Prospects for Quantum Equivariant Neural Networks

by

Grecia Castelazo

B.S. Physics; Electrical Engineering and Computer Science
Massachusetts Institute of Technology (2022)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2022

Author ....
Department of Electrical Engineering and Computer Science
August 15, 2022

Certified by ....
Seth Lloyd
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by ....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Prospects for Quantum Equivariant Neural Networks

by

Grecia Castelazo

Submitted to the Department of Electrical Engineering and Computer Science
on August 15, 2022, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Convolutional neural networks (CNNs) exploit translational invariance within images. Group equivariant neural networks comprise a natural generalization of convolutional neural networks by exploiting other symmetries arising through different group actions. Informally, a linear map is *equivariant* if it transfers symmetries from its input space into its output space. Equivariant neural networks guarantee equivariance for arbitrary groups, reducing the system design complexity. Motivated by the theoretical/experimental development of quantum computing, in particular with the quantum advantage derived from other quantum algorithms/subroutines for group theoretic and linear algebraic problems, we explore the potential of quantum computers to realize these structures in machine learning. This work reviews the mathematical machinery necessary from group representation theory, surveys the theory of equivariance, and combines results in non-commutative harmonic analysis and geometric deep learning. Convolutions and cross-correlations are examples of functions which are equivariant to the actions of a group. We present efficient quantum algorithms for performing linear finite-group convolutions and cross-correlations on data stored as quantum states. Potential implementations and quantizations of the infinite group cases also discussed.

Thesis Supervisor: Seth Lloyd
Title: Professor of Mechanical Engineering

Acknowledgements

There are many people that were crucial to my experience as and undergraduate and master's student at MIT. First and foremost I would like to thank my research faculty professor Seth Lloyd for sowing genuine joy for quantum information and quantum computation. I am deeply grateful for his succint teaching, unconditional support, professional and personal mentorship before and after joining his group. He has made the most supportive of advisors during my last year at MIT. I also have to thank Bobak T. Kiani for supervising all of the work that went into this thesis, for getting so much of his time and advice. Not less for his brilliant creativity, unique intuition and his compassionate leadership.

I am thankful to the Research Laboratory of Electronics at MIT for providing a space during the time of my master's studies and the last years of my undergraduate. I am also thankful to the Center of Theoretical Physics at MIT, for the inspiration I got in the weekly quantum information meetings. To the Physics Department for being my home for most of my undergraduate, and for the weekly Physics Colloquiums, which kept spiking curiosity and inspiration while performing my research on physics inspired algorithms. I was very fortunate to have received guidance and inspiration from renowned scientists and experienced researchers. I would like to thank Yen-Jie Lee for providing my first research opportunity in Particle Physics, and for supporting me as an early undergraduate. I would like to thank William Oliver and Antti Vepsalainen for the opportunities on experimental Quantum Engineering, and for the discussions that would serve of much inspiration earlier on (see preface Chapter 1); as well for encouraging, advising, and supporting me as a late undergraduate. I hope to make a good compromise of the theoretical and experimental areas of quantum

computing, including potential applications to the study of High Energy Physics, in my professional career and future doctoral studies.

While I am not a first generation in graduating from college, I am a first generation graduate student in STEM and aspiring scientist. I owe the soundness to have pursued my achievements to the support of my family and friends, in particular to my parents, Ana Martinez and Gustavo Castelazo, who have dedicated most of their lives to love me and invest in all of my self imposed dreams and expectations.

I dedicate this thesis to my grandmother, Célida Espinoza, who grew up in 20th century rural Sinaloa, Mexico. She had a brilliant mind, remarkable intution and an unwavering tenacity; although she was not allowed to attend school, she would've made a great physicist or mathematician if born in different circumstances.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Preface. The motivation of my study on quantum machine learning dates back to a conversation I had with my supervisor during my last year of undergraduate degree while performing the final experiments for my senior project on superconducting qubits. When I asked for his opinion on quantum machine learning he asked whether I meant "ML for quantum" or "quantum for ML"? I have found this distinction fascinating ever since: the first question has wide experimental application – how do we use machine learning to understand quantum mechanics better? – and the second question explores an entire field – how can quantum mechanics enhance our ability to find patterns in data?

Motivation. Machine learning (ML) encompasses a wide variety of algorithms, and modeling, classifying and categorizing tools for data processing tasks, becoming a predominant apparatus in most scientific disciplines in recent years. In fact, there are many examples of research cross fertilization between ML and the physical sciences, where applications of ML methods have simplified observations in experimental physics, and developments in ML have been driven by physical insights [5]. In the fields of quantum engineering we find instances where classical ML has aided quantum by speeding up parameter tuning, quantum device simulation and control in several qubit realizations such as superconducting qubits [22], quantum dots, nitrogen vacancy (NV) diamond centers and ion traps [17].

Machine learning also has the ability to provide approximate simulations of systems in nature, by learning models of a system and predicting the system's behavior. While machine learning is a rapidly expanding field fueled by the rapid progression of computer power, quantum systems produce atypical patterns that are hard to produce and recognize classically. Similarly, quantum information processors potentially have the ability to recognize patterns that are hard to recognize classically. Quantum computers can process information differently from classical computers on the basis of non-classical effects such as quantum coherence and entanglement. To this end, quantum machine learning encompasses the techniques of quantum software that could enable machine learning that is more powerful than what can be performed on classical computers; although its progress is currently constrained by hardware and software interaction limitations.

A *quantum algorithm* is a set of instructions to be performed on a quantum computer for solving a particular problem. In quantum machine learning, quantum algorithms are used as subroutines of larger implementations to classify and sample classically inaccessible systems. Speedups in machine learning are currently characterized as a function of measures from complexity theory: query complexity and gate complexity. *Query complexity* quantifies the queries to the information source for the algorithm. *Gate complexity* quantifies the number of elementary quantum operations (or gates) required to obtain the desired result. A quantum algorithm exhibits a *quantum speedup* by having lower query complexity, gate complexity, or both, in comparison to its classical counterpart.

Most quantum machine learning algorithms involve the loading of classical data into a quantum system, which remains an outstanding challenge. Theoretically, a qRAM (quantum Random Access Memory) uses n qubits to address any quantum superposition of $N = 2^n$ memory cells: its architecture exponentially reduces the requirements for a memory call by requiring entanglement among exponentially less gates [25]; yet no functioning implementations exist as of today. There is also hope on platforms such as quantum annealers and programmable quantum optical arrays, as specific-purpose quantum information processors that can realize deep learning architectures [1]. While fault tolerant universal quantum computers are still years away, there is still a growing interest in the community for quantum machine learning on near-term quantum devices.

Overview The main contributions of this thesis are

1. A concrete quantum algorithm for equivariant transformations (group convolutions and cross-correlations) on finite groups.

2. Setting the grounds to quantize the spherical CNNs – useful for rotationally equivariant data, i.e., data whose classification and analysis is invariant under $S^2 \to SO(3)$, a compact non-abelian group.

3. Discussing the prospects for quantum equivariant neural networks, and their conceptual role in the context of geometric unification of ML problems.

Structure This thesis is organized as follows. Chapter 2 presents the mathematical background through rigorous definitions and illustrations on group theory, representation theory, harmonic analysis and geometric deep learning. Equivariant neural networks are a subset of the broader topic of geometric deep learning, i.e., learning with data from the perspectives of symmetry and invariance. We also provide some background on quantum computing as well as existing quantum algorithms that will prove useful later on.

The body of the thesis begins in chapter 3, where we present our recent results on efficient quantum algorithms for performing linear finite group convolutions and cross-correlations on data stored as quantum states. Convolutions and cross-correlations provide a means to apply linear equivariant transformations. In chapter 4, we explore the case of infinite locally compact groups, such as SO(3). We present an overview of existing classical equivariant architectures; then, using the theoretical framework of [11], we propose a pathway to quantize spherical CNNs, SO(3) group cross-correlations with quantum oracles in the higher layers of a spherical CCNs – useful for rotationally equivariant data.

Then in chapter 5 –discussion and conclusions– we explore how our mathematical machinery can be a step towards the construction quantum equivariant neural networks. Furthermore, we discuss implementations: existing libraries of quantum CNNs, prospects of existing software to run QML programs and inherent limitations of QML. Finally, we provide a summary of the results and conclusions.

Chapter 2

Mathematical background

This thesis touches on group representation theory, quantum algorithms, harmonic analysis and geometric deep learning. In this chapter, we introduce some mathematical background that will be necessary for understanding the main results and conjectures.

2.1 Group theory

Equivariance, as the focus of study of this thesis, requires definition of groups and homogeneous spaces.

Definition 1 (group [20]). *A group $(G, \cdot)$ is a set endowed with an associative binary operation $G \times G \to G$, an identity element e, and where each element $g \in G$ has an inverse g^{-1} also in the set, which satisfies $gg^{-1} = e = g^{-1}g$.*

- *when $\cdot$ is commutative, G is* abelian
- *when the set is endowed with a topology where $\cdot$ and the inverse map are continuous, G is a* topological group
- *when such topology is compact, G is a* compact group
- *when G is a smooth manifold and $\cdot$ and the inverse map are smooth, G is a* Lie group

Examples. The following groups will be either illustrative of or relevant to our study.

1. The additive group of $\mathbb{Z}/n\mathbb{Z}$: the integers modulo n. Every finite cyclic group of order n is isomorphic to $\mathbb{Z}/n\mathbb{Z}$.

2. The dihedral group D_n: the group of symmetries of the regular n-gon in the plane. D_n consists of reflections and rotations by $2\pi/n$. The dihedral group D_n is of order $2n$ and is represented by $D_n = \mathbb{Z}/n\mathbb{Z} \rtimes \mathbb{Z}/2\mathbb{Z}$.

3. SO(3): the group of rotations in 3D, a compact, non-abelian Lie group.

Definition 2 (subgroup)**.** *A subgroup $(H, \cdot)$ of a group $(G, \cdot)$ is a group such that $H \subseteq G$ and is denoted by $H \leq G$.*

Definition 3 (coset space)**.** *Given a subgroup H and an element g of a group G, the* left coset *is defined as $gH = \{gh|h \in H\}$. The set of left cosets partitions G in the* left coset space *G/H. The definitions for right cosets Hg and the right coset space $H\backslash G$ are analogous.*

2.2 Representation theory

Group representation theory studies groups by how they act on vector spaces: elements of the group are represented as linear maps between vector spaces. Representations of a group translate the action of groups onto matrix operations. Group representations (i) represent actions on vector spaces, and (ii) form bases for spaces of functions on groups.

A group *homomorphism* between groups G and H is a map $f : G \to H$ such that $f(g_1g_2) = f(g_1)f(g_2)$. If f is a bijection (one-to-one and onto), then f is an *isomorphism.*

The group action on a space is a group homomorphism ϕ of a given group into the group of transformations of the space. The set of elements $x \in G$ such that $\varphi(x)$ is the identity is the *kernel* of φ, Ker φ. The kernel and image of a homomorphism φ are always subgroups, Ker $\varphi \leq G$ and Im $\varphi \leq G$.

Definition 4 (homogeneous space)**.** *The action of a group G is transitive on a space $\mathcal{X}$ if for any pair of elements $x, y \in \mathcal{X}$ there exists an element $g \in G$ s.t. $y = gx$. A* homogeneous space *$\mathcal{X}$ of a group G is a space where the group acts transitively.*

Definition 5 (representation [20])**.** *Let G be a group and $\mathcal{V}$ a vector space over some field. A* linear representation *is a group homomorphism $\rho : G \to GL(\mathcal{V})$, the general linear group* [1]*. If $\mathcal{V}$ is an inner product space and ρ is continuous and preserves the inner product, it is called a* unitary representation*.*

Example 1. Consider the multiplicative group $G = U(1)$ of numbers of the form $g_\theta = e^{i\theta}$. Let ρ be the map

$$\rho(e^{i\theta}) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta. \end{pmatrix} \tag{2.1}$$

ρ is a representation of G on $\mathbb{R}^2$. It is straightforward to verify that $g_{\theta_1} g_{\theta_2} = g_{\theta_1+\theta_2}$ and $\rho(e^{i(\theta_1+\theta_2)}) = \rho(e^{i\theta_1})\rho(e^{i\theta_2})$.

Example 2. [52] Consider the finite group Z_6. Its group action was modular arithmetic, which is not linear. There are multiple representations for cyclic groups. In this case, we can form a 2D vector representation through the space, using rotation matrices of $\frac{2\pi}{6}$. The kth representation is

$$\begin{pmatrix} \cos\frac{k2\pi}{6} & -\sin\frac{k2\pi}{6} \\ \sin\frac{k2\pi}{6} & \cos\frac{k2\pi}{6} \end{pmatrix}, \; k \in \{0, 1, 2, 3, 4, 5\} \tag{2.2}$$

It is straightforward to verify that this is a representation by checking that $r^k \cdot r^{6-k} = e$. The group action is done by repeated application to the point $(1, 0)$, which rotates around the circle.

A representation is unitary if $\rho(g)$ is a unitary matrix, i.e., $\rho(g)^{-1} = \rho^\dagger(g)$[2] for all g. A representation is irreducible if it contains no proper invariant subspaces with

[1] for finite dimensional $\mathcal{V}$ and $m = \dim \mathcal{V}$, $GL(\mathcal{X})$ is the space of $m \times m$ invertible matrices
[2] $\rho^\dagger(g)$ is the adjoing of the matrix = transpose and complex conjugate

respect to the action of the group. On the contrary, a representation is reducible if it decomposes as a direct sum of irreducible subrepresentations. We will often be interested in obtaining the irreducible representations of a group, i.e., decomposing a representation in its irreducible parts.

