THE UNIVERSITY OF CHICAGO

TROPICAL GEOMETRY, NEURAL NETWORKS, AND LOW-COHERENCE FRAMES

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY

LIWEN ZHANG

CHICAGO, ILLINOIS

JUNE 2018

To my family

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to extend the sincerest gratitude to my advisor, Lek-Heng Lim, for being supportive, encouraging and resourceful. Lek-Heng often provides us with brilliant and interesting insights into academic research and daily life. From time to time, I have been impressed by his power of connecting things that were seemingly totally unrelated. The idea of applying tropical geometry to neural network attributes to Lek-Heng as well. His enthusiasm and optimism torwards research always inspire and energize us to solve challenging problems. Lek-Heng has also offered us tremendous opprotunity to receive academic training outside the campus and supported us to attend summer schools, conferences and internships. This thesis and, more importantly, the courage of attacking difficult research problems would not have been possible without his support.

I am also extremely grateful to Ryota Tomioka for his mentorship during the early stage of my time at University of Chicago and later during my internship at Microsoft Research. Both experience had extensive impact on my graduate study. Ryota is always very approachable and generous in sharing his time and ideas which has lead to many helpful discussions and tons of useful hands-on guidance. I also learned a lot from Ryota about how to write a research paper and give an academic presentation, skills that will be useful for the rest of my career.

I am thankful to my committee Risi Kondor and John Goldsmith for their time and feedback on this thesis. I have "sneaked in" and attended handful of Risi's group meetings and learned a lot from him and his machine learning group. John has provided me with

many useful pointers and his own insights into natural language which are valuable for my future study in this direction. Besides, I had a great time collaborating with some excellent researchers. A large portion of the work of tropical geometry and neural networks stemmed from many hours of discussion with Gregory Naitzat. Subhransu Maji, together with Ryota, guided me to complete my first research project and conference submission since I joined University of Chicago. To everyone who have contributed to my knowledge base through discussions or collaborations, I am truely grateful.

Last but not the least, to my beloved family, I really appreciate your everlasting help and support. To all my friends, too many to name, thank you for bringing so much joy and memorable moments to my life. It was a long journey full of ups and downs. It was you who share me the strength to accomplish it.

Abstract

This dissertation consists of three pieces of work. The first work aims to set up the connection between tropical geometry and feedforward neural networks. We discovered that, mathematically, a feedforward neural network equipped with rectified linear units (ReLU) is a tropical rational function. This connection provides a new approach to understand and analyze deep neural networks. Among other things, we show that the decision boundary derived from an ReLU neural network is contained by a tropical hypersurface of a tropical polynomial in companion with the network. Moreover, we associate functions represented by feedforward neural networks with polytopes and show that a two layer network can be fully characterized by zonotopes which also serve as the building blocks for deeper networks. Also, the number of vertices on the polytopes provides an upper bound on the number of linear regions of the function expressed by the network. We show that this upper bound grows exponentially with the number of layers but only polynomially with respect to number of hidden nodes in each layer.

In the second work, we propose an attention model in continuous vector space for content-based neural memory access. Our model represents knowledge graph entities as low-dimensional vectors while expressing context-dependent attention as a Gaussian scoring function over the vector space. We apply such a model to perform tasks such as knowledge graph completion and complex question answering. The proposed attention model can handle both the propagation of the uncertainty when following a series of relations and also the conjunction of conditions in a natural way. On a dataset of soccer players who participated

in the FIFA World Cup 2014, we demonstrate that our model can handle both path queries and conjunctive queries well.

The third work focus on building finite complex frames generated by cyclic vectors under the action of non-commutative groups. We inspect group frames in the space of operators associated with the group's von Neumann algebra. The searching for a proper cyclic vector is then transformed to finding the intersection of a convex set that prescribes the coherence constraints and a subset of Hermitian rank-one operators. An alternating projection algorithm is employed to search for their intersection and an heuristic extrapolation technique is adapted to accelerate the computation. In the experiments, we applied our model to Heisenberg groups and finite affine groups. In the case of Heisenberg group, our method is able to find cyclic vectors that generate equiangular tight frames up to numerical precision.

Chapter 1

Overview

The resurrection of deep neural network has lead to innovation and state of arts in various tasks. At the same time, it has inspired lots of new research directions in both application and theory. Active investigations have been made to understand the remarkable performance of neural networks. But it still remains a big challenge to fully understand the mechanism of a deep neural network. In the first work presented in this dissertation, we establish, for the first time, connection between feedforward neural networks and tropical geometry. Tropical geometry is a relatively new and fast-growing field of algebraic geometry but has few intersection with artificial intelligence so far. By bridging tropical geometry with neural networks, we open the opportunity for researchers to empoloy tools from the former to tackle problems in the latter. As the very first step, we establish this tropical perspective of neural networks in Chapter 2. Under mild assumptions, we show that the family of feedforward neural networks with ReLU activation is equivalent to the semi-field of tropical rational functions. In the meantime, we put together some basic observations and theory which follow immediately from this connection. Among other things, we found that a neural network with one hidden layer can be characterized by zonotopes which serve as building blocks for deeper networks. Besides, decision boundary derived from neural networks can be related to tropical hypersurface which is a major subject of interest in tropical geometry. We also discovered that linear regions of a neural network corresponds to vertices of polytopes associated with trop-

ical rational functions. We also recapitulate the exponential expressivenss of deep networks by using tools from tropical geometry and show that a deeper network is capable of dividing the domain into exponentially more linear regions than a shallow network.

As for applications, neural networks have been empowering us to perform more and more sophisticated tasks. Recently, neural word embedding has achieved extraordinary success in the area of natural language processing. By representing words in continuous vector space and training the embedding with skip-gram and negative-sampling, (Mikolov et al., 2013a,b) showed that low-dimensional embedding is an efficient way to encode semantic relations among words by achieving state-of-the-art results on various linguistic tasks. Embedding has been widely employed for knowledge representation in many other tasks as well. Multi-relational knowledge graphs' embedding is one of them. In Chapter 3, we propose a new embedding model, TransGaussian, for knowledge graph representation. TransGaussian model represents knowledge graph entities as vectors while expressing context-dependent attention as a Gaussian scoring function over the vector space. To perform question answering on knowledge graph, we train a recurrent neural network so that it maps a question posed in natural language to a Gaussian function under which the answer of the question receives a high score. Meanwhile, the proposed attention model can handle both the propagation of the uncertainty when following a series of relations and also the conjunction of conditions in a natural way. On a dataset of soccer players who participated in the FIFA World Cup 2014, we demonstrate that our model can handle both path queries and conjunctive queries well.

