topography in mouse visual cortex," *Journal of Comparative Neurology*, vol. 160, no. 3, pp. 269–289, 1975.

[15] N. Nanda, L. Chan, T. Liberum, J. Smith, and J. Steinhardt, "Progress measures for grokking via mechanistic interpretability," *arXiv preprint arXiv:2301.05217*, 2023.

[16] B. Chughtai, L. Chan, and N. Nanda, "A toy model of universality: Reverse engineering how networks learn group operations," *arXiv preprint arXiv:2302.03025*, 2023.

[17] E. I. Moser, E. Kropff, and M.-B. Moser, "Place cells, grid cells, and the brain's spatial representation system," *Annu. Rev. Neurosci.*, vol. 31, pp. 69–89, 2008.

[18] A. Guanella, D. Kiper, and P. Verschure, "A model of grid cells based on a twisted torus topology," *International journal of neural systems*, vol. 17, no. 04, pp. 231–240, 2007.

[19] J. Orchard, H. Yang, and X. Ji, "Does the entorhinal cortex use the fourier transform?," *Frontiers in computational neuroscience*, vol. 7, p. 179, 2013.

[20] R. J. Gardner, E. Hermansen, M. Pachitariu, Y. Burak, N. A. Baas, B. A. Dunn, M.-B. Moser, and E. I. Moser, "Toroidal topology of population activity in grid cells," *Nature*, vol. 602, no. 7895, pp. 123–128, 2022.

[21] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, *et al.*, "Vector-based navigation using grid-like representations in artificial agents," *Nature*, vol. 557, no. 7705, pp. 429–433, 2018.

[22] C. J. Cueva and X.-X. Wei, "Emergence of grid-like representations by training recurrent neural networks to perform spatial localization," *arXiv preprint arXiv:1803.07770*, 2018.

[23] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," *arXiv preprint arXiv:2104.13478*, 2021.

[24] G. B. Folland, *A course in abstract harmonic analysis*, vol. 29. CRC press, 2016.

[25] L. Moschella, V. Maiorca, M. Fumero, A. Norelli, F. Locatello, and E. Rodola, "Relative representations enable zero-shot latent space communication," *arXiv preprint arXiv:2209.15430*, 2022.

[26] G. Isely, C. Hillar, and F. Sommer, "Deciphering subsampled data: adaptive compressive sampling as a principle of brain communication," *Advances in neural information processing systems*, vol. 23, 2010.

[27] C. J. Hillar and F. T. Sommer, "When can dictionary learning uniquely recover sparse data from subsamples?," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6290–6297, 2015.

[28] C. J. Garfinkle and C. J. Hillar, "On the uniqueness and stability of dictionaries for sparse representation of noisy signals," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 5884–5892, 2019.

[29] W. Pitts and W. S. McCulloch, "How we know universals: The perception of auditory and visual forms," *The Bulletin of mathematical biophysics*, vol. 9, pp. 127–147, 1947.

[30] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[31] J. Bassey, L. Qian, and X. Li, "A survey of complex-valued neural networks," *arXiv preprint arXiv:2101.12249*, 2021.

[32] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks (2017)," *arXiv preprint arXiv:1705.09792*, 2017.

[33] S. Löwe, P. Lippe, M. Rudolph, and M. Welling, "Complex-valued autoencoders for object discovery," *arXiv preprint arXiv:2204.02075*, 2022.

[34] R. Rao and D. Ruderman, "Learning lie groups for invariant visual perception," *Advances in neural information processing systems*, vol. 11, 1998.

[35] J. Sohl-Dickstein, C. M. Wang, and B. A. Olshausen, "An unsupervised algorithm for learning lie group transformations," *arXiv preprint arXiv:1001.1027*, 2010.

[36] K. Desai, B. Nachman, and J. Thaler, "Symmetry discovery with deep learning," *Physical Review D*, vol. 105, no. 9, p. 096031, 2022.

[37] W. Rudin, "Fourier analysis on groups," *Bull. Amer. Math. Soc*, vol. 70, pp. 230–232, 1964.

[38] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *Josa a*, vol. 2, no. 2, pp. 284–299, 1985.

[39] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[40] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[43] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," in *GitHub*, 2018.

[44] R. A. Horn, R. A. Horn, and C. R. Johnson, *Topics in matrix analysis*. Cambridge university press, 1994.

[45] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[46] R. Kakarala, "The bispectrum as a source of phase-sensitive invariants for fourier descriptors: a group-theoretic approach," *Journal of Mathematical Imaging and Vision*, vol. 44, pp. 341–353, 2012.

[47] B. Sturmfels, *Algorithms in invariant theory*. Springer Science & Business Media, 2008.

[48] F. Smach, C. Lemaître, J.-P. Gauthier, J. Miteran, and M. Atri, "Generalized fourier descriptors with applications to objects recognition in svm context," *Journal of mathematical imaging and Vision*, vol. 30, pp. 43–71, 2008.

[49] R. Kakarala, "Completeness of bispectrum on compact groups," *arXiv preprint arXiv:0902.0196*, vol. 1, 2009.