Definition 6 (irreducible representations)**.** *Let $\rho : G \to GL(\mathcal{V})$ be a representation of G on a vector space $\mathcal{V}$, and a vector subspace $\mathcal{W}$ of $\mathcal{V}$. When $\mathcal{W}$ is invariant under the action of G (i.e. $\rho(g)(w)\ \forall g \in G,\ w \in \mathcal{W}$), the restriction of ρ to $\mathcal{W}$ is a representation of G on $\mathcal{W}$, called a* subrepresentation*. We can call ρ an* irreducible representation *when the only subrepresentations of ρ are $\mathcal{V}$ itself and the zero vector space.*

The main idea is that decomposable unitary representations matrices can be formed upon smaller *irreducible* matrix blocks, i.e., $\rho_i(g)$,

$$\rho(g) = \mathbf{S}^{-1} \begin{pmatrix} \rho_0(g) & 0 & \dots & 0 \\ 0 & \rho_1(g) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \rho_k(g) \end{pmatrix} \mathbf{S}, \tag{2.3}$$

this notation works for a generic g. It satisfies that $\rho(g_1)\rho(g_2)$ gives back an element in $G - \rho(g')$, with the same block matrix structure. Also, in direct sum notation,

$$\rho(g) = \rho_0(g) \oplus \rho_1(g) \oplus \cdots \oplus \rho_k(g). \tag{2.4}$$

These irreducible representations form orthonormal basis functions analogous to basis-vectors from Hilbert spaces. The Peter-Weyl theorem [14] states that these irreducible representations can be arranged to form a complete basis-set for integrable L^2 functions.

We will take into account the following facts about irreducible representations (irreps) further down in the main work.

Theorem 7 (Maschke's theorem)**.** *Let V be a representation of the compact group*

G. If U is a subrepresentation of V, then there exists a subrepresentation W of V such that $V = U \oplus V$.

In particular, every representation of a finite group is a direct sum of irreps.

- Unirreps. Every finite-dimensional unitary representation of a compact group is a direct sum of unitary irreducible representations.
- The sum of the dimensions squared of all the irreducible representation of a group G equals group size G : $\sum_\sigma |d_\sigma|^2 = |G|$.

For abelian groups, the irreducible representations all have dimension equal to one. For non-abelian groups, there is at least one irreducible representation which has dimension greater than one. Remarkably, for finite groups, unitary irreducible representations always exist.

2.3 Theory of equivariance

Theoretically, a *symmetry* of a system is a transformation that leaves a certain property of such system or object unchanged. The symmetry on the set Ω underlying the space $\mathcal{X}(\Omega)$ will influence the structure of the functions defined on such space. To prevent further confusion for the reader, it is important to distinguish between the properties of *invariance* and *equivariance*, since both terms are ubiquitous in the related literature. A function $f : \mathcal{X}(\Omega) \to \mathcal{Y}$ is G-*invariant* if $f(\rho(g)x) = f(x)$ for all $g \in G$ and $x \in \mathcal{X}(\Omega)$, i.e., its output is unaffected by the group action on the input. A function $f : \mathcal{X}(\Omega) \to \mathcal{Y}$ is G-equivariant if $f(\rho(g)x) = \rho(g)f(x)$ for all $g \in G$, i.e., group action on the input affects the output on the same way.

In other words, a mapping $h(\cdot)$ is invariant to a set of transformations G if when we apply any transformation g to the input of h, the output remains unchanged by g. A mapping $h(\cdot)$ is equivariant to a set of transformations G if when we apply any transformation g to the input of h, the output is also transformed by g.

Definition 8 (Equivariance [32])**.** *Let G be a group and $\mathcal{X}_1$, $\mathcal{X}_2$ be two sets with corresponding G-actions*

$$T_g : \mathcal{X}_1 \to \mathcal{X}_1 \quad T'_g : \mathcal{X}_2 \to \mathcal{X}_2 \tag{2.5}$$

Let V_1 and V_2 be vector spaces with basis elements labeled by elements of $\mathcal{X}_1$ and $\mathcal{X}_2$ respectively, and let $L_{V_1}(L_{V_2})$ be the set of functions mapping $\mathcal{X}_1(\mathcal{X}_2)$ to $V_1(V_2)$. Let $\mathbb{T}$ and $\mathbb{T}'$ be the induced actions of group elements onto V_1 and V_2 respectively (e.g., permutations). A map $\phi : L_{V_1} \to L_{V_2}$ is equivariant if

$$\phi(\mathbb{T}_g(f)) = \mathbb{T}'_g(\phi(f)) \quad \forall f \in L_{V_1}. \tag{2.6}$$

A linear map is *equivariant* if it transfers symmetries from the function's input space into its output space. In other words, a map (for example, a neural network ψ) is equivariant if the result of $\psi[\mathbb{T}_g f(x)]$ – the network acting on the transformed input function, is equivalent to $\mathbb{T}_2\psi[f(x)]$ – the transform acting on the network output. The property of equivariance can be visualized as a commutative diagram:

$$\begin{tikzcd} L_{V_1} \arrow[r, "\mathbb{T}_g"] \arrow[d, "\phi"] & L_{V_1} \arrow[d, "\phi"] \\ L_{V_2} \arrow[r, "\mathbb{T}'_g"] & L_{V_2} \end{tikzcd} \tag{2.7}$$

Following this definition, CNNs are checked to be equivariant by comparing the results of transforming the input function and the output function. Refer to Appendix A: a network will be G-equivariant if the output looks the same by applying the rotation before or after.

2.4 Harmonic analysis

To begin our exploration of quantum algorithms for equivariance, we first look at harmonic analysis in the form of the Fourier transform. The quantum Fourier transform (QFT) is one of the principal algorithmic tool and sources of speedup underlying most efficient quantum algorithms.

Let us begin with the Fourier transforms over finite Abelian and non-Abelian groups. The definitions for the following two subsections are paraphrased from [8]. Some of the proofs of the main statements in this section are deferred to Appendix A.

2.4.1 Abelian Quantum Fourier Transform

Recall the group $\mathbb{Z}/n\mathbb{Z}$, the additive group of integers modulo N. The quantum Fourier transform, QFT for short, is a unitary operation $F_{\mathbb{Z}/n\mathbb{Z}}$.

Fourier transform over finite Abelian groups Let $\omega_N := e^{i2\pi/N}$ be the Nth root of unity. The action of the unitary $F_{\mathbb{Z}/n\mathbb{Z}}$ on a basis state $|x\rangle \in \mathbb{Z}/n\mathbb{Z}$ is

$$F_{\mathbb{Z}/n\mathbb{Z}} : |x\rangle \to \frac{1}{\sqrt{N}} \sum_{y \in F_{\mathbb{Z}/n\mathbb{Z}}} \omega_N^{x \cdot y} |y\rangle \,. \tag{2.8}$$

Following up with the last section that a finite Abelian group G has $|G|$ distinct one-dimensional irreducible representations $\rho \in \hat{G}$. Recall that such representations are functions $\rho : G \to \mathbb{C}$ such that $\rho(g_1 + g_2) = \rho(g_1)\rho(g_2)$ $g_1, g_2 \in G$, where additive notation is used for Abelian groups. The quantum Fourier transform (QFT) is defined as follows:

Quantum Fourier transform F_G over G

$$F_G : |x\rangle \to \frac{1}{\sqrt{|G|}} \sum_{\rho \in \hat{G}} \rho(x) |\rho\rangle \tag{2.9}$$

for each $x \in G$.

Let the *shift* operator P_s for $s \in G$, be defined as $P_s : |x\rangle \to |x + s\rangle$ for any $s \in G$. It can be shown that measurements in the Fourier basis produce the same statistics for pure states $|\psi\rangle$ as for its shift $P_s|\psi\rangle$,

$$F_G P_s F_G^\dagger = \sum_{\rho \in \hat{G}} \rho(s) |\rho\rangle \langle\rho| \tag{2.10}$$

i.e., a G-invariant mixed state is diagonalized by F_G.

Below, we will use the following

Theorem 9 (Fundamental theorem of finite abelian groups [18])**.** *Every finite abelian group G is a direct product of cyclic groups whose orders are prime powers uniquely determined by the group.*

$$G \cong (\mathbb{Z}/p_1^{r_1}\mathbb{Z}) \times \ldots \times (\mathbb{Z}/p_k^{r_k}\mathbb{Z}) \tag{2.11}$$

Remark. *As a consequence, it can be shown later on that the QFT over G can be decomposed as the tensor product of QFTs as*

$$F_{\mathbb{Z}/p_1^{r_1}\mathbb{Z}} \otimes \ldots \otimes F_{\mathbb{Z}/p_k^{r_k}\mathbb{Z}}. \tag{2.12}$$

The matrix representation of the Fourier transformation over $\mathbb{Z}/N\mathbb{Z}$ from the basis of states $\{|x\rangle : x \in G\}$ to the basis $\{|\rho\rangle : \rho \in \hat{G}\}$,

$$F_{\mathbb{Z}/N\mathbb{Z}} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega_N & \omega_N^2 & \ldots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \ldots & \omega_N^{2N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2N-2} & \ldots & \omega_N^{(N-1)(N-1)} \end{pmatrix} \tag{2.13}$$

expressing it in the basis state,

$$F_{\mathbb{Z}/N\mathbb{Z}} = \frac{1}{\sqrt{N}} \sum_{x,y \in \mathbb{Z}/N\mathbb{Z}} \omega_N^{x \cdot y} |y\rangle \langle x|, \tag{2.14}$$

$F_{\mathbb{Z}/N\mathbb{Z}}$ is a unitary transformation since $F_{\mathbb{Z}/N\mathbb{Z}} F_{\mathbb{Z}/N\mathbb{Z}}^\dagger = F_{\mathbb{Z}/N\mathbb{Z}}^\dagger F_{\mathbb{Z}/N\mathbb{Z}} = 1$.

Proof. Assume that $N = 2^n$, and represent the integer $x \in \mathbb{Z}/N\mathbb{Z}$ as $x_0, x_1, \ldots, x_{n-1}$ where $x = \sum_{j=0}^{n-1} 2^j x_j$. Let us rewrite the Fourier transform of $|x\rangle$ as

$$\begin{aligned}
F_{\mathbb{Z}/2^n\mathbb{Z}} |x\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \omega_{2^n}^{(\sum_{j=0}^{n-1} 2^j y_j)} |y_0, \ldots, y_{n-1}\rangle \\
&= \frac{1}{\sqrt{2^n}} \bigotimes_{j=0}^{n-1} \sum_{y_j \in \{0,1\}} e^{i2\pi x \cdot y_j / 2^{n-j}} |y_j\rangle \\
&= \bigotimes_{j=0}^{n-1} \frac{|0\rangle + e^{i2\pi \sum_{k=0}^{n-1} 2^{j+k-n} x_k} |1\rangle}{\sqrt{2}} \\
&=: \bigotimes_{j=0}^{n-1} |z_j\rangle .
\end{aligned} \tag{2.15}$$

thus the Fourier transform of $|x\rangle$ can in fact be written as the tensor product of n qubits. □

2.4.2 Non-Abelian Quantum Fourier Transform

We now turn to the non-Abelian Fourier transform.

The Fourier transform of the basis vector $|x\rangle$ corresponding to a given $x \in G$ is a weighted superposition over a complete set of irreps $\hat{G}$,

$$|\hat{x}\rangle = \frac{1}{\sqrt{|G|}} \sum_{\rho \in \hat{G}} d_\rho |\rho, \rho(x)\rangle , \tag{2.16}$$

where $|\rho(x)\rangle$ is a normalized state with entries given by the $d_\rho \times d_\rho$ matrix $\rho(x)/\sqrt{d_\rho}$,

$$\begin{aligned}
|\rho(x)\rangle &:= (\rho(x) \otimes I_{d_\rho}) \sum_{j=1}^{d_\rho} \frac{|j, j\rangle}{\sqrt{d_\rho}} \\
&= \sum_{j=1}^{d_\rho} \frac{\rho(x)_j, j}{\sqrt{d_\rho}} |j, k\rangle .
\end{aligned} \tag{2.17}$$

The quantum Fourier transform over G returns a state also composed by a weighted superposition over the irreps,

$$
\begin{aligned}
F_G &:= \sum_{x \in G} |\hat{x}\rangle \langle x| \\
&= \sum_{x \in G} \sum_{\rho \in \hat{G}} \sqrt{\frac{d_\rho}{|G|}} \sum_{j,k=1}^{d_\rho} \rho(x)_{j,k} |\rho, j, k\rangle \langle x| .
\end{aligned} \tag{2.18}
$$

F_G is a unitary transformation, and is also precisely the transformation that block -diagonalizes the left and right regular representations, simultaneously. For the left regular representation L of $G: L_i |j\rangle = |ij\rangle \ \forall\, i, j \in G$, we have that ([8], Eq. 123)

$$
\hat{L}_i = F_G L_i F_G^\dagger = \sum_{j \in G} |\hat{ij}\rangle \langle \hat{j}| = \bigoplus_{\rho \in \hat{G}} \big(\rho(i) \otimes I_{d_\rho}\big). \tag{2.19}
$$

Analogously, for the right regular representation R of $G: R_i |j\rangle = |ji^{-1}\rangle \ \forall\, i, j \in G$, yielding,

$$
\hat{R}_i = F_G R_i F_G^\dagger = \bigoplus_{\rho \in \hat{G}} \big(I_{d_\rho} \otimes \rho(i)^*\big). \tag{2.20}
$$

These identities will be key when convolving filters and inputs in the convolutions and cross correlations operations over non-abelian group which require matrix multiplication over irreps, and where diagonalization is non-trivial.

2.4.3 Generic Quantum Fourier Transform

The method of [40] has given rise to the first subexponential-size quantum circuits for the QFT over the linear groups $\mathrm{GL}_k(q)$, $\mathrm{SL}_k(q)$, the finite Lie groups, for any prime power q.