In Chapter 4, we study frames that are generated by cyclic vectors under the action of non-commutative groups. Non-commutative groups are behind the construction of many well-known frames and orthonormal systems. For example, a Gabor frame $\left\{e^{2\pi i\beta lt}g(t-\alpha k) : k,l \in \mathbb{Z}\right\} \subset L^2(\mathbb{R})$ is constructed by translations and modulations of the atomic function g; A wavelet basis $\left\{D_A^m T_v\psi : m \in \mathbb{Z}, v \in \mathbb{Z}^n\right\} \subset L\big(\mathbb{R}^n\big)$ is closely related with the group

generated by the dilation operator D_A and the translation operator T_v; The standard basis of $\mathbb{R}^n$ can be considered as the orbit of $e_1 \coloneqq [1, 0, \ldots, 0]^\top$ under the symmetric group S_n. In this work, we investigate the problem of building low-coherence finite complex frames and propose a general framework for finding a cyclic vector that generates a group frame with coherence constraint. We first map group frames to the space of operators associated with the group's von Neumann algebra. Thus, the searching for a proper cyclic vector is transformed to finding the intersection of a convex set that prescribes the coherence constraints and a subset of Hermitian rank-one operators. An alternating projection algorithm is employed to search for their intersection. And this algorithm is proved to have local convergence rate. We also derive an equivalent formula for carrying out the algorithm in the space of group representation. However, viewing the algorithm as algernating projection allows us to adapt an heuristic extrapolation technique which leads to much faster convergence in our experiments. We tried out our method on Heisenberg groups and finite affine groups of different dimensions. In the case of Heisenberg group, our method is able to find cyclic vectors that generate equiangular tight frames up to numerical precisions.

Chapter 2

A Tropical Geometrical View of Deep Neural Networks

2.1 Introduction

Deep neural networks have recently received much limelight for their enormous success in a variety of applications across many different areas of artificial intelligence, computer vision, speech recognition, and natural language processing LeCun et al. (2015); Hinton et al. (2012); Krizhevsky et al. (2012); Bahdanau et al. (2014); Kalchbrenner and Blunsom (2013). However, it is also well-known that the theoretical and mathematical understanding of their workings remains incomplete.

There have been several attempts to analyze deep neural networks from different perspectives to shed light on their theoretical properties and explain their efficacy. Notably, earlier studies have suggested that a deep architecture could be efficiently used to express a complicated family of functions while still maintaining a relatively simple structure. A deep neural network uses its parameters more efficiently and hence requires exponentially less parameters to express certain families of functions Delalleau and Bengio (2011); Bengio and Delalleau (2011); Montufar et al. (2014); Eldan and Shamir (2016). Recent work in Zhang et al. (2016) showed empirically that several successful neural networks can shatter the training set when sizes of the networks are large enough. In addition, the authors pointed out that these models, while possessing a high representation power, are also regularized and

"simple" at the same time, in the sense that they generalize to data not seen during the training stage. It remains a challenge to explain these properties of neural networks and to find the right formal measure of complexity that captures their generalization capabilities.

In this work, we focus on feedforward neural networks with rectified linear units, arguably the most fundamental and rudimentary neural network, and also one of the most widely used (possibly in conjunction with recurrent or convolutional neural networks) type of neural networks in deep learning. We show that such a neural network is the analogue of a *rational function*, i.e., a ratio of two multivariate polynomials f, g in variables $x_1, \dots, x_d$,

$$\frac{f(x_1, \dots, x_d)}{g(x_1, \dots, x_d)},$$

in *tropical algebra* or *tropical algebraic geometry*. This is a new area in algebraic geometry that has seen an explosive growth in the recent decade but remains relatively obscure outside pure mathematics. In fact, it is a surprise to us that the two subjects — deep learning and tropical algebraic geometry — are even related.

We have been vague about the word "polynomials" in the previous paragraph. We do not mean usual polynomials on the real line $\mathbb{R}$ (e.g., in Taylor approximation), or trigonometric polynomials on the circle $\mathbb{S}^1$ (e.g., in Fourier approximation), or multivariate versions of these, but *tropical polynomials* that we will define in Section 2.2. For usual and trigonometric polynomials, it is known that doing *rational approximation* — approximating a target function by a ratio of two polynomials instead of a single polynomial — vastly improves the quality of approximation without increasing the degree. This gives our analogue: A neural network[1] is the tropical ratio of two tropical polynomials, i.e., a tropical rational function. More precisely, if we view a neural network as a function $\nu : \mathbb{R}^d \to \mathbb{R}^p$, $x = (x_1, \dots, x_d) \mapsto (\nu_1(x), \dots, \nu_p(x))$, then each ν_i is a tropical rational function. Hence-

1. In this work, unless specified otherwise, a neural network will always mean a feedforward neural network with rectified linear units.

forth, statements of the form "a neural network is (some type of real-valued function)" are to be interpreted in the coordinatewise sense — every output node of the neural network is such a function of the input nodes.

In fact, for the special case $p = 1$, which arises in neural networks for classification problems, we show that:

> *the family of functions represented by feedforward neural networks with rectified linear units is identical to the family of tropical rational functions.*

Our main goal is to demonstrate how the new theory can be used to analyze neural networks. Among other things this allows us to capture the complexity of the function represented by a neural network and the complexity of the decision boundary derived from it in classification problems, or more importantly, how such complexity changes as the number of layers increase. The complexity of the function represented by a neural network can be captured by the number of linear regions the neural network has which can be investigated by counting the number of vertices on the polytopes constructed from the tropical polynomials associated with the neural network. In classification problems, the decision boundary derived from a neural network partitions the input space into regions and it is a subset of tropical hypersurface — a piecewise linear polyhedral surface.

Tropical geometry allows us to derive an upper bound on these numbers and thereby measure the complexity of its geometry. When the depth and input dimension of our neural network is fixed, we find that this upper bound is polynomial in the number of nodes on each layer but otherwise it is exponential. Intuitively, a complex classification problem — one that has many "exceptions to the rules" — requires a complex decision boundary to separate.

2.2 Tropical algebra

We give a brief review of tropical algebra and introduce some notations from tropical geometry to be used in the rest of the chapter. Tropical algebraic geometry is an active new area in mathematics. Roughly speaking it is an analogue of classical algebraic geometry over $\mathbb{C}$, the field of complex numbers, but where one replaces $\mathbb{C}$ by a semiring called the tropical ring, to be defined below. In addition to providing many analogues of well-known results in classical algebraic geometry, it also serves as a powerful tool for establishing results in algebraic geometry that are much more difficult to obtain using more standard techniques. See Itenberg et al. (2009); Maclagan and Sturmfels (2015) for a comprehensive overview of the subject.

As our main goal is to describe neural networks in the language of tropical algebra and to make use of a number of well established results in tropical algebraic geometry to gain further understanding of operations in neural networks, we just need to know some very basic objects. The most basic and fundamental of which is the *tropical semiring* $\big(\mathbb{R}\cup\{-\infty\}, \oplus, \odot\big)$, often also denoted by $\mathbb{T}$.

Let $\mathbb{N} = \{n \in \mathbb{Z} : n \geq 0\}$. The two operations $\oplus$ and $\odot$, called *tropical addition* and *tropical multiplication* respectively, are defined as follows.

Definition 2.2.1 (Tropical sum, product, and quotient)**.** *For $x, y \in \mathbb{R}$, their tropical sum is*

$$x \oplus y \coloneqq \max\{x, y\};$$

their tropical product is

$$x \odot y \coloneqq x + y,$$

and the tropical quotient of x over y is

$$x \oslash y \coloneqq x - y.$$

For any $x \in \mathbb{R}$, we have

$$-\infty \oplus x = x, \quad -\infty \odot x = -\infty, \quad 0 \odot x = x.$$

Thus $-\infty$ is the tropical additive identity and 0 is the tropical multiplicative identity. Furthermore, these operations satisfy the usual laws of arithmetic:

- Associativity: For any $x, y, z \in \mathbb{R}$,

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z,$$
$$x \odot (y \odot z) = (x \odot y) \odot z.$$

- Commutativity: For any $x, y \in \mathbb{R}$,

$$x \oplus y = y \oplus x,$$
$$x \odot y = y \odot x.$$

- Distributivity: For any $x, y, z \in \mathbb{R}$,

$$x \odot (y \oplus z) = (x \odot y) \oplus (x \odot z).$$

$\mathbb{R} \cup \{-\infty\}$ is therefore a semiring under the operations $\oplus$ and $\odot$. While it is not a ring (lacks additive inverse), one may nonetheless generalize many algebraic objects (e.g., matrices, polynomials, tensors, etc) and notions (e.g., rank, determinant, degree, etc) over the tropical

semiring — the study of these, in a nutshell, constitutes the subject of tropical algebra.

For an integer $a \in \mathbb{N}$, raising $x \in \mathbb{R}$ to the ath power is of course the same as multiplying x to itself a times. When usual multiplication is replaced by tropical multiplication, we obtain the operation of taking *tropical power*:

$$x^{\odot a} := \underbrace{x \odot \cdots \odot x}_{a \text{ times}} = a \cdot x, \tag{2.1}$$

where the last $\cdot$ denotes usual product of real numbers. We extend tropical power to $\mathbb{R} \cup \{-\infty\}$ by defining, for any $a \in \mathbb{N}$,

$$-\infty^{\odot a} := \begin{cases} -\infty & \text{if } a > 0, \\ 0 & \text{if } a = 0. \end{cases}$$

Observe that a tropical semiring, while not a field, possesses one quality of a field, namely, every $x \in \mathbb{R}$ has a tropical multiplicative inverse, which is of course just the usual additive inverse of x,

$$x^{\odot(-1)} := -x.$$

One may therefore also raise $x \in \mathbb{R}$ to a negative power $a \in \mathbb{Z}$ by raising its tropical multiplicative inverse $-x$ to the positive power $-a$,

$$x^{\odot a} = (-x)^{\odot(-a)}. \tag{2.2}$$

As is the case in usual real arithmetic, the tropical additive inverse $-\infty$ does not have a tropical multiplicative inverse and $-\infty^{\odot a}$ is undefined for $a < 0$.

For notational simplicity, we will henceforth write x^a instead of $x^{\odot a}$ for tropical power (we will not have the occasion to use usual powers of real numbers in the remainder of this chapter). Evidently, tropical powers satisfy the following:

- For any $x, y \in \mathbb{R}$ and $a \in \mathbb{N}$,

$$(x \oplus y)^a = x^a \oplus y^a.$$

- For any $x, y \in \mathbb{R}$ and $a \in \mathbb{N}$,

$$(x \odot y)^a = x^a \odot y^a.$$

- For any $x \in \mathbb{R}$,

$$x^0 = 0.$$

- For any $x \in \mathbb{R}$ and $a, a' \in \mathbb{N}$,

$$(x^a)^{a'} = x^{a \cdot a'}.$$

- For any $x \in \mathbb{R}$ and $a, a' \in \mathbb{Z}$,

$$x^a \odot x^{a'} = x^{a+a'}.$$

- For any $x \in \mathbb{R}$ and $a, a' \in \mathbb{Z}$,

$$x^a \oplus x^{a'} = x^a \odot (x^{a-a'} \oplus 0) = x^a \odot (0 \oplus x^{a-a'}).$$

We are now in a position to define tropical polynomials and tropical rational functions. In the following, x and x_i will denote variables (i.e., indeterminates).