2.5 Geometric deep learning

Equivariant Neural Networks are part of a broader topic of geometric deep learning. The field encompasses all learning methods with data that has some underlying geometric relationships.

This symmetry-based effort at unification is reminiscent to the symmetry principles allowing to unify the fundamental forces of nature with the exception of gravity in the Standard Model. Conjectured by the "5G" paper – Grids, Groups, Graphs, Geodesics, and Gauges [4], the current state of deep learning is reminiscent of the situation of geometry in the XIXth century: Euclidean, projective, hyperbolic and elliptic geometry were finally unified by the breakthrough insight of Felix Klein to define geometry as the study of *invariants*, structures preserved under *symmetry* transformations.

Such a "geometric unification" endeavour in the spirit of the Erlangen Program serves the purpose of, on one hand, providing a common mathematical framework to study the most successful neural network architectures, such as CNNs, RNNs, GNNs, and Transformers. On the other, it gives a constructive procedure to incorporate prior physical knowledge into neural architectures and provide principled ways to build future architectures yet to be invented.

2.6 Quantum algorithms

Quantum computers achieve speedup over classical computers by exploiting the properties of interference and entanglement between quantum amplitudes. Quantum mechanics is known by the ability to represent a large number of amplitudes in a few *qubits* = quantum bits. A lot of quantum algorithms – including Shor's algorithm, obtain the exponential interference property leading to a *quantum speedup* orchestrated by the unitary operation of the *quantum Fourier Transform* (QFT), which is an algebraic operation.

The main motivation for this work derives from the study of quantum algorithms for algebraic problems and linear algebra. The focus of study of this work mainly arises from implementations of equivariant transformations in deep learning in the context of group convolutional neural networks [10]. While each section will present the particular pieces of related work that are directly related to each topic, we summarize here the current of quantum algorithms for related problems. We classify them in

(1) quantum algorithms for group theory, (2) quantum algorithms for linear algebra and finally, (3) classical results on Equivariant and Convolutional Neural Networks. Recall that the *quantum Fourier Transform* (QFT), which is an algebraic operation.

The exposition of these points is inspired by the exposition presented in [6].

2.6.1 Quantum algorithms in group theory

The pioneering efforts for solving group theoretic problems in quantum computing dates back to quantum algorithms aimed at solving the **hidden subgroup problem** [8]. The algorithm proposed by [50] for solving the hidden shift problem, employed group deconvolution on quantum states storing a superposition of queried function values. These ideas have been further developed in [45, 46]. For instance, an algorithm provided by [40], performs **generic group Fourier transforms** on a quantum computer which is a basis of interest for further transformations performed in this work. In the context of quantum circuit analysis, group convolution has been used to analyze rates of convergence of ensembles of unitaries [19, 13].

2.6.2 Quantum algorithms in linear algebra

Most of the core methods of this study are first formulated on algorithms for performing linear algebraic operations on a quantum computer. We employ methods for **block encoding** unitary operators [36, 24] and applying **linear combinations** of unitary matrices [33, 26] vert extensively in our algorithms.

Prior quantum algorithms have proposed efficient methods for performing **matrix multiplication** and solving **linear systems** of equations for dense matrices. The most related papers are those for applying circulant [3] or Toeplitz matrices [51, 55, 37]. From an applied perspective, some related work achieves tp pre-condition matrices using circulant matrices or implementing **Green's functions** by taking advantage of symmetries in a problem [49, 47].

[3] Circulant matrices are a specific instance of the more general form of group cross-correlation matrices studied here.

2.6.3 Equivariant and group convolutional neural networks

In the past few years, many algorithms for equivariant neural networks have been proposed and analyzed [32, 29, 10, 44]. These algorithms employ and analyze weight sharing schemes that are inherent in equivariant transformations. This work has motivated a long line of research aiming to take advantage of symmetries in data [54, 15, 53] with applications particularly in physics and chemistry [3, 48, 31, 9].

Many quantum algorithms have converted machine learning algorithms into quantum algorithms that are related to convolutions. For example, [28] constructs a quantum algorithm that mimics the operation of a classical convolutional neural network (e.g., for image recognition). Quantum versions of convolutional neural networks which parameterize convolutions as quantum gates have also been proposed [12, 34, 42].

Chapter 3

Quantum algorithm for finite group equivariant transformations

Preliminaries

Neural Networks In general, deep neural networks can be represented as a chain of operations W_i with optimized parameters, and interleaved with nonlinearities σ_i – e.g., the $\mathrm{ReLU}(x_i) = \max(x_i, 0)$,

$$f_{\mathrm{out}} = W_n(\ldots \sigma_2(W_2(\sigma_1(W_1 f_{\mathrm{in}}))) \ldots) \tag{3.1}$$

In Convolutional Neural Networks (CNNs), these operations are convolutions. Equivariant neural networks leverage properties of equivariance and symmetries of certains types of data to reduce model complexity. While there exist different types of network depending on the group, whether equivariance applies on the local or global transformations, and whether the feature maps are scalar or general fields [20]. In the very case of our study, we focus on finite groups where convolution is reduced to summing over elements of the group.

3.1 Introduction

Convolutions and cross-correlations are examples of functions which are equivariant to the actions of a group. In fact, [32] proves that a feedforward neural network layer is equivariant to the action of a group if and only if each layer of the neural network performs a generalized form of convolution or cross correlation. In this section we overview the group equivariant transformations of group convolution and cross-correlation and present our recent published work [6] on efficient quantum algorithms for these linear group operations on data stored as quantum states. For the remaining of this chapter, we restrict ourselves to finite groups where the exposition of representations and group Fourier transforms is simpler.

3.2 Preliminaries

Given functions f and g which map group elements $u \in G$ to complex or real numbers, a convolution over a group G is defined as

$$(f \circledast g)(u) = \sum_{v \in G} f(uv^{-1}) g(v). \tag{3.2}$$

Similarly, cross-correlation is defined as

$$(f \star g)(u) = \sum_{v \in G} f(vu^{-1}) g(v). \tag{3.3}$$

Convolutions and cross-correlations are linear operations, and thus they can be converted into matrix formulations.

3.2.1 Converting group convolution to a linear algebraic formulation

Given two functions that map group elements to complex numbers $m, x, : G \to \mathbb{C}$ – the filter and input, we vectorize these functions over group elements by associating every group element to a basis of the vector space. Denote these vectors of dimension $|G|$ as $\vec{m}$ (filter)and $\vec{x}$ (input). Any convolution or cross-correlation can be converted into a matrix weighted sum of the left and right regular representations L_u and R_u, respectively.

Note that, for convolution $(m \circledast x)(u)$

$$\begin{aligned}(m \circledast x)(u) &= \sum_{v \in G} m(uv^{-1})x(v) \\ &= \sum_{v \in G} m(v^{-1})x(vu) \\ &= \sum_{v \in G} m(v)x(v^{-1}u) \\ &= \sum_{v \in G} m(v) \left[L_v \vec{x}\right]_u ,\end{aligned} \tag{3.4}$$

where the notation $[\cdot]_i$ indicates the i-th component of the vector within the brackets. In the second and third lines above, we re-order the sum over all group elements by transforming $v \to vu$ and $v \to v^{-1}$ respectively. Converting the above into a vector form over the output, we have the final result:

$$\vec{m} \circledast \vec{x} = \sum_{i \in G} m_i L_i \vec{x} = M^{\circledast} \vec{x} \quad ; M^{\circledast} = \sum_{i \in G} m_i L_i \tag{3.5}$$

noting that the others require similar steps. The results are summarized in the following lemma.

Lemma 10 (Group operations as matrices)**.** *Given a group G, let $\vec{m} \in \mathbb{C}^{|G|}$ and $\vec{x} \in \mathbb{C}^{|G|}$ be the filter and input for a group operation. Then, group convolutions and cross-correlations correspond to matrix weighted sums of the left or right regular representations.*

$$
\begin{aligned}
(m \circledast x)(u) &= \sum_{v \in G} m(uv^{-1})x(v) &\overset{\text{convolution}}{\Longleftrightarrow}\quad & \vec{m} \circledast \vec{x} = M^{\circledast}\vec{x}, \quad M^{\circledast} = \sum_{i \in G} m_i L_i \\
(m \circledast_R x)(u) &= \sum_{v \in G} m(v^{-1}u)x(v) &\overset{\text{right convolution}}{\Longleftrightarrow}\quad & \vec{m} \circledast_R \vec{x} = M^{R\circledast}\vec{x}, \quad M^{R\circledast} = \sum_{i \in G} m_i R_i \\
(m \star x)(u) &= \sum_{v \in G} m(vu^{-1})x(v) &\overset{\text{cross-correlation}}{\Longleftrightarrow}\quad & \vec{m} \star \vec{x} = M^{\star}\vec{x}, \quad M^{\star} = \sum_{i \in G} m_i L_i^{-1} \\
(m \star_R x)(u) &= \sum_{v \in G} m(u^{-1}v)x(v) &\overset{\text{right cross-correlation}}{\Longleftrightarrow}\quad & \vec{m} \star_R \vec{x} = M^{R\star}\vec{x}, \quad M^{R\star} = \sum_{i \in G} m_i R_i^{-1}
\end{aligned}
\tag{3.6}
$$

For each of the operations above, we also have a corresponding convolution theorem which applies the operation in the Fourier domain of the group.

For standard convolution, we have:

$$
\begin{aligned}
(\widehat{m \circledast x})(\rho) &= \sum_{u \in G} \rho(u) \sum_{v \in G} m(uv^{-1})x(v) \\
&= \sum_{u \in G} \sum_{v \in G} \rho(u)\rho(v^{-1})\rho(v)m(uv^{-1})x(v) \\
&= \sum_{v \in G} \sum_{u \in G} m(uv^{-1})\rho(uv^{-1})x(v)\rho(v) \\
&= \sum_{v \in G} \left[\sum_{u \in G} m(uv^{-1})\rho(uv^{-1}) \right] x(v)\rho(v) \\
&= \sum_{v \in G} \hat{m}(\rho)x(v)\rho(v) \\
&= \hat{m}(\rho)\hat{x}(\rho).
\end{aligned}
\tag{3.7}
$$

Since $(m \circledast_R x)(u) = (x \circledast m)(u)$, then the above argument can also be applied to show that $(\widehat{m \circledast_R x})(\rho) = \hat{x}(\rho)\hat{m}(\rho)$.

For standard cross-correlation, we similarly can show that:

$$
\begin{aligned}
\widehat{(m \star x)}(\rho) &= \sum_{u \in G} \rho(u) \sum_{v \in G} m(vu^{-1}) x(v) \\
&= \sum_{u \in G} \sum_{v \in G} \rho(u) \rho(v^{-1}) \rho(v) m(vu^{-1}) x(v) \\
&= \sum_{v \in G} \sum_{u \in G} m(vu^{-1}) \rho(uv^{-1}) x(v) \rho(v) \\
&= \sum_{v \in G} \left[\sum_{u \in G} m(vu^{-1}) \rho(vu^{-1})^{\dagger} \right] x(v) \rho(v) \\
&= \sum_{v \in G} \hat{m}(\rho)^{\dagger} x(v) \rho(v) \\
&= \hat{m}(\rho)^{\dagger} \hat{x}(\rho).
\end{aligned} \tag{3.8}
$$

Since $(m \star_R x)(u) = (x \star m)(u)$, similarly the above argument can be applied to show that $\widehat{(m \circledast_R x)}(\rho) = \hat{x}(\rho)\hat{m}(\rho)^{\dagger}$.

Lemma 11 (Convolution theorems [23])**.** *Given a group G, let $\vec{m} \in \mathbb{C}^{|G|}$ and $\vec{x} \in \mathbb{C}^{|G|}$ be the filter and input for a group operation. Let $\hat{m}(\rho)$ and $\hat{x}(\rho)$ indicate the value of the Fourier transform of the filter and input for irreducible representation ρ. Then, one can perform group operations in the Fourier regime by applying the corresponding convolution theorem.*

$$
\begin{aligned}
(m \circledast x)(u) &= \sum_{v \in G} m(uv^{-1}) x(v) & &\overset{\text{convolution}}{\Longleftrightarrow} & \widehat{(m \circledast x)}(\rho) &= \hat{m}(\rho)\hat{x}(\rho) \\
(m \circledast_R x)(u) &= \sum_{v \in G} m(v^{-1}u) x(v) & &\overset{\text{right convolution}}{\Longleftrightarrow} & \widehat{(m \circledast_R x)}(\rho) &= \hat{x}(\rho)\hat{m}(\rho) \\
(m \star x)(u) &= \sum_{v \in G} m(vu^{-1}) x(v) & &\overset{\text{cross-correlation}}{\Longleftrightarrow} & \widehat{(m \star x)}(\rho) &= \hat{m}(\rho)^{\dagger}\hat{x}(\rho) \\
(m \star_R x)(u) &= \sum_{v \in G} m(u^{-1}v) x(v) & &\overset{\text{right cross-correlation}}{\Longleftrightarrow} & \widehat{(m \star_R x)}(\rho) &= \hat{x}(\rho)\hat{m}(\rho)^{\dagger}
\end{aligned} \tag{3.9}
$$

The reader can find examples of these concepts and the application of the convolution theorems on Appendix B.

3.2.2 Equivariance

Here, we show explicitly that convolution and cross-correlation are equivariant actions. Let G be a group and $\mathcal{X}_1, \mathcal{X}_2$ be two sets with corresponding G-actions

$$T_g : \mathcal{X}_1 \to \mathcal{X}_1 \qquad T'_g : \mathcal{X}_2 \to \mathcal{X}_2.$$

Let V_1 and V_2 be vector spaces with basis elements labeled by elements of $\mathcal{X}_1$ and $\mathcal{X}_2$, respectively, and let $L_{V_1}, (L_{V_2})$ be the set of functions mapping $\mathcal{X}_1(\mathcal{X}_2)$ to $V_1(V_2)$.