Definition 2.2.2 (Tropical monomial)**.** *A tropical monomial of d variables $x_1, \ldots, x_d$ is an*

expression of the form

$$c \odot x_1^{a_1} \odot x_2^{a_2} \odot \cdots \odot x_d^{a_d}$$

where $c \in \mathbb{R} \cup \{-\infty\}$ *and* $a_1, \ldots, a_d \in \mathbb{N}$. *As a convenient shorthand, we will also write a tropical monomial in multiindex notation as* cx^α *where* $\alpha = (a_1, \ldots, a_d) \in \mathbb{N}^d$ *and* $x = (x_1, \ldots, x_d)$. *It is also natural to write*

$$x^\alpha = 0 \odot x^\alpha$$

since 0 *is the tropical multiplicative identity.*

Definition 2.2.3 (Tropical polynomial)**.** *Following notations above, a tropical polynomial* $f(x) = f(x_1, \ldots, x_d)$ *is a finite tropical sum of tropical monomials*

$$f(x) = c_1 x^{\alpha_1} \oplus \cdots \oplus c_r x^{\alpha_r},$$

where $\alpha_i = (a_{i1}, \ldots, a_{id}) \in \mathbb{N}^d$ *and* $c_i \in \mathbb{R} \cup \{-\infty\}$, $i = 1, \ldots, r$. *We always assume that a monomial with a given multiindex appears at most once in the sum, i.e.,* $\alpha_i \neq \alpha_j$ *for any* $i \neq j$.

Definition 2.2.4 (Tropical rational function)**.** *Following notations above, a tropical rational function is a (usual) difference of two tropical polynomials* $f(x)$ *and* $g(x)$. *This is the natural tropical analogue of a rational function since*

$$f(x) - g(x) = f(x) \oslash g(x).$$

Henceforth we will denote a tropical rational function by $f \oslash g$, *where* f *and* g *are understood to mean tropical polynomial functions.*

It is routine to verify that the set of tropical polynomials $\mathbb{T}[x_1, \ldots, x_d]$ forms a semiring under the standard extension of $\oplus$ and $\odot$ to tropical polynomials, and likewise the set of tropical rational functions $\mathbb{T}(x_1, \ldots, x_d)$ forms a semifield. We regard a tropical polynomial $f = f \oslash 0$ as a special case[2] of a tropical rational function and thus $\mathbb{T}[x_1, \ldots, x_d] \subseteq \mathbb{T}(x_1, \ldots, x_d)$. Henceforth any result stated for a tropical rational function would implicitly also apply to a tropical polynomial.

A d-variate tropical polynomial $f(x)$ defines a function $f : \mathbb{R}^d \to \mathbb{R}$ that is a *convex function* in the usual sense as taking max and sum of convex functions preserve convexity Boyd and Vandenberghe (1993). As such, a tropical rational function $f \oslash g : \mathbb{R}^d \to \mathbb{R}$ is a *DC function* or *difference-convex function* Hartman et al. (1959); An and Tao (2005).

We will also have the occasion to use vector-valued tropical polynomials and tropical rational functions, defined formally below.

Definition 2.2.5 (Tropical polynomial map and tropical rational map)**.** *A function* $F : \mathbb{R}^d \to \mathbb{R}^p$, $x = (x_1, \ldots, x_d) \mapsto (f_1(x), \ldots, f_p(x))$, *is called a tropical polynomial map if each* $f_i : \mathbb{R}^d \to \mathbb{R}$ *is a tropical polynomial,* $i = 1, \ldots, p$, *and a tropical rational map if* $f_1, \ldots, f_p$ *are tropical rational functions. We denote the set of tropical polynomial maps by* $\mathfrak{H}(d, n)$ *and the set of tropical rational maps by* $\mathfrak{R}(d, n)$.

One may also view tropical polynomial maps as *tropical polynomial vector fields* and tropical rational maps as *tropical rational function vector fields* .

2.3 Tropical hypersurfaces

There are tropical analogues of many notions in classical algebraic geometry Itenberg et al. (2009); Maclagan and Sturmfels (2015). But for our goals in this work, it suffices to limit

2. This is not "division by zero" since the "tropical zero," i.e., the additive identity in $\mathbb{T}$, is not 0 but $-\infty$.

our discussions to *tropical hypersurfaces*, the tropical analogue of algebraic curves in classical algebraic geometry. In the following, we will briefly introduce tropical hypersurfaces and describe a few properties relevant to our subsequent discussions on deep neural networks.

Intuitively, the *tropical hypersurface* of a tropical polynomial f is the set of points x where f is not linear at x.

Definition 2.3.1 (Tropical hypersurface)**.** *Given a tropical polynomial*

$$f(x) = c_1 x^{\alpha_1} \oplus \cdots \oplus c_r x^{\alpha_r},$$

the tropical hypersurface of f is the set

$$\mathcal{T}(f) \coloneqq \big\{x \in \mathbb{R}^d : c_i x^{\alpha_i} = c_j x^{\alpha_j} = f(x) \text{ for some } \alpha_i \neq \alpha_j\big\}.$$

In other words, a tropical hypersurface comprises points x at which the value of f at x is attained by two or more monomials in f. It is often also characterized as the "corner locus" of the function f. A tropical hypersurface divides the domain of f into convex cells on each of which f is linear. These cells are convex polyhedrons, i.e., defined by a set of linear inequalities with integer coefficients: $\{x \in \mathbb{R}^d : Ax \leq b\}$ for $A \in \mathbb{Z}^{m \times d}$ and $b \in \mathbb{R}^m$. For example, the cell where a tropical monomial $c_j x^{\alpha_j}$ attains the maximum is given by $\{x \in \mathbb{R}^d : c_j + \alpha_j^\top x \geq c_i + \alpha_i^\top x \text{ for all } i \neq j\}$. Tropical hypersurfaces of polynomials in two variables (i.e., in $\mathbb{R}^2$) are called *tropical curves*.

Just like ordinary multivariate polynomials, every tropical polynomial comes with an associated *Newton polygon*.

Definition 2.3.2 (Newton polygon)**.** *The Newton polygon of a tropical polynomial $f(x) = c_1 x^{\alpha_1} \oplus \cdots \oplus c_r x^{\alpha_r}$ is the convex hull of $\alpha_1, \ldots, \alpha_r \in \mathbb{N}^d \subseteq \mathbb{R}^d$, regarded as points in $\mathbb{R}^d$,*

$$\Delta(f) \coloneqq \mathrm{Conv}\big\{\alpha_i \in \mathbb{R}^d : c_i \neq -\infty,\, i = 1, \ldots, r\big\}.$$

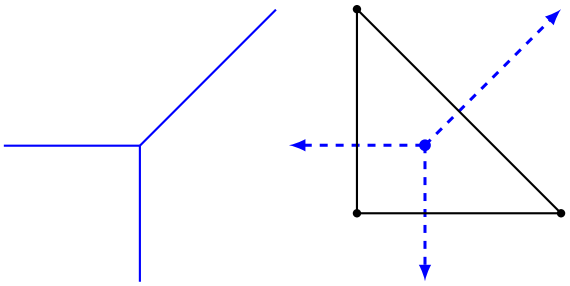


Figure 2.1: $x \oplus y \oplus 0$. Left: Tropical curve. Right: (Dual subdivision of) Newton polygon and tropical curve.

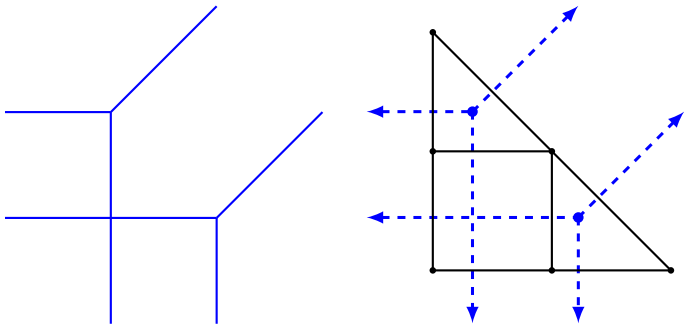


Figure 2.2: $1 \odot x_1^2 \oplus 1 \odot x_2^2 \oplus 2 \odot x_1 x_2 \oplus 2 \odot x_1 \oplus 2 \odot x_2 \oplus 2$. Left: Tropical curve. Right: Dual subdivision of Newton polygon and tropical curve.

A tropical polynomial f determines a subdivision of $\Delta(f)$, constructed as follows. First, lift each α_i from $\mathbb{R}^d$ into $\mathbb{R}^{d+1}$ by appending c_i as the additional coordinate. Denote the convex hull of the lifted $\alpha_1, \ldots, \alpha_r$ as

$$\mathcal{P}(f) \coloneqq \text{Conv}\{(\alpha_i, c_i) \in \mathbb{R}^d \times \mathbb{R} : i = 1, \ldots, r\}. \tag{2.3}$$

Next, let $\text{UF}\big(\mathcal{P}(f)\big)$ denote the collection of upper faces in $\mathcal{P}(f)$. Let $\pi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ be the projection that drops the last component of a point. The subdivision determined by f is then

$$\delta(f) \coloneqq \big\{\pi(p) \in \mathbb{R}^d : p \in \text{UF}\big(\mathcal{P}(f)\big)\big\}.$$

$\delta(f)$ forms a polyhedral complex with support $\Delta(f)$. By (Maclagan and Sturmfels, 2015,

Proposition 3.1.6), the tropical hypersurface $\mathcal{T}(f)$ is the $(d-1)$-skeleton of the polyhedral complex dual to $\delta(f)$. This means that each vertex in $\delta(f)$ corresponds to one "cell" in $\mathbb{R}^d$ where the function f is linear. As a consequence, the number of vertices in $\mathcal{P}(f)$ provides an upper bound on the number of linear regions of f.

For the case $d = 2$, the discussions above imply that $\mathcal{T}(f)$ is a planar graph dual to $\delta(f)$ in the following sense:

(i) each 2-dimensional face in $\delta(f)$ corresponds to a vertex in $\mathcal{T}(f)$;

(ii) each edge of a face in $\delta(f)$ corresponds to an edge in $\mathcal{T}(f)$. In particular, an edge from $\Delta(f)$ corresponds to an unbounded edge in $\mathcal{T}(f)$ while other edges correspond to bounded edges.

Figures 2.1 and 2.2 show examples of tropical curves and dual subdivisions of their Newton polygon for two tropical polynomials in two variables. We plot $\delta(f)$ and $\mathcal{T}(f)$ in the same figures to show their duality.

Figure 2.3 illustrates how we may find the dual subdivision for the tropical polynomial $f(x_1, x_2) = 1 \odot x_1^2 \oplus 1 \odot x_2^2 \oplus 2 \odot x_1 x_2 \oplus 2 \odot x_1 \oplus 2 \odot x_2 \oplus 2$. First, we find the convex hull

$$\mathcal{P}(f) = \text{Conv}\{(2, 0, 1), (0, 2, 1), (1, 1, 2), (1, 0, 2), (0, 1, 2), (0, 0, 2)\}.$$

Then, by projecting its upper envelope to $\mathbb{R}^2$, we obtain $\delta(f)$, the dual subdivision of Newton polygon.

2.3.1 Transformations of tropical polynomials

We will describe how $\mathcal{P}(f)$ transforms under taking tropical power, tropical sum, and tropical product. These results will be used in our analysis of neural networks.

The effect of taking tropical powers is straightforward.