Convolution. Let $\phi_m : L_{V_1} \to L_{V_2}$ be the map performing convolution with a fixed filter $\vec{m}$ on an input $\vec{f}$,

$$\phi_m(\vec{f}) = \vec{m} \circledast \vec{f}.$$

Let T_g, T'_g denote the right actions of the group,

$$T_g, T'_g : u \to ug, \tag{3.10}$$

and let $\mathbb{T}_g$ and $\mathbb{T}'_g$ be the induced actions of group elements onto V_1 and V_2 respectively. From definition 1.1, the map $\phi_m : L_{V_1} \to L_{V_2}$ is *equivariant* to the action of T_g since

$$\phi_m(\mathbb{T}_g\vec{f}) = \mathbb{T}'_g(\phi_m(\vec{f})).$$

Proof. Recall the convolution definition from 3.9

$$\left[\phi_m(\vec{f})\right]_u = \left[\vec{m} \circledast \vec{f}\right]_u = \sum_{v \in G} m(uv^{-1}) f(v)$$

Let ϕ_m act on $\mathbb{T}_g\vec{f}$,

$$\begin{aligned} \left[\phi_m(\mathbb{T}_g\vec{f})\right]_u = \left[\vec{m} \circledast \mathbb{T}_g\vec{f}\right]_u &= \sum_{v \in G} m(uv^{-1}) \left[\mathbb{T}_g\vec{f}\right]_v \\ &= \sum_{v \in G} m(uv^{-1}) f(vg) \end{aligned}$$

Now redefine the sum above over $v' = vg$, and now we have $v^{-1} = g(v')^{-1}$,

$$
\begin{aligned}
\sum_{v \in G} m(uv^{-1}) f(vg) &= \sum_{v' \in G} m(ugv'^{-1}) f(v') \\
&= \left[\vec{m} \circledast \vec{f}\right]_{ug} \\
&= \left[\phi_m(\vec{f})\right]_{ug} \\
&= \left[\mathbb{T}'_g(\phi_m(\vec{f}))\right]_u \qquad \forall\, u \in G.
\end{aligned}
\tag{3.11}
$$

We conclude that $\phi_m(\mathbb{T}_g \vec{f}) = \mathbb{T}'_g(\phi_m(\vec{f}))$; hence convolution is equivariant to the right actions of the group. □

Cross-correlation. Let $\phi_m : L_{V_1} \to L_{V_2}$ be the map performing cross-correlation with a fixed filter $\vec{m}$ on an input $\vec{f}$,

$$\phi_m(\vec{f}) = \vec{m} \star \vec{f}.$$

Let T_g and T'_g denote the right actions of the group,

$$T_g, T'_g : u \to ug$$

Let $\mathbb{T}_g$ and $\mathbb{T}'_g$ be the induced action of group elements onto V_1 and V_2. From definition 1.1, the map $\phi_m : L_{V_1} \to L_{V_2}$ is equivariant to T_g since

$$\phi_m(\mathbb{T}_g \vec{f}) = \mathbb{T}'_g(\phi_m(\vec{f})).$$

Proof. Recall the cross correlation definition from 3.9

$$\left[\phi_m(\vec{f})\right]_u = \left[\vec{m} \star \vec{f}\right]_u = \sum_{v \in G} m(vu^{-1}) f(v).$$

Let ϕ_m act on $\mathbb{T}_g\vec{f}$,

$$\begin{aligned}\left[\phi_m(\mathbb{T}_g\vec{f})\right]_u = \left[\vec{m}\star\mathbb{T}_g\vec{f}\right]_u &= \sum_{v\in G} m(uv)\left[\mathbb{T}_g\vec{f}\right]_v \\ &= \sum_{v\in G} m(vu^{-1})f(vg)\end{aligned}$$

Now redefine the sum above over $v' = vg$, and now we have $v = v'g^{-1}$,

$$\begin{aligned}\sum_{v\in G} m(vu^{-1})f(vg) &= \sum_{v'\in G} m(v'g^{-1}u^{-1})f(v') \\ &= \left[\vec{m}\star\vec{f}\right]_{ug} \\ &= \left[\mathbb{T}'_g(\phi_m(\vec{f}))\right]_u \qquad \forall u\in G.\end{aligned}$$

Concluding that $\phi_m(\mathbb{T}_g\vec{f}) = \mathbb{T}'_g(\phi_m(\vec{f}))$; hence cross-correlation is equivariant to the right actions of the group. □

We observe in this subsections that there are two methods for performing group convolution or cross-correlation on a quantum computer. The first is as a weighted sum of matrices, as described in Lemma 10. The second is to perform Fourier transforms to inputs, perform convolution in the Fourier regime through multiplication as described in Lemma 11. The second method – the method of our choice in the quantum algorithm, leverages the advantages of group Fourier transforms, which are efficiently performable on a quantum computer [8, 40].

3.2.3 Block encodings

Throughout this study, we employ the block encoding framework to implement linear transformations on a quantum computer [36]. In this framework, a desired linear but not necessarily unitary transformation $A \in \mathbb{C}^{2^w\times 2^w}$ bounded in the spectral norm by $\|A\| \leq 1$ is encoded in a unitary operator $U \in \mathbb{C}^{2^{(w+a)}\times 2^{(w+a)}}$ with a ancilla qubits such that the top left block of U is precisely A.

$$U = \begin{pmatrix} A & \cdot \\ \cdot & \cdot \end{pmatrix}, \qquad \left(\langle 0^a| \otimes I_w\right) U \left(|0^a\rangle \otimes I_w\right) = A, \tag{3.12}$$

where I_w is the identity operation on the w qubits encoding A, i.e., applying the unitary U to a quantum state $|0^a\rangle |\psi\rangle$ and post-selecting on the measurement outcome $|0^a\rangle$ on the ancilla qubits is equivalent to applying the operation A on $|\psi\rangle$. We can write

$$U |0^a\rangle |\psi\rangle = |0^a\rangle A |\psi\rangle + |\text{garbage}\rangle , \tag{3.13}$$

where $|\text{garbage}\rangle$ is the "remaining" state that is orthogonal to the subspace $|0^a\rangle$ (i.e., $[\langle 0^a| \otimes I_w] |\text{garbage}\rangle = 0$). The probability of successfully post-selecting $|0\rangle^a$ is equal to $\| A |\psi\rangle \|_2^2$.

3.2.4 Quantum implementation as sum of unitaries

Let $w = \lceil \log_2 |G| \rceil$ indicate the number of qubits needed to block encode a given group operation. In the quantum case, we assume that we have access to either of the below oracles, $\mathcal{A}_m$ or O_m, as well as their inverses, which provide values of the convolution filter as b-bit descriptions or amplitudes of a quantum state:

$$\begin{aligned} \mathcal{A}_m &: |0^w\rangle \to \frac{1}{\sqrt{\|\vec{m}\|_1}} \sum_{i \in G} \sqrt{|m_i|}\, |i\rangle , \\ O_m &: |i\rangle |0^b\rangle \to |i\rangle |m_i\rangle . \end{aligned} \tag{3.14}$$

Digital to analog oracle conversion

Given oracle O_m, our goal is to construct $\mathcal{A}_m$. Here, we assume that O_m returns values normalized such that the magnitude of the maximum value of m_i is equal to 1. This is chosen to maximize the success probability of oracle conversion which can be performed by following the steps below.

1. Beginning with the state $|0^w\rangle |0^b\rangle$, obtain an equal superposition of states in the

support of m.

$$|0^w\rangle\,|0^b\rangle \rightarrow \frac{1}{\sqrt{|\operatorname{supp}(m)|}} \sum_{i \in \operatorname{supp}(m)} |i\rangle\,|0^b\rangle\,, \tag{3.15}$$

where supp(m) returns the set of basis states in the support of m. If the filter m has full support, then this is equivalent to applying Hadamard gates to each qubit.

2. Call oracle O_m and perform (classical) transformations to obtain the magnitude of the filter resulting in

$$\frac{1}{\sqrt{|\operatorname{supp}(m)|}} \sum_{i \in \operatorname{supp}(m)} O_m\,|i\rangle\,|0^b\rangle \rightarrow \frac{1}{\sqrt{|\operatorname{supp}(m)|}} \sum_{i \in \operatorname{supp}(m)} |i\rangle\,||m_i|\rangle\,. \tag{3.16}$$

3. Append a qubit and conditionally rotate the qubit by $\sqrt{|m_i|}$.

$$\begin{aligned} &\frac{1}{\sqrt{|\operatorname{supp}(m)|}} \sum_{i \in \operatorname{supp}(m)} |i\rangle\,|m_i\rangle\,|0\rangle \rightarrow \\ &\frac{1}{\sqrt{|\operatorname{supp}(m)|}} \sum_{i \in \operatorname{supp}(m)} |i\rangle\,|m_i\rangle \left(\sqrt{|m_i|}\,|0\rangle + \sqrt{1-|m_i|}\,|1\rangle\right). \end{aligned} \tag{3.17}$$

4. Measuring the last appended register, the oracle conversion is successful when the outcome of the measurement is $|0\rangle$. We note, that this register need not be measured right away and can be included in the block encoding to be measured later.

The runtime of this procedure depends on the probability of successfully measuring the $|0\rangle$ state in the last step. This probability is $|\operatorname{supp}(m)|^{-1} \sum_{i \in \operatorname{supp}(m)} |m_i|$ and is equal to the average value of $|m_i|$. If values of m_i are $\Theta(1)$ and do not decay with the dimension of the group, then this success probability is also $\Omega(1)$. Finally, additional gates are needed to obtain an equal superposition over states in the support of m as in step 1. This, in most cases, requires a number of operations that scale poly-logarithmically with the dimension of the state. For example, for filters with support over all states, this is equivalent to applying Hadamard gates to each qubit.

Lemma 12 (Linear combination of unitaries, paraphrased from Lemma 2.1 of [33])**.** *Let $V = \sum_i a_i U_i$ be a linear combination of unitary matrices U_i with $a_i > 0$. Let A be a unitary matrix that maps $|0^w\rangle$ to $\frac{1}{\sqrt{a}} \sum_i \sqrt{a_i} |i\rangle$ where $a := \sum_i a_i$. Let $U := \sum_i |i\rangle \langle i| \otimes U_i$, then $W := A^\dagger U A$ satisfies for any state $|\psi\rangle$*

$$W |0^w\rangle |\psi\rangle = \sqrt{p} |0^w\rangle V |\psi\rangle + |\Psi_\perp\rangle, \tag{3.18}$$

where $p = a^{-2}$ and the unnormalized state $|\Psi_\perp\rangle$ (depending on $|\psi\rangle$) satisfies $(|0^w\rangle \langle 0^w| \otimes I) |\Psi_\perp\rangle = 0$. In other words, W is a block encoding of the matrix V [24].

Lemma 13. *(Block encoding of group convolution or cross-correlation) Given oracle access to a filter $\vec{m}$, where $\|\vec{m}\| = 1$, we can block encode the matrix $M = \sum_i m_i U_i$ corresponding to group convolution or cross-correlation (Lemma 10 outlines corresponding choice of $U_i \in \{L_i, R_i + \text{inverses}\}$). This is achieved by two calls to the oracle $\mathcal{A}_m$, one call to the oracle O_m, and efficient (classical) circuits for permutations based on group operations U_i.*

Note. The normalization $\|\vec{m}\| = 1$ is required to ensure that the largest singular value of the linear operation is bounded by 1. This can be derived via the triangle inequality, e.g., for convolution

$$\|M^{\circledast}\| = \| \sum_{i \in G} m_i L_i \| \leq \sum_{i \in G} |m_i| = \|\vec{m}\|. \tag{3.19}$$

This allows for block encoding a matrix within a larger unitary matrix.

Proof. We choose $U_i \in \{L_i, R_i + \text{inverses}\}$ accordingly to the desired operation, such permutations can be performed efficiently on a classical circuit. We make a call to O_m and apply a phase transformation to U_i proportional to the phase of m_i. Finally, we apply the results of Lemma 12: setting A to $\mathcal{A}_m$ and the corresponding U_i. □

We can combine the facts derived above to apply group operations to an input state, stated below,

Proposition 14 (Applying group operations to an input state). *Given an input state $|x\rangle = \sum_i x_i |i\rangle$ containing the input state $\vec{x}$ normalized such that $\|\vec{x}\|_2 = 1$ and oracle access to the convolution filter $\vec{m}$, we can construct a state $|m \circ x\rangle$, equal to the normalized output of $\vec{m} \circ \vec{x}$, where $\circ$ corresponds to the four equivariant operations. The runtime scales as $O(T_B \|\vec{m} \circ \vec{x}\|_2^{-1})$ where T_B is the runtime of the block encoding of Lemma 13.*

Proof. Let matrix M correspond to the linear operator where $\vec{m} \circ \vec{x} = M\vec{x}$ for a given group operation. From Lemma 12, we get the state

$$|0^w\rangle M |x\rangle + |x_\perp\rangle \tag{3.20}$$

where $|x_\perp\rangle$ is the "garbage" projected state. The probability of success – measuring the first register and obtaining $|0\rangle$, is equal to $\|\vec{m} \circ \vec{x}\|_2^2$. □

Runtimes of these operations are constrained by the term $\|\vec{m} \circ \vec{x}\|$ to be small, which is bounded by the condition number of the corresponding matrix. These matrices are diagonalized for abelian groups, and block-diagonalized for non-abelian groups by the group Fourier transform as discussed in the next section. These results can be used to derive the condition number of any given linear operation.

If the group G is a cyclic gorup, then the cross-correlation operation over the group produces a circulant [1] matrix. As of now, there are previous quantum algorithms for performing matrix operations on circulant matrices [55, 51].