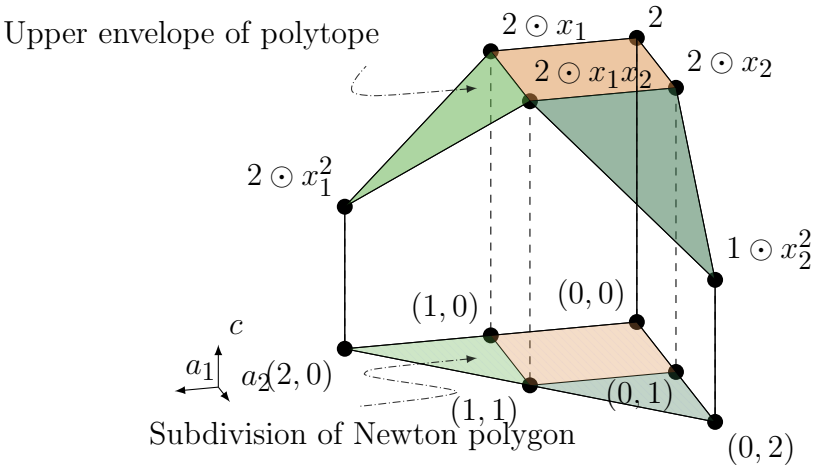


Figure 2.3: $1 \odot x_1^2 \oplus 1 \odot x_2^2 \oplus 2 \odot x_1 x_2 \oplus 2 \odot x_1 \oplus 2 \odot x_2 \oplus 2$. The dual subdivision can be obtained by projecting the edges on the upper faces of the polytope.

Proposition 2.3.1. *Let f be a tropical polynomial and a nonnegative integer $a \in \mathbb{N}$, we have*

$$\mathcal{P}(f^a) = a\mathcal{P}(f).$$

The polytope $a\mathcal{P}(f) = \{ax : x \in \mathcal{P}(f)\} \subseteq \mathbb{R}^{d+1}$ is a scaled version of $\mathcal{P}(f)$ having the same shape but different volume.

In order to describe the effect of tropical sum and product, we recall a few notions from convex geometry. The *Minkowski sum* $P_1 + P_2$ of two sets P_1 and P_2 in $\mathbb{R}^d$ is the set

$$P_1 + P_2 \coloneqq \big\{x_1 + x_2 \in \mathbb{R}^d : x_1 \in P_1, x_2 \in P_2\big\}.$$

Given $\lambda_1, \lambda_2 \geq 0$, the *weighted Minkowski sum* is

$$\lambda_1 P_1 + \lambda_2 P_2 \coloneqq \big\{\lambda_1 x_1 + \lambda_2 x_2 \in \mathbb{R}^d : x_1 \in P_1, x_2 \in P_2\big\}.$$

Minkowski sum is clearly commutative and associative and generalizes to sum of more than two objects. In particular, the Minkowski sum of line segments is called a *zonotope*.

Let $\mathcal{V}(P)$ denote the vertex set, i.e., the set of vertices, of any polytope P. It is easy to

see that the Minkowski sum of two polytopes is equal to the convex hull of the Minkowski sum of their vertex sets. i.e.,

$$P_1 + P_2 = \mathrm{Conv}\big(\mathcal{V}(P_1) + \mathcal{V}(P_2)\big).$$

With this observation, the following is immediate.

Proposition 2.3.2. *Let $f, g \in \mathfrak{H}(d, 1)$. Then*

$$\begin{aligned}\mathcal{P}(f \odot g) &= \mathcal{P}(f) + \mathcal{P}(g),\\ \mathcal{P}(f \oplus g) &= \mathrm{Conv}\big(\mathcal{V}(\mathcal{P}(f)) \cup \mathcal{V}(\mathcal{P}(g))\big).\end{aligned}$$

We reproduce below part of (Gritzmann and Sturmfels, 1993, Theorem 2.1.10), which we will later use for counting vertices in various polytopes.

Theorem 2.3.3 (Gritzmann–Sturmfels)**.** *Let $P_1, \ldots, P_k$ be polytopes in $\mathbb{R}^d$ and let m denote the total number of nonparallel edges of $P_1, \ldots, P_k$. Then the number of vertices of $P_1 + \cdots + P_k$ is bounded by*

$$2\sum_{j=0}^{d-1} \binom{m-1}{j}.$$

The upper bound is obtained if all P_i's are zonotopes and all their generating edges are in general position.

Corollary 2.3.4. *Let $P \subset \mathbb{R}^{d+1}$ be a zonotope generated by m line segments $P_1, \ldots, P_m$. Let $\pi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ be the projection operator defined previously that drop the last component of a point. Suppose P satisfies the conditions (A) the generating line segments are in general position; (B) the set of projected vertices $\{\pi(v) : v \in \mathcal{V}(P)\} \subset \mathbb{R}^d$ are in general position.*

Then P has

$$\sum_{j=0}^{d} \binom{m}{j}$$

vertices on its upper faces. If either of condition (A) or (B) is violated, then this becomes an upper bound.

Proof. Let V_1 be the set of vertices on the upper envelope of P and V_2 be the set of vertices on the lower envelope. By Theorem 2.3.3, P has $2\sum_{j=0}^{d}\binom{m-1}{j}$ vertices in total. Denote this number as n_1. We have $|V_1 \cup V_2| = n_1$. Meanwhile, it is well-known that zonotopes are centrally symmetric. Therefore, there are equal number of vertices on upper envelope and lower envelope, i.e., $|V_1| = |V_2|$. On the other hand, since the projected vertices are assumed to be in general position, P' is a d-dimensional zonotope generated by m non-parallel line segments. Hence, by Theorem 2.3.3 again P' has $2\sum_{j=0}^{d-1}\binom{m-1}{j}$ vertices. Denote this number by n_2. For any vertex $v \in P$, $\pi(v)$ is a vertex of P' if and only if v belong to both upper envelope and lower envelope, i.e., $v \in V_1 \cap V_2$. Therefore, the number of vertices on P' is equal to $|V_1 \cap V_2|$. Further we have $|V_1 \cap V_2| = n_2$. Consequently, by straight combinatorial argument, we know the number of vertices on the upper envelope is

$$\begin{aligned}
|V_1| &= \frac{1}{2}(|V_1 \cup V_2| - |V_1 \cap V2|) + |V_1 \cap V2| \\
&= \frac{1}{2}(n_1 - n_2) + n_2 \\
&= \sum_{j=0}^{d} \binom{m}{j}.
\end{aligned}$$

□

2.3.2 Partition of $\mathbb{R}^d$ by tropical rational maps

By construction, a tropical polynomial is a continuous piecewise linear function and therefore the notion of *linear region* applies. In fact it applies more generally to tropical ration

function and tropical rational maps. In the present section we will define and develop tropical characterization of the linear regions of $F \in \mathfrak{R}(d, n)$.

Linear regions of the rational tropical map is an important notion. In Section 2.6.3, we will rely on the number of linear regions to measure how complex the geometry of the decision boundary of a neural network is.

Within the tropical formalism we will call the number of linear regions of $F \in \mathfrak{R}(d, n)$ a *linear degree* of F, but to save up on the space we will call it *degree*. One should not confuse our definition with any other use of the term degree within tropical geometry. Why we choose this terminology should become clear by the end of this section when we look at the degree of composition of tropical rational maps, where linear degree behaves similar to that of a polynomials.

Definition 2.3.3 (Degree, linear regions)**.** *A* linear region *of $F \in \mathfrak{R}(d, m)$ is a maximal connected subset of the domain on which F is linear. Each such region is a polyhedron in $\mathbb{R}^d$. In addition,*

- *The number of linear regions of F is called the (linear)* degree *of F and is denoted by* $\deg(F)$*;*
- *The set of all linear regions is denoted by $\mathcal{D}(F) \coloneqq \{\mathcal{D}_1, \ldots, \mathcal{D}_{\deg(F)}\}$;*
- *The boundaries between adjacent linear regions is denoted by $\mathcal{T}(F)$. When $F \in \mathfrak{H}(d, m)$ this set is exactly the union of tropical hyper-surfaces $\mathcal{T}(F_i), i = 1, \ldots, m$. Therefore present definition of $\mathcal{T}(F)$ is an extension of the Definition 2.3.1.*

Composition of tropical rational maps plays a crucial role in neural network. In the rest of this section, we set up a bound on the degree of composition of two tropical rational maps in terms of the degrees of the individual components. To make our discussion solid, we resort to the notion of *general exponent* and *convex degree* which will be defined momentarily. First,

consider the family of tropical rational functions

$$F^\alpha := \alpha^\top F = \bigodot_{j=1}^{n} F_j^{a_j} \in \mathfrak{R}(d,1),$$

where $F \in \mathfrak{R}(d,n)$ is a tropical rational map and $\alpha = (a_1, \cdots, a_n) \in \mathbb{Z}^n$. While for some specific choice of α, F^α may have fewer linear regions than F, e.g, $\alpha = (0, \ldots, 0)$, we will see that, for general choice of α, F^α divides $\mathbb{R}^d$ into the same set of linear regions as F. To state this rigorously, we define *general exponent* as the following.

Definition 2.3.4 (General exponent)**.** *For any $F \in \mathfrak{R}(d,n)$, a vector $\alpha = (a_1, \cdots, a_n) \in \mathbb{Z}^n$ is said to be a general exponent of F if the set of linear regions in F^α and F are identical.*

The following lemma shows the existence of general exponent. Meanwhile, it shows that there always exists a non-negative general exponent for any $F \in \mathfrak{R}(d,n)$.

Lemma 2.3.5 (Existence of general exponent)**.** *Let $F \in \mathfrak{R}(d,n)$, then*

(i) $\deg(F^\alpha) = \deg(F)$ *if and only if α is a general exponent;*

(ii) *Moreover, we can always find a general exponent $\alpha \in \mathbb{N}^n$.*

Proof. Boundaries $\mathcal{T}(F^\alpha)$ and $\mathcal{T}(F)$ are formed by $x \in \mathbb{R}^d$ where F^α and F are non-differentiable. It follows that $\mathcal{T}(F^\alpha) \subseteq \mathcal{T}(F)$ which implies $\deg(F^\alpha) < \deg(F)$, unless $\mathcal{T}(F^\alpha) = \mathcal{T}(F)$. This completes the proof of (i). To show (ii), we demonstrate that we can find $\alpha \in \mathbb{N}^n$ such that for all $\mathcal{A}|\mathcal{B}$ adjacent polyhedra in $\mathcal{D}(F)$, the $d-1$ face separating those polyhedra in $\mathcal{T}(F)$ is present in $\mathcal{T}(F^\alpha)$ and so $\mathcal{T}(F^\alpha) \supseteq \mathcal{T}(F)$. Let the differentials of F on $\mathcal{A}$ and $\mathcal{B}$ be $D[F]|_\mathcal{A}, D[F]|_\mathcal{B} \in \mathbb{Z}^{n\times d}$. We must have $D[F]|_\mathcal{A} \neq D[F]|_\mathcal{B}$ (otherwise, $\mathcal{A}$ and $\mathcal{B}$ can be merged into a single polyhedron), so there are at least two columns $w_\mathcal{A}, w_\mathcal{B} \in \mathbb{Z}^n$, in $D[F]|_\mathcal{A}$ and $D[F]|_\mathcal{B}$, such that $v = w_\mathcal{A} - w_\mathcal{B} \neq 0$. Let I_F be the set of all such columns from all neighboring polyhedra in $\mathcal{D}[F]$. For any α satisfying $\alpha^\top v \neq 0$ we must have $D[F^\alpha]|_\mathcal{A} \neq D[F^\alpha]|_\mathcal{B}$. Therefore, it suffices to show that we can find $\alpha \in \mathbb{N}^n$ such that

$\alpha^\top v \neq 0$ for all $v \in I_F$. For a given $v \in I_F$, the set of vector α satisfying $\alpha^\top v = 0$ forms a co-dimension one hyperplane $P_v = \{\omega \in \mathbb{R}^n : \omega^\top v = 0\}$. To complete the proof choose $\alpha \in \big(\mathbb{N}^n \cap \big(\mathbb{R}^n \setminus (\cup_{v \in I_F} P_v)\big)\big)$. □

Lemma 2.3.5 allows us to translate the study of linear regions of tropical rational maps into real-valued tropical rational functions without losing any information about the degree of the maps which further allows us to apply tools developed in Section 2.3.1 to vector-valued tropical rational maps. In our next step we develop characterization of linear regions of tropical maps. While for $F \in \mathfrak{H}(d, m)$ the polyhedra forming $\mathcal{D}(F)$ are all convex, this is not necessarily the case for a general $F \in \mathfrak{R}(d, n)$. So geometric arguments that are easy to apply in convex setting are no loner valid. What saves the day is that there is a way to subdivide each of the non-convex linear regions into convex ones and get back into the convex settings. More specifically, we resolve this by defining *convex degree* as the following.