3.3 Quantum implementations via Convolution Theorems

There is an important distinction between abelian and non-abelian groups that arises from the property of the irreducible representations of each class of groups, which stands out when performing group operations in the Fourier regime. Abelian groups

[1] a circulant matrix is an $n \times n$ matrix whose rows are composed of cyclically shifter versions of a list of length n. They have applications on digital image processing.

have the nice property that all of their irreducible representations are scalars. Furthermore, the Fourier transform for an abelian group can be easily obtained given the fact that any finite abelian group is a direct product of cyclic groups per the following theorem.

Theorem 15 (Fundamental theorem of finite abelian groups [18])**.** *Every finite abelian group is a direct product of cyclic groups whose orders are prime powers uniquely determined by the group.*

Given this convenient theorem, the algorithm for performing abelian group operations is rather simple and we consider that case first. Then, we will generalize to the case of non-abelian groups which requires more detail.

3.3.1 Block encoding for abelian groups

Based on the fundamental theorem of finite abelian groups, one can form the Fourier transform for a finite abelian group by taking tensor products over the corresponding Fourier transform (DFT matrix) for the groups in the direct product. For example, if an abelian group G is isomorphic to k cyclic groups of dimension d_i respectively, then

$$F_G = \bigotimes_{i=1}^{k} F_{d_i} \qquad \text{(abelian groups)}, \tag{3.21}$$

where F_d is the discrete Fourier transform matrix of dimension d. This provides a direct means for diagonalizing convolutions and cross-correlations. For example, for convolution over an abelian group, we can form a matrix with the corresponding eigenvalues and eigenvectors.

$$\begin{aligned} F_G M^{\circledast} \vec{x} &= \sqrt{|G|} (F_G \vec{m}) \odot (F_G \vec{x}) \\ &= \sqrt{|G|} \operatorname{diag}(F_G \vec{m}) F_G \vec{x}, \end{aligned} \tag{3.22}$$

where the $\odot$ is entry-wise multiplication. This implies that

$$M^{\circledast} = F_G^{\dagger}\,\mathrm{diag}(\sqrt{|G|}F_G m)F_G, \tag{3.23}$$

where the eigenvalues of $M^{\circledast}$ are the entries of $\sqrt{|G|}\ F_G m$ and the eigenvectors are the columns of F_G. Note, that in the above, we assume the F_G are normalized to be unitary and hence we have the additional factor of $\sqrt{|G|}$ not typically seen in the convolution theorem. Since outputs are quantum states, this additional factor will be removed due to the normalization of the state.

Assume we are given access to an oracle $O_{\mathcal{F}m}$ which returns entries of $\hat{m}_i = \mathrm{diag}(\sqrt{|G|}F_G m)_{ii}$ in a separate register:

$$O_{\mathcal{F}m} : |i\rangle\,|0^b\rangle \to |i\rangle\,|\hat{m}_i\rangle\,. \tag{3.24}$$

This oracle can be efficiently constructed if the entries m_i are efficiently computable with a classical circuit, e.g. when m is sparse or when the group Fourier transform can be analytically computed.

Any finite abelian group G of size n is isomorphic to a direct product of c cyclic groups of dimension $n_1, \ldots, n_c$. Therefore, the Fourier transform for a finite abelian group is simply $F_G = F_{n_1} \otimes \cdots \otimes F_{n_c}$ where F_m is the standard unitary discrete Fourier transform matrix of dimension m. To apply a convolution matrix, we need to apply the Fourier transform, a diagonal matrix, and an inverse Fourier transform (see 3.23). We use a block encoding to perform the diagonal matrix operation $\mathrm{diag}(\sqrt{|G|}F_G m)$ as below.

Lemma 16 (Block encoding of diagonal matrix [55])**.** *Let $A \in \mathbb{C}^{2^w \times 2^w}$ be a diagonal matrix and each entry of A has entries ≤ 1. Given access to the oracle O_A such that,*

$$O_A : |i\rangle\,|0^b\rangle \to |i\rangle\,|A_{ii}\rangle\,, \tag{3.25}$$

where A_{ii} is a binary description of the ith diagonal element on b bits. One can implement a unitary block encoding U such that $\|A - (\langle 0^{w+3}| \otimes I)U(|0^{w+3}\rangle \otimes I)\| \leq \epsilon$

with $O(\mathrm{polylog}\frac{1}{\epsilon} + w)$ gates and two calls to O_A.

Furthermore, the block encoding for a given group operation is a direct application of the Fourier transforms and Lemma 16.

Lemma 17 (Fourier block encoding of abelian convolution or cross-correlation)**.** *For an abelian grouo G, let $w = \lceil \log_2(|G|) \rceil$. Assume we are given oracle access $O_{\mathcal{F}_m}$ to the convolution filter $\hat{m}$ in the Fourier regime. Assume the filter is normalized such that $|\hat{m}(\rho)| \leq 1$ for all entries. One can obtain the block encoding of the group operation in the shape of the unitary operator U, e.g., for convolution $\|M^{\circledast} - (\langle 0^{w+3}| \otimes I) U (|0^{w+3}\rangle \otimes I)\| \leq \epsilon$, with $O(\mathrm{polylog}\frac{1}{\epsilon} + w)$ additional gates and application of the (inverse) group Fourier transform, and two calls to the oracle $O_{\mathcal{F}_m}$.*

Proof. Look at 3.23: we implement F_G and $F_G^{\dagger}$ through the proper quantum Fourier transforms for the corresponding dimensions of the group. For the diagonal matrix multiplication of $A = \mathrm{diag}\sqrt{|G|}F_G\vec{m})$, we block encode A into U using oracle $O_{\mathcal{F}_m}$ and Lemma 16. $\square$

3.3.2 Block encoding for non-abelian groups

Unlike the abelian case, irreducible representations of non-abelian groups are matrices, and convolution applied in the Fourier regime requires matrix multiplication over the irreducible representation. In this setting, we now assume oracle access to $O_{\mathcal{F}m}$ which provides matrix entries of the Fourier transform of a convolution filter in a given irreducible representation,

$$O_{\mathcal{F}m} : |\rho, a, b\rangle\, |0\rangle \to |\rho, a, b\rangle\, |\hat{m}(\rho)_{ab}\rangle\,, \tag{3.26}$$

where $\rho \in \hat{G}$ indexes the irreducible representations and $\hat{m}(\rho)_{ab}$ is the a, b-th entry of the matrix $\hat{m}(\rho)$.

Quantum algorithms efficiently perform group Fourier transforms over many non-abelian groups (*e.g.,* dihedral and symmetric groups) [40, 8]. The quantum group Fourier transform for a group G returns a state containing a weighted superposition over irreducible representations [8]:

$$\begin{aligned} F_G &= \sum_{x\in G} |\hat{x}\rangle \langle x| \\ &= \sum_{x\in G}\sum_{\rho\in\hat{G}} \sqrt{\frac{d_\rho}{|G|}} \sum_{j,k=1}^{d_\rho} \rho(x)_{j,k} |\rho, j, k\rangle \langle x| , \end{aligned} \tag{3.27}$$

where $|\hat{x}\rangle$ is the group Fourier transform of a given basis vector $|x\rangle$ and $\hat{G}$ is the set of irreducible representations of G and the factor $\sqrt{d_\rho/|G|}$ enforces F_G to be unitary. As discussed earlier on in chapter 2, F_G has the ability of block diagonalizing the left and right regular representations into the irreps, e.g., for the left regular representation [8],

$$\hat{L}_i = \sum_{j\in G} \langle \hat{ij}|\, \hat{j} = F_G L_i F_G^\dagger = \bigoplus_{\rho\in G} \rho(i) \otimes I_{d_\rho} \tag{3.28}$$

.

Over non-abelian groups, convolutions and cross-correlations require matrix multiplication over irreps, thus we cannot diagonalize the state above as in the abelian case. Instead, to convolve $\vec{m}$ with $\vec{x}$, we need to apply a matrix in this form,

$$\vec{m} \circledast \vec{x} = F_G^{-1} \left[\bigoplus_{\rho\in\hat{G}} \hat{m}(\rho) \otimes I_{d_\rho} \right] F_G \vec{x}, \tag{3.29}$$

where $\hat{m}(\rho)$ is the Fourier transformed matrix for irrep ρ with dimensionality d_ρ.

Lemma 18 (Fourier block encoding of general group convolution or cross-correlation)**.** *For a group G with irreducible representations of dimension $\leq d_{\max}$, let $w = \lceil \log_2(|G|) \rceil$. Assume oracle access $O_{\mathcal{F}_m}$ to the convolution filter $\hat{m}$ in the Fourier regime, where the filter $\hat{m}$ is normalized such that $|]\hat{m}(\rho)_{ab}| \leq 1 \;\forall\; a, b$. We can obtain a unitary operator U that is a block encoding of the group operation. For group convolution, $\|M^\circledast - d_{\max}(\langle 0^{w+3}|\otimes I)U(|0^{w+3}\rangle \otimes I)\| \leq \epsilon$; with one application of the (inverse) group Fourier transform, two calls to the oracle $O_{\mathcal{F}_m}$, and $O(\mathrm{polylog}\frac{d_{\max}}{\epsilon} + w)$ additional gates.*

Proof. Let us consider the example of group convolution, one must perform the following three operations

$$F_G^\dagger \left[\bigoplus_{\rho \in \hat{G}} \hat{m}(\rho) \otimes I_{d_\rho} \right] F_G. \tag{3.30}$$

Let $|G| \leq 2^w$ so we can encode the data in w qubits. To form the block encoding we follow methods in [24]. For our block encoding, we construct a data register of w qubits and ancillary registers of $w+3$ qubits where the $|0\rangle$ measurement in this register corresponds to the location of the block encoding. We first apply the group Fourier transform to the data register. The middle operation $\left[\bigoplus_{\rho \in \hat{G}} \hat{m}(\rho) \otimes I_{d_\rho}\right]$ is a block diagonal matrix which we block encode using Lemma 48 of [24]. Each row or column of the matrix is at most $d_{\max}$ sparse. Note, this lemma also requires oracles that provide the locations of each sparse entry in a given row or column of the matrix; in our case, since matrices are block diagonal, locating these entries is easy. Applying this operation up to error ϵ in operator norm requires two calls to the oracle $O_{\mathcal{F}m}$ and $O(\operatorname{poly}\log \frac{d_{\max}}{\epsilon})$ additional gates [24]. Finally, one applies an inverse group Fourier transform $F_G^\dagger$ to the data register to obtain the given encoding. □

Remark. $d_{\max}$ *corresponds to the maximum sparsity of any row or column of the block diagonal matrix in our block encoding. The number of irreducible representations of a group is equal to the number of conjugacy classes of the group, so groups with many conjugacy classes tend to have lower dimensional irreducible representations. For all abelian groups,* $d_{\max}$ *is trivially equal to* 1*. For many non-abelian groups,* $d_{\max}$ *is also strictly bounded, e.g.,* $d_{\max} = 2$ *for dihedral groups* D_{2n} *for all* n *[8].*

3.3.3 Linear group operations on quantum states

With the block encodings described above, we can apply linear group operations to an input state $|x\rangle$ and leverage the runtime benefits of the quantum group Fourier transform to efficiently perform linear group operations. First, we show how to perform group convolution directly on an input state.

Proposition 19 (Applying group convolution to $|x\rangle$)**.** *For a group* G *with irreducible representations of dimension* $\leq d_{\max}$*, let* $w = \lceil \log_2(|G|) \rceil$*. Assume oracle access* $O_{\mathcal{F}_m}$

to the convolution filter $\hat{m}$ in the Fourier regime, where the filter $\hat{m}$ is normalized such that $|]\hat{m}(\rho)_{ab}| \leq 1 \;\forall\; a, b$. Given a quantum state $|x\rangle = \sum_i x_i |i\rangle$ with the input state $\vec{x}$ such that $\|\vec{x}\|_2 = 1$, we can construct a state $|\bar{y}\rangle$ such that

$$\| \, |\bar{y}\rangle - |\vec{m} \circ \vec{x}\rangle \, \| \leq \epsilon \tag{3.31}$$

i.e., $|\bar{y}\rangle$ is ϵ-close to the true normalized desired output. The runtime of this operation scales as $O(T_B \kappa d_{\max}/\|M\|)$ for T_B runtime of the block encoding of Lemma 16 or Lemma 18 and κ is the condition number of M.

Proof. We first apply the general block encoding to a state with the data encoded in the register. The measurement is successful when ancillary registers are measured in the $|0\rangle$ basis. The least of singular values of the linear operation is $\|M\|/\kappa$ and the block encoding has a normalization factor of $d_{\max}$. The worst-case success probability is of $\left(\kappa \frac{d_{\max}}{\|M\|}\right)^{-1}$. $\square$

Definition 20 (Condition number)**.** *The condition number κ can be calculated by analyzing the norms of the diagonal or block diagonal matrices in the block encoding. For example, for abelian groups,*

$$\kappa = \frac{\max_{\rho \in \hat{G}} |\hat{m}(\rho)|}{\min_{\rho \in \hat{G}} |\hat{m}(\rho)|}. \tag{3.32}$$

For non-abelian groups, we analyze the singular values of the Fourier transform over its irreducible representations. Let $s_{min}(M)$ and $s_{max}(M)$ be the smallest and largest singular values of a matrix M, then for non-abelian groups,

$$\kappa = \frac{\max_{\rho \in \hat{G}} s_{\max}(\hat{m}(\rho))}{\min_{\rho \in \hat{G}} s_{\min}(\hat{m}(\rho))}. \tag{3.33}$$

3.3.4 Inverse group operations: deconvolution

It has been shown [24] that given group operations in the format of block encodings, we can efficiently perform polynomial transformations to the singular values of the block encoded matrix. We can use this fact to apply inverse convolutions or cross-correlations, i.e., *deconvolution.*

Problem setup Given the state $|y\rangle$ containing the output of $\vec{y} = \vec{m} \circ \vec{x}$, where $\circ$ corresponds to convolution or cross-correlation. One would like to reconstruct the input $\vec{x}$ to the group operation as a quantum state $|x\rangle$.

In the proposition below, we provide an algorithm for deconvolution given the block encoding using oracle O_m from Lemma 13.

Proposition 21 (Deconvolution)**.** *For a group G, assume we are given oracle access O_m for the convoluion filter m. Given a quantum state $|y\rangle$, containing $\vec{y} = \vec{m} \circ \vec{x}$, such that $\|\vec{y}\|_2 = 1$, one can construct a state $|\bar{x}\rangle$, ϵ-close to the true normalized input $\vec{x}$, and $\circ$ corresponds to any of the four equivariant operations 10. Deconvolution as of this algorithm has a runtime of $O\big(T_B \frac{\kappa^2}{\|M\|\text{polylog}} \frac{\kappa^2}{\|M\|_\epsilon}\big)$ where T_B is the runtime of the block encoding of Lemma 13 and κ is the condition number of the linear group operator M.*

On the other hand, when performing deconvolution in the Fourier regime using the block encoding of Lemma 18, the runtime of the operation scales as $O\big(\frac{T_B}{d_{\max}} \frac{\kappa^2}{\|M\|\text{polylog}} \frac{\kappa^2}{\|M\|_\epsilon}\big)$.

See [6] for proofs and the formal statement.

3.4 Discussion

Our results allows us to structure a framework and methodology for performing group convolutions and cross-correlations on a quantum computer. In well-conditioned cases, the runtimes of the equivariant operations scale logarithmically with the dimension of the group. Our algorithms output quantum states storing the vectorial output of the operations, which can be later post-processed or analyzed through various schemes, e.g. [35, 30, 39, 27].

As mentioned earlier on in the chapter, it has been shown in machine learning literature that group equivariant neural networks can be decomposed into layers of group convolutions followed by nonlinear activation functions [32].

Though classical algorithms exist for approximately performing equivariant transformations over infinite dimensional groups [21, 31, 41, 11], we leave this case for conceptual treatment in the following chapter.

Chapter 4

Quantization of Spherical CNNs

Preliminaries

Convolutional Neural Networks (CNNs) are the standard method when it comes to learning with 2D planar images by exploiting the translationally invariant structures within images. However, planar Convolutional Neural Networks are not capable of analyzing spherical images, which are of interest to a number of problems in metheorology, astrophysics and in the development of omnidirectional computer vision. Planar projections of these spherical images for analysis with conventional CNNs would not be successful due to the space dissortions that such projections would cause on the images since we are working with different group actions; therefore, a new mathematical framework is required. In the last chapter we discussed some results after expanding to more general classes of group actions through the study of equivariance. Our algorithms of the last section are constrained to finite groups and both the space of the sphere and the group of rotation are infinite.

By definition, the object of final interest would be a network capable of recognizing patterns in the spherical space regardless of rotations, instead of shifts. This requires a different mathematical framework. Spherical CNNs have been introduced by [11] as a CNN where inputs are spherical functions f in S^2 that are lifted to functions on SO(3), through spherical cross-correlation with a filter k, $f \star k$. Note that f and k are functions on S^2, while $f \star k$ is a function on SO(3), as discussed below. The interest of this author is to first provide a literature review of the aforementioned framework, and propose potential quantizations for spherical CNNs. We assume oracle access to the convolution filter, and drawing the block encoding ideas from our original work in last chapter and motivated by the expression for equivariant convolutions on compact groups [52], [32], we would like to treat the infinite rotation group through a direct sum of finite-dimensional irreducible representations, i.e., the core object present in our earlier formulations.

4.1 Introduction

Motivation Infinite groups such as SO(3) –the group of 3D rotations, can be treated by working with a direct sum of the finite-dimensional irreducible representations. This requires converting the input data through to the irreducible representation. It is complex in general to deal with nonlinear features of maps and data, and implementations typically must be worked out on a per-group basis. In this study we work in the framework of the theory of spherical CNNs presented in [11]. We exploit the similarities of their formulations by working with group equivariant transformations and using the language of representation theory.

Setup As discussed earlier, planar projections of the spherical signals are deemed to fail due to the structural differences between the plane, the sphere and the groups of 2D translations and 3D rotations, respectively. We can start by highlighting the aforementioned structural differences between the plane and the sphere: the space for 2D translations is itself isomorphic to the plane; while the group of 3D rotations $SO(3)$ is not isomorphic to the sphere S^2 on which the group acts. The sphere S^2 is for instance not a group, but a homogeneous space upon which SO(3) acts transitively.

As we know, Convolutional Neural Networks can be represented as a chain of operations – i.e., convolutions – with optimized parameters and interleaved nonlinearities. With the structural differences in mind, we can discuss how convolutions and cross-correlations are defined on the planed and the sphere. While our work in the last chapter included the equivariant operations of convolution, cross-correlations and their right counterparts, in spherical CNNs cross-convolutuons are used in the forward pass, thus we focus on this operation for the rest of the chapter.

While in planar convolution, the output feature map at $x \in \mathbb{Z}^2$ is computed as the inner product between the input feature map and a filter, shifted by x; in spherical convolution, the output feature map at $R \in \mathrm{SO}(3)$ is computed as the inner product between the input feature map and a filter, rotated by R. The output feature map is indexed by a rotation: a function on SO(3). Below, we introduce the SO(3) group and the required mathematical terminology for the model of spherical CNNs.

4.1.1 SO(3) group

The *special orthogonal* group $SO(3)$[1], is the group of all rotations about the origin of three-dimensional Euclidean space. The group is non-abelian because rotations in 3D are not commutative. The group order is infinite, because you can rotate in this group by any continuous angle (or sets of angles). The group action is the product of three 3D rotation matrices $R_z(\alpha)R_y(\beta)R_z(\gamma)$ where $\alpha, \gamma \in [0, 2\pi], \beta \in [0, \pi]$ (Euler angles) and

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{4.1}$$

It is known that the SO(3) correlation/convolution satisfies a Fourier theorem with respect to the SO(3) Fourier transform, and the same is true for our definition of S2 correlation/convolution. Hence, the S^2 and SO(3) correlation/convolution can be implemented efficiently using generalized FFT algorithms. Technically, the two-sphere S^2 is not a group and therefore does not have irreducible representations. However, S^2 is a quotient of groups SO(3)/SO(2) and satisfies the relation $Y_m^l = D_{m0}^l|_{S^2}$.

[1] $SO(3)$ is represented as the 3×3 matrices of determinant one

4.1.2 Terminology

Spherical Signals Spherical images and filters can be modeled as continuous functions $f : S^2 \to \mathbb{R}^K$, where K is the number of channels.

Rotations Points on the sphere are represented as 3D unit vectors x and rotations are performed by the matrix-vector product Rx.

Rotation of Spherical signals The spherical correlation requires point rotation for $x \in S^2$ and also filter rotation for functions f on the sphere. Let us introduce L_R as the rotation operator that takes a function f and produces a rotated function $L_R f$ by composing f with the rotation R^{-1}:

$$[L_R f](x) = f(R^{-1}x). \tag{4.2}$$

Due to the inverse on R we have $L_{RR'} = L_R L_{R'}$.

Inner products The inner product on the vector space of spherical signals is defined as

$$\langle \psi, f \rangle = \int_{S^2} \sum_{k=1}^{K} \psi_k(x) f_k(x) dx \tag{4.3}$$

The integration measure dx denotes the standard rotation invariant measure on the sphere, $d^3x = d\alpha \sin\beta d\beta / 4\pi$, in spherical coordinates. The invariance of the measure ensures that

$$\int_{S^2} f(Rx) dx = \int_{S^2} f(x) dx, \tag{4.4}$$

for any rotation $R \in \mathrm{SO}(3)$. Using this fact, we can prove that $L_{R^{-1}}$ is adjoint to L_R, and thus L_R is unitary:

$$
\begin{aligned}
\langle L_R\psi, f\rangle &= \int_{s^2} \sum_{k=1}^{K} \psi_k(R^{-1}x) f_k(x) dx \\
&= \int_{S^2} \sum_{k=1}^{K} {}_k(x) f_k(Rx) dx \\
&= \langle \psi, L_{R^{-1}} f\rangle.
\end{aligned} \tag{4.5}
$$

The value of the output feature map evaluated at rotation $R \in \mathrm{SO}(3)$ is computed as an inner product between the input feature map and a filter, rotated by R. Since the output feature map is indexed by a rotation, the output feature map is modelled as a function on SO(3).

Spherical correlation For spherical signals f and ψ, we can define the correlation as:

$$
[\psi \star f](R) = \langle L_R\psi, f\rangle = \int_{S^2} \sum_{k=1}^{K} \psi_k(R^{-1}x) f_k(x) dx. \tag{4.6}
$$

As we know, the output of the spherical correlation is a function on SO(3), conterintuitive to the conventional definition of spherical convolution which otherwise gives a function on the sphere but where the filter is constrained to be circularly symmetric about the Z axis, which greatly limits the expressive capacity of the network.

Rotation of SO(3) **signals** We need to generalize the rotation operator L_R so that it can act on signals defined on SO(3). For $f : \mathrm{SO}(3) \to \mathbb{R}^K$, and $R, Q \in \mathrm{SO}(3)$:

$$
[L_R f](Q) = f(R^{-1}Q) \tag{4.7}
$$

The term $R^{-1}Q$ denotes the composition of rotations.

Rotation Group Correlation Using the same analogy as before, we can define the correlation of two signals on the rotation group, $f, \psi : \mathrm{SO}(3) \rightarrow \mathbb{R}^K$, as follows

$$[\psi \star f](R) = \langle L_R \psi, f \rangle = \int_{\mathrm{SO}(3)} \sum_{k=1}^{K} \psi_k(R^{-1}Q) f_k(Q) dQ. \tag{4.8}$$

The integration measure dQ is the invariant measure on SO(3), which may be expressed in ZYZ-Euler angles as $d\alpha \sin{(\beta)} d\beta d\gamma / (8\pi^2)$.

Equivariance Correlation is defined in terms of the rotation operator L_R which naturally acts on the input space of the network, but equivariance –a property shared by all kinds of convolution and correlation– allows to operate with L_R on the second layer and beyond.

A layer Φ is *e*quivariant if $\Phi \circ L_R = T_R \circ \Phi$, for some operator T_R. Using the definition of correlation and the unitarity of L_R,

Proof.

$$[\psi \star [L_Q f]](R) = \langle L_R \psi, L_Q f \rangle = \langle L_{Q^{-1}R} \psi, f \rangle = [\psi \star f](Q^{-1}R) = [L_Q[\psi f]](R).$$

this proof is valid for spherical correlation, S^2 and rotation group correlation SO(3).

□

4.1.3 Equivariant convolutions on compact groups

Recall the equivariant convolution layer equation for a general group G

$$\psi(f) = (f \star w)(u) = \int_G f \uparrow^G (ug^1) w \uparrow^G (g) d\mu(g) \tag{4.9}$$

Motivated by the fact that the convolutional integral becomes a product of irreducible representations, just like the convenient theorems where convolutions in

Fourier space become products. The expression simplifies to

$$\psi(f) = f_0 w_0 \oplus \vec{f_1} w_1 \oplus \ldots \oplus \vec{f_k} w_k, \tag{4.10}$$

thus we multiply the irreps by weights without mixing across irreps.

4.1.4 Irreducible representations on SO(3)

Nonlinearity There is an important equation for equivariant nonlinearity in SO(3), the **Clebsch-Gordan** tensor product and enables mixing between irreps. Let $\mathrm{CG}_{j,k,i}$ be the Clebsch-Gordan coefficients

$$\vec{f_i'} = \sum_j \sum_k \mathrm{CG}_{j,k,i} \cdot \vec{f_j} \vec{f_k}, \tag{4.11}$$

It can also be written as $\mathrm{CG}_{j,k,i} \vec{f} \otimes \vec{f}$, they maintain equivariance after multiplying all irreducible representations with all irreducible representations after a change of basis.

4.1.5 Spherical harmonics

The spherical harmonics $Y_m^l : S^2 \to \mathbb{C}$ are a complete orthogonal family of functions. The spherical harmonics are related to the Wigner D functions by $D_{mn}^l(\alpha, \beta, \gamma) = Y_m^l(\alpha, \beta) e^{in\gamma}$, so that $Y_m^l(\alpha, \beta) = D_{m0}^l(\alpha, \beta, 0)$.

The inverse SO(3) Fourier transform is defined as

$$f(R) = \sum_{l=0}^{b} (2l+1) \sum_{m=-l}^{l} \sum_{n=-l}^{l} \hat{f}_{mn}^l D_{mn}^l(R) \tag{4.12}$$

4.2 Discussion

Spherical CNNs are important for many applications in the applied and pure sciences. For instance, omnidirectional vision for drones, robots, and autonomous cars, climate modelling and weather predictions. In chemistry, spherical symmetries are crucial when modeling molecules and predicting molecular design properties. In astrophysics and cosmology, data is naturally equipped with spherical geometry. A quantization of the CNNs could achieve a good number of important realizations for quantum computers, such as virtual drug screening.

Chapter 5

Discussion and conclusions

This thesis presented a set of quantum algorithms for group convolution, cross-correlation, and equivariant transformations. These quantum algorithms run in time logarithmic in the dimension of the symmetry group, and are thus exponentially faster than the corresponding classical algorithms.

Our algorithms provide a path towards quantizing the linear operations in group-equivariant neural networks [10, 32, 44, 38] and exploring potential quantum speedups in these machine learning models. Furthermore, one can apply our framework in a variational algorithm where a quantum circuit is parameterized and optimized as a convolutional filter [7]. More generally, our work provides a means to speed up linear operations for kernel matrices in the form of convolutions or cross-correlations commonly found in algorithms for machine learning and numerical methods [43, 16, 2]. This generalizes results from previous quantum algorithms for implementing circulant or Toeplitz matrices [51, 55] and calculating Green's functions via convolutional formulations [49] using our algorithms for inverting group convolutions.

In the context of geometric deep learning, our results and observations on applying quantum mechanics to equivariance can be seen the context of geometric learning as part the current attempt of *unification* of a broad class of problems in deep learning from the perspectives of symmetry and invariance.

5.1 Implementation on software

PennyLane is an open-source software framework built around the concept of quantum differentiable programming, based on the integration of classical ML libraries with quantum hardware and simulators that allows the user to train quantum circuits. It can be a resource in the development of quantum machine learning and variational quantum circuits.

TensorFlow Quantum is a library for hybrid quantum-classical machine learning. (TFQ) is a quantum machine learning library for rapid prototyping of hybrid quantum-classical ML models. Research in quantum algorithms and applications can leverage Google's quantum computing frameworks, all from within TensorFlow.

TensorFlow Quantum focuses on quantum data and building hybrid quantum-classical models. It integrates quantum computing algorithms and logic designed in Cirq[1], and provides quantum computing primitives compatible with existing TensorFlow APIs, along with high-performance quantum circuit simulators. TensorFlow Quantum (TFQ) provides layer classes designed for in-graph circuit construction.

5.2 Relevant Publication

- **G. Castelazo**, Q. T. Nguyen, G. De Palma, D. Englund, S. Lloyd, B.T. Kiani, "Quantum algorithms for group convolution, cross-correlation, and equivariant transformations" quant-ph. Phys. (2021); arXiv:2109.11330v1.

[1] Cirq is a Python library for writing, manipulating, and optimizing quantum circuits and running them against quantum computers and simulators.

Appendix A

Equivariant Neural Networks

A.1 Example on G-Equivariant Convolutions on Finite Groups

We play with a simulation [52] to build a classical equivariant network for an example finite group, in this case, Z_6 of vertices on a hexagon.

Finite Group Z_6. Group of rotations of a hexagon. This group is indexed by the vertex coordinates $\{0, 1, 2, 3, 4, 5\}$. We can view rotation r^n as the operation $(x + n)$ mod 6. The group is indeed closed and we have $\{e, r, r^2, r^3, r^4, r^5\}$. The Cayley table of Z_6 is

Recall the definition of the group convolution operator $*$, for a linear map ψ

	e	r	r^2	r^3	r^4	r^5
e	e	r	r^2	r^3	r^4	r^5
r	r	r^2	r^3	r^4	r^5	e
r^2	r^2	r^3	r^4	r^5	e	r
r^3	r^3	r^4	r^5	e	r	r^2
r^4	r^4	r^5	e	r	r^2	r^3
r^5	r^5	e	r	r^2	r^3	r^4

Table A.1: Cayley table of Z_6

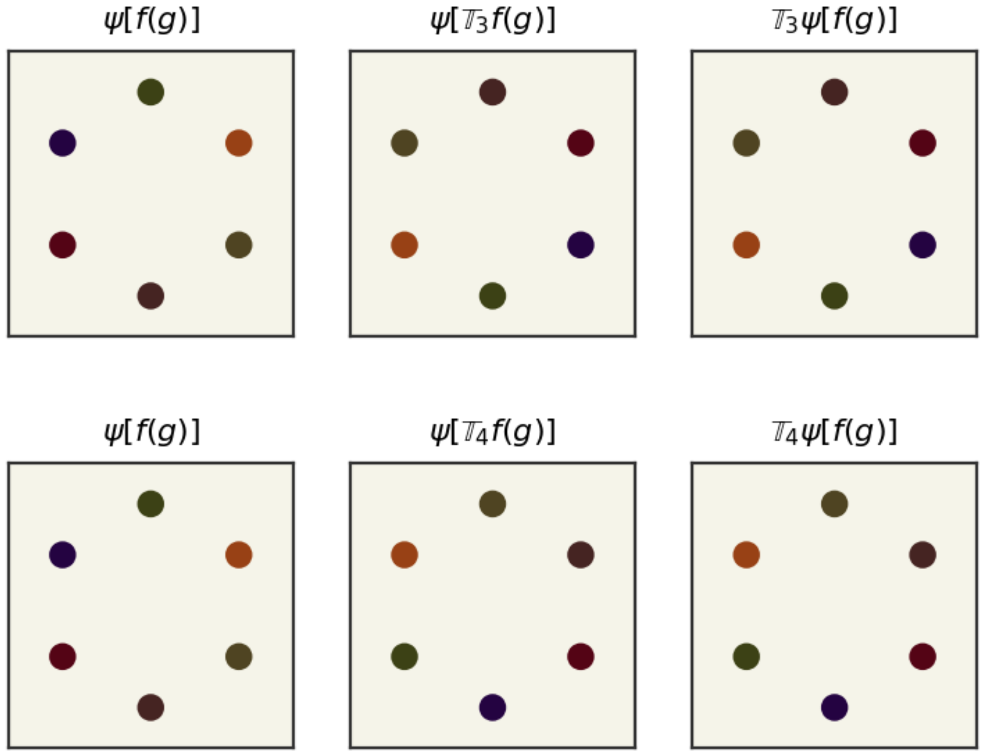

Figure A-1: A network will be G-equivariant if the output looks the same by applying the rotation transformation $\mathbb{T}_g$ before or after.

$$\psi(f) = (f \star \omega)(u) = \sum_{g \in G} f(ug^{-1})\omega(g) \tag{A.1}$$

where $f : H \to \mathbb{R}^n$ and $\omega H' \to \mathbb{R}^n$ are functions of quotient spaces H and H'.

In this example, it can be verified by comparing the results of transforming the input function and the output functions.

- We first compute $\psi[\mathbb{T}_g f(x)]$ – the network map acting on the transformed input function.
- Secondly, we compute $\mathbb{T}_g \psi[f(x)]$ – the transform acting on the network output.

The example shown above is to show the reader the effects of equivariance in a classical simulation and we are reproducing the code of the reference [52].

Appendix B

Convolution theorem over D_3

B.1 Representations of D_n

Dihedral groups The dihedral group D_n is the group of symmetries of the regular n-gon in the plane. The dihedral group D_n is of order $2n$ and is represented by $D_n = \mathbb{Z}/n\mathbb{Z} \rtimes \mathbb{Z}/2\mathbb{Z}$, with the group law

$$(x, a) \cdot (y, b) = (x + (-1)^a y, a + b),$$

for $x, y \in \mathbb{Z}/n\mathbb{Z}$ and $a, b \in \mathbb{Z}/2\mathbb{Z}$.

The dihedral group D_n with $2n$ elements is isomorphic to a semidirect product of the cyclic groups $\mathbb{Z}/n\mathbb{Z}$ and $\mathbb{Z}/2\mathbb{Z}$. Let r be the generator of $\mathbb{Z}/n\mathbb{Z}$ and s be the generator of $\mathbb{Z}/2\mathbb{Z}$, then the dihedral group D_n can be written compactly as

$$\langle r, s | s^2 = e, r^n = e, srs^{-1} = r^{-1} \rangle. \tag{B.1}$$

Irreducible representations of D_n For n even, we have the following 1-dimensional irreducible representations

$$\begin{aligned}\sigma_{tt}((x,a)) &= 1\\ \sigma_{ts}((x,a)) &= (-1)^a\\ \sigma_{st}((x,a)) &= (-1)^x\\ \sigma_{ss}((x,a)) &= (-1)^{x+a}.\end{aligned} \tag{B.2}$$

For n odd, we have σ_{tt} and σ_{ts} only. The 2-dimensional irreducible representations are of the form

$$\sigma_h((x,0)) = \begin{pmatrix} e^{2\pi ihx/n} & 0 \\ 0 & e^{-2\pi ihx/n}\end{pmatrix} \qquad \sigma_h((x,1)) = \begin{pmatrix} 0 & e^{2\pi ihx/n} \\ e^{-2\pi ihx/n} & 0\end{pmatrix}, \tag{B.3}$$

for $h \in \{1, 2, \dots, \lceil \frac{n}{2} \rceil - 1\}$. The sum of the squared dimensions of the irreducible representations is equal to $2n$, which is the size of the group:

$$\sum_\sigma |d_\sigma|^2 = 2n = |G|.$$

B.1.1 Representations of D_3

The dihedral group D_3 is obtained by composing the six symmetries of an equilateral triangle. The dihedral group D_3 and the cyclic group C_6 are the only two groups that have order 6. Unlike C_6 (which is abelian), D_3 is non-abelian. Products of group elements of D_3 are shown in the Cayley table shown in B.1.

Like all dihedral groups, group elements of D_3 are generated by s and r, where s is a rotation by π radians about an axis passing through the center and one of the vertices of a regular n-gon and r is a rotation by $2\pi/n$ about the center of the n-gon (see B-1).

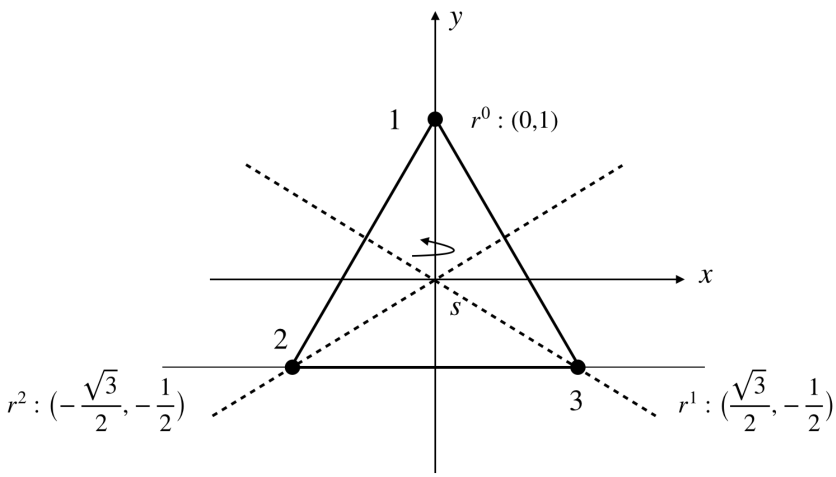

Figure B-1: The dihedral group D_3 is the symmetry group of an equilateral triangle, that is, it is the set of all transformations such as reflection, rotation, and combinations of these, that leave the shape and position of this triangle fixed.

	1	r	r^2	s	rs	r^2s
1	1	r	r^2	s	rs	r^2s
r	r	r^2	1	rs	r^2s	s
r^2	r^2	1	r	r^2s	s	rs
s	s	r^2s	rs	1	r^2	r
rs	rs	s	r^2s	r	1	r^2
r^2s	r^2s	rs	s	r^2	r	1

Table B.1: The Cayley table of D_3.

Left and right regular representations of D_3. The regular representations of D_3 are obtained by associating a basis vector to each element of the group $\{1, r, r^2, s, rs, r^2s\}$.

$$\begin{aligned}
\vec{e}_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
\vec{e}_r &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \\
\vec{e}_{r^2} &= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \\
\vec{e}_s &= \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \\
\vec{e}_{rs} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \\
\vec{e}_{r^2s} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned} \tag{B.4}$$

For all $u \in D_3$, the left regular representations L_u are:

$$L_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad L_r = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{B.5}$$

$$L_{r^2} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \qquad L_s = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{B.6}$$

$$L_{rs} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad L_{r^2s} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{B.7}$$

For all $u \in D_3$, the right regular representations L_u are:

$$R_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad R_r = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{B.8}$$

$$R_{r^2} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad R_s = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{B.9}$$

$$R_{rs} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad R_{r^2s} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{B.10}$$

Irreducible representations of D_3 Since D_3 is a non-abelian group, at least one of its irreducible representations is a matrix. B.2 shows the irreducible representations of D_3, obtained from B.2 and B.3, noting that $h \in \{1, 2, \ldots, \lceil \frac{n}{2} \rceil - 1\} = \{1\}$.

B.2 The Group Fourier Transform and the Convolution Theorem over D_3

The group Fourier transform table (i.e. F_G) for D_3 can be constructed by aligning the elements of the 2-dimensional representation ρ_3 elementwise $(\rho_{3_{11}}, \rho_{3_{12}}, \rho_{3_{21}}, \rho_{3_{22}})$, yielding a 6×6 transformation matrix.

The normalized (unitary) Fourier transformation matrix F_G is defined as [8]

$$F_G = \sum_{x \in G} |\hat{x}\rangle \langle x| = \sum_{x \in G} \sum_{\rho \in \hat{G}} \sqrt{\frac{d_\rho}{|G|}} \sum_{j,k=1}^{d_\rho} \rho(x)_{j,k} |\rho, j, k\rangle \langle x|$$

where $\hat{G}$ is the set of irreducible representations and the $\sqrt{\frac{d_\rho}{|G|}}$ factor enforces F_G as

Table B.2: Irreducible representations of D_3

	ρ_1 $\sigma_{tt}((x,a))$	ρ_2 $\sigma_{ts}((x,a))$	ρ_3 $\sigma_1((x,a))$
(0,0)	1	1	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
(1,0)	1	1	$\begin{pmatrix} \omega^1 & 0 \\ 0 & \omega^{-1} \end{pmatrix}$
(2,0)	1	1	$\begin{pmatrix} \omega^2 & 0 \\ 0 & \omega^{-2} \end{pmatrix}$
(0,1)	1	-1	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
(1,1)	1	-1	$\begin{pmatrix} 0 & \omega^1 \\ \omega^{-1} & 0 \end{pmatrix}$
(2,1)	1	-1	$\begin{pmatrix} 0 & \omega^2 \\ \omega^{-2} & 0 \end{pmatrix}$

unitary. For D_3, we have:

$$F_G = \begin{pmatrix} 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \\ 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} \\ 1/\sqrt{3} & \omega^1/\sqrt{3} & \omega^2/\sqrt{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/\sqrt{3} & \omega^1/\sqrt{3} & \omega^2/\sqrt{3} \\ 0 & 0 & 0 & 1/\sqrt{3} & \omega^{-1}/\sqrt{3} & \omega^{-2}/\sqrt{3} \\ 1/\sqrt{3} & \omega^{-1}/\sqrt{3} & \omega^{-2}/\sqrt{3} & 0 & 0 & 0 \end{pmatrix}. \tag{B.11}$$

Let m and f be functions that map group elements of G to complex numbers, if we associate each element $u \in G$ to a basis vector e_u in some vector space V, we can represent m and f as vectors,

$$\vec{m} = \sum_{u \in G} m(u) e_u, \qquad \vec{f} = \sum_{u \in G} f(u) e_u.$$

As an example, we take

$$\vec{m} = \vec{f} = \begin{pmatrix} 1 & \omega^1 & \omega^2 & 0 & 0 & 0 \end{pmatrix}, \tag{B.12}$$

where we have chosen e_u to be the standard basis of $\mathbb{C}^{|G|}$.

Calculating the Fourier transform through matrix multiplication on $\vec{m}$ and $\vec{f}$ we obtain

$$\widehat{m} = \widehat{f} = F_G\vec{m} = F_G\vec{f} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \sqrt{3} \end{pmatrix} \tag{B.13}$$

Note that $1+\omega^1+\omega^2 = 0$ since the cube root of unity (i.e. $\omega^3 = 1$) can be factorized as

$$\omega^3 - 1 = (\omega - 1)(\omega^2 + \omega + 1) = 0.$$

In the following we compare the computation of the Fourier transform of $\vec{m} \circledast \vec{f}$ by two different methods: (1) through direct calculation of the convolution and applying the Fourier transform, and (2) by computing the individual Fourier transforms of $\vec{m}$ and $\vec{f}$ via the convolution theorem.

First, we complete case (1). Recall the definition of a convolution over a group G

$$(m \circledast f)(u) = \sum_{v \in G} m(uv^{-1})f(v)$$

Calculating the convolution expansion over the indexed elements of D_3, where $1, 2, 3, 4, 5, 6$ correspond to (refer to B.1 for group element multiplication uv^{-1})

$$(m \circledast f)(1) = m(1)\cdot f(1) + m(3)\cdot f(2) + m(2)\cdot f(3) + m(4)\cdot f(4) + m(5)\cdot f(5) + m(6)\cdot f(6)$$

$$(m \circledast f)(2) = m(2)\cdot f(1) + m(1)\cdot f(2) + m(3)\cdot f(3) + m(5)\cdot f(4) + m(6)\cdot f(5) + m(4)\cdot f(6)$$

$$(m \circledast f)(3) = m(3)\cdot f(1) + m(2)\cdot f(2) + m(1)\cdot f(3) + m(6)\cdot f(4) + m(4)\cdot f(5) + m(5)\cdot f(6)$$

$$(m \circledast f)(4) = m(4)\cdot f(1) + m(5)\cdot f(2) + m(6)\cdot f(3) + m(1)\cdot f(4) + m(3)\cdot f(5) + m(2)\cdot f(6)$$

$$(m \circledast f)(5) = m(5)\cdot f(1) + m(6)\cdot f(2) + m(4)\cdot f(3) + m(2)\cdot f(4) + m(1)\cdot f(5) + m(3)\cdot f(6)$$

$$(m \circledast f)(6) = m(6)\cdot f(1) + m(4)\cdot f(2) + m(5)\cdot f(3) + m(3)\cdot f(4) + m(2)\cdot f(5) + m(1)\cdot f(6).$$

In particular, for m and f

$$\begin{aligned}
(m \circledast f)(1) =& 1 \cdot 1 + \omega^2 \cdot \omega^1 + \omega^1 \cdot \omega^2 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 = 3 \\
(m \circledast f)(2) =& \omega^1 \cdot 1 + 1 \cdot \omega^1 + \omega^2 \cdot \omega^2 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 = 3\omega^1 \\
(m \circledast f)(3) =& \omega^2 \cdot 1 + \omega^1 \cdot \omega^1 + 1 \cdot \omega^2 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 = 3\omega^2 \\
(m \circledast f)(4) =& (m \circledast f)(5) = (m \circledast f)(6) = 0.
\end{aligned}$$

Written in vector form,

$$\vec{m} \circledast \vec{f} = \begin{pmatrix} 3 & 3\omega^1 & 3\omega^2 & 0 & 0 & 0 \end{pmatrix},$$

calculating the Fourier transform of $\vec{m} \circledast \vec{f}$ through matrix multiplication with F_G,

$$\widehat{m \circledast f} = F_G(\vec{m} \circledast \vec{f}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3\sqrt{3} \end{pmatrix}.$$

Now, we overview case (2). Recall the convolution theorem (3.9)

$$(\widehat{m \circledast f})(\rho_i) = \widehat{m}(\rho_i)\widehat{f}(\rho_i). \tag{B.14}$$

First we compute the group Fourier transform of m and f over the irreducible representations,

$$\widehat{m}(\rho_1) = \widehat{f}(\rho_1) = 0 \qquad \widehat{m}(\rho_2) = \widehat{f}(\rho_2) = 0 \qquad \widehat{m}(\rho_3) = \widehat{f}(\rho_3) = \begin{pmatrix} 0 & 0 \\ 0 & 3 \end{pmatrix} \tag{B.15}$$

Applying the convolution theorem, we have

$$\begin{aligned}
\widehat{m \circledast f}(\rho_1) &= \widehat{m}(\rho_1)\widehat{f}(\rho_1) = 0 \\
\widehat{m \circledast f}(\rho_2) &= \widehat{m}(\rho_2)\widehat{f}(\rho_2) = 0 \\
\widehat{m \circledast f}(\rho_3) &= \widehat{m}(\rho_3)\widehat{f}(\rho_3) = \begin{pmatrix} 0 & 0 \\ 0 & 3 \end{pmatrix}
\end{aligned} \tag{B.16}$$

Aligning the elements of $\widehat{m \circledast f}(\rho_i)$ in order for all the irreps ρ_i on a 6-dim vector on the same standard basis we obtain

$$\widehat{m \circledast f} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 9 \end{pmatrix} \tag{B.17}$$

Note that B.2 and B.17 differ by a factor of normalization $\sqrt{d_\rho/|G|}$ since in method (1) F_G is already forced to be unitary, while method (2) is based on purely classical calculations.

Alternatively, we can also perform the group Fourier transform in the block encoding form of 3.29,

$$\begin{aligned}\widehat{m \circledast f} &= \Big[\bigoplus_{v \in G} \widehat{m}(\rho_i) \otimes I_{d_\rho}\Big]\widehat{f} \\ &= \left(\begin{bmatrix}0\end{bmatrix} \oplus \begin{bmatrix}0\end{bmatrix} \oplus \begin{bmatrix}0 & 0 \\ 0 & 3\end{bmatrix} \otimes I_2\right)\begin{pmatrix}0\\0\\0\\0\\0\\\sqrt{3}\end{pmatrix} = \begin{pmatrix}0\\0\\0\\0\\0\\3\sqrt{3}\end{pmatrix}\end{aligned} \tag{B.18}$$

which already includes the normalization factor on F_G.

The example shown above is also included in our published paper [6], it was designed and written in its entirety by the author of this thesis.

Bibliography

[1] Steven H Adachi and Maxwell P Henderson. Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*, 2015.

[2] Kendall Atkinson and Weimin Han. Numerical solution of fredholm integral equations of the second kind. In *Theoretical Numerical Analysis*, pages 473–549. Springer, 2009.

[3] Alexander Bogatskiy, Brandon Anderson, Jan Offermann, Marwah Roussi, David Miller, and Risi Kondor. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pages 992–1002. PMLR, 2020.

[4] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021.

[5] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

[6] Grecia Castelazo, Quynh T Nguyen, Giacomo De Palma, Dirk Englund, Seth Lloyd, and Bobak T Kiani. Quantum algorithms for group convolution, cross-correlation, and equivariant transformations. *arXiv preprint arXiv:2109.11330*, 2021.

[7] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[8] Andrew M Childs and Wim Van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):1, 2010.

[9] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2019.

[10] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[11] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018.

[12] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.

[13] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Physical Review A*, 80(1):012304, 2009.

[14] Anton Deitmar. *The Peter-Weyl Theorem*, pages 127–134. Springer New York, New York, NY, 2002.

[15] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. In *International Conference on machine learning*, pages 1889–1898. PMLR, 2016.

[16] Claude R Dietrich and Garry N Newsam. Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix. *SIAM Journal on Scientific Computing*, 18(4):1088–1107, 1997.

[17] Zi-Han Ding, Jin-Ming Cui, Yun-Feng Huang, Chuan-Feng Li, Tao Tu, and Guang-Can Guo. Fast high-fidelity readout of a single trapped-ion qubit via machine-learning methods. *Phys. Rev. Applied*, 12:014038, Jul 2019.

[18] David Steven Dummit and Richard M Foote. *Abstract algebra*, volume 3. Wiley Hoboken, 2004.

[19] Joseph Emerson, Etera Livine, and Seth Lloyd. Convergence conditions for random quantum circuits. *Physical Review A*, 72(6):060302, 2005.

[20] Carlos Esteves. Theoretical aspects of group equivariant neural networks. *arXiv preprint arXiv:2004.05154*, 2020.

[21] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020.

[22] Emmanuel Flurin, Leigh S Martin, Shay Hacohen-Gourgy, and Irfan Siddiqi. Using a recurrent neural network to reconstruct quantum dynamics of a superconducting qubit from physical observations. *Physical Review X*, 10(1):011006, 2020.

[23] William Fulton and Joe Harris. *Representation theory: a first course*, volume 129. Springer Science & Business Media, 2013.

[24] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, 2018.

[25] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008.

[26] LONG Gui-Lu and Liu Yang. Duality computing in quantum computers. *Communications in Theoretical Physics*, 50(6):1303, 2008.

[27] Peter Hoyer. Efficient quantum transforms. *arXiv preprint quant-ph/9702028*, 1997.

[28] Iordanis Kerenidis, Jonas Landman, and Anupam Prakash. Quantum algorithms for deep convolutional neural networks. In *International Conference on Learning Representations*, 2019.

[29] Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. *Advances in Neural Information Processing Systems*, 32:7092–7101, 2019.

[30] Bobak Toussi Kiani, Giacomo De Palma, Dirk Englund, William Kaminsky, Milad Marvian, and Seth Lloyd. Quantum advantage for differential equation analysis. *Physical Review A*, 105(2):022415, 2022.

[31] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. *Advances in Neural Information Processing Systems*, 31:10117–10126, 2018.

[32] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755. PMLR, 2018.

[33] Robin Kothari. *Efficient algorithms in quantum query complexity*. PhD thesis, University of Waterloo, 2014.

[34] Junhua Liu, Kwan Hui Lim, Kristin L Wood, Wei Huang, Chu Guo, and He-Liang Huang. Hybrid quantum-classical convolutional neural networks. *arXiv preprint arXiv:1911.02998*, 2019.

[35] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

[36] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019.

[37] Anuradha Mahasinghe and JB Wang. Efficient quantum circuits for toeplitz and hankel matrices. *Journal of Physics A: Mathematical and Theoretical*, 49(27):275301, 2016.

[38] Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. In *International Conference on machine learning*, pages 4363–4371. PMLR, 2019.

[39] Ashley Montanaro. Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015.

[40] Cristopher Moore, Daniel Rockmore, and Alexander Russell. Generic quantum fourier transforms. *ACM Transactions on Algorithms (TALG)*, 2(4):707–723, 2006.

[41] Nathanaël Perraudin, Michaël Defferrard, Tomasz Kacprzak, and Raphael Sgier. Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications. *Astronomy and Computing*, 27:130–146, 2019.

[42] Arthur Pesah, M Cerezo, Samson Wang, Tyler Volkoff, Andrew T Sornborger, and Patrick J Coles. Absence of barren plateaus in quantum convolutional neural networks. *Physical Review X*, 11(4):041011, 2021.

[43] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[44] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. In *International Conference on Machine Learning*, pages 2892–2901. PMLR, 2017.

[45] Martin Roetteler. Quantum algorithms for abelian difference sets and applications to dihedral hidden subgroups. In *11th Conference on the Theory of Quantum Computation, Communication and Cryptography*, 2016.

[46] Martin Rötteler. Quantum algorithms for highly non-linear boolean functions. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete algorithms*, pages 448–457. SIAM, 2010.

[47] Changpeng Shao and Hua Xiang. Quantum circulant preconditioner for a linear system of equations. *Physical Review A*, 98(6):062321, 2018.

[48] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[49] Yu Tong, Dong An, Nathan Wiebe, and Lin Lin. Fast inversion, preconditioned quantum linear system solvers, fast green's-function computation, and fast evaluation of matrix functions. *Physical Review A*, 104(3):032422, 2021.

[50] Wim Van Dam, Sean Hallgren, and Lawrence Ip. Quantum algorithms for some hidden shift problems. *SIAM Journal on Computing*, 36(3):763–778, 2006.

[51] Lin-Chun Wan, Chao-Hua Yu, Shi-Jie Pan, Fei Gao, Qiao-Yan Wen, and Su-Juan Qin. Asymptotic quantum algorithm for the toeplitz systems. *Physical Review A*, 97(6):062322, 2018.

[52] Andrew D White. Equivariant neural networks, 2022.

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

[54] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, volume 30, 2017.

[55] SS Zhou and JB Wang. Efficient quantum circuits for dense circulant and circulant like operators. *Royal Society open science*, 4(5):160906, 2017.