Definition 2.3.5 (Convex degree)**.** *Given $V \in \mathfrak{R}(d, n)$, we define, $\overline{\deg}(V)$, a convex degree of V, to be*

$$\overline{\deg}(V) \coloneqq \min\big\{\deg(P) : \mathcal{T}(V) \subseteq \mathcal{T}(P); P \in \mathfrak{H}(d, r), r \in \mathbb{N}\big\}.$$

That is, convex degree is the minimum number of convex linear regions among all tropical polynomial that subdivide linear regions of V.

For any $V \in \mathfrak{R}(d, n)$ there exists at least one tropical polynomial maps that subdivides $\mathcal{T}(V)$, therefore convex degree is well defined (e.g. a map $P \in \mathfrak{H}(d, 2n)$ constructed by concatenation $P = \{p_1, q_1, \ldots, p_n, q_n\}$, where $V_i = p_i \oslash q_i$.)

Since the linear regions of a tropical polynomial map are always convex, we have $\deg(P) = \overline{\deg}(P)$ for all tropical polynomial map P. Before we proceeding further we introduce one more notation:

Definition 2.3.6 (Restriction of tropical map to affine subspace)**.** *Let $F \in \mathfrak{R}(d,n)$, and let $m \leq d$. For any $\Omega \subset \mathbb{R}^d$, we will write $F|_\Omega$ to denote the restriction of map F to Ω. We will write $\overline{\deg}(F|_{\mathbb{A}^m})$ to denote the maximum convex degree obtained by restricting F to an m-dimensional affine space in $\mathbb{R}^d$. i.e., $\overline{\deg}(F|_{\mathbb{A}^m}) \coloneqq \max\{\overline{\deg}(F|_\Omega) : \Omega \subset \mathbb{R}^d$ is an m dimensional affine space$\}$.*

Provided with the definitions above, we are ready to show the main observation on the composition of tropical rational maps.

Theorem 2.3.6 (Degree of composition of rational maps)**.** *Given $V \in \mathfrak{R}(n,m)$ and $W \in \mathfrak{R}(d,n)$, Define $Z \in \mathfrak{R}(d,m)$*

$$Z_i(x) \coloneqq V_i \circ W,\ i = 1, \ldots, m$$

then

$$\deg(Z) \leq \overline{\deg}(Z) \leq \overline{\deg}(V|_{\mathbb{A}^d}) \cdot \overline{\deg}(W)$$

Proof. To prove the upper bound. We construct polynomials, $P(x) \in \mathfrak{H}(d,n)$ and $Q(y) \in \mathfrak{H}(n,1)$ that admit $\mathcal{T}(V \circ W) \subseteq \mathcal{T}(Q \circ P)$ and for which

$$\overline{\deg}(Q \circ P) \leq \overline{\deg}(V|_{\mathbb{A}^d}) \cdot \overline{\deg}(W).$$

We start with two tropical polynomials whose existence is insured by the definition of convex degree, i.e:

$$P'(x) : \mathbb{R}^d \to \mathbb{R},\ Q'(y) : \mathbb{R}^n \to \mathbb{R},$$

such that $\deg(P') = \overline{\deg}(W), \deg(Q') = \overline{\deg}(V)$, and which admit $\mathcal{T}(V) \subseteq \mathcal{T}(P')$, $\mathcal{T}(W) \subseteq \mathcal{T}(Q')$.

First, construct P as $P_i(x) := P'(x) \odot x^{\alpha_i}, i = 1, \ldots, n$, where the linear terms x^{α_i} are chosen so that non of the differentials in all combinations of adjacent linear regions of P_i are

identical. Second, construct $Q(y) = Q'(y) \odot y^{\beta}$, so that non of the gradients in linear regions of Q are zero and such that $\overline{\deg}(V|_{\mathbb{A}^d}) = \deg(Q|_{\mathbb{A}^d})$. This selection of Q and P insures that in the composition $Q \circ P$ the convex linear regions of P can only be subdivided by Q and no two adjacent linear regions are merged and therefor $\mathcal{T}(V \circ W) \subset \mathcal{T}(Q \circ P)$. Observe, that P breaks $\mathbb{R}^d$ into up to $\deg(W)$ convex polyhedra $\mathcal{D}_i, i = 1, \ldots, \deg(W)$. Then on each of those polyhedra $x \in \mathcal{D}_j$ $(P(x))_i, i = 1, \ldots, n$ are linear and $P(\mathcal{D}_j) = \{P(x)_1, \ldots, P(x)_n : x \in \mathcal{D}_i\}$ is a convex subset on a d dimensional affine space in $\mathbb{R}^n$. The tropical rational map Q divides $P(\mathcal{D}_i)$ into at most $\overline{\deg}(V|_{\mathbb{A}^d})$ linear regions. It follows that when confined to $\mathcal{D}_j$ the tropical rational map $Q \circ P$ can generate at most $\overline{\deg}(V|_{\mathbb{A}^d}) \cdot \overline{\deg}(W)$ linear regions, which completes the proof. □

We complete this section with a proposition that will be important to our study of decision boundaries produced by neural networks. In this proposition we look at a partition of the input space that arises when we compare a tropical rational function with a constant function.

Proposition 2.3.7 (Level sets)**.** *Let $f \oslash g \in \mathfrak{R}(d, 1)$ Then*

(i) *given a constant $c > 0$, the* level set

$$\mathcal{B} \coloneqq \{x \in \mathbb{R}^d : f(x) \oslash g(x) = c\}$$

partitions $\mathbb{R}^d$ into at most $\deg(f)$ connected polyhedral regions above c, and at most $\deg(g)$ such regions below c.

(ii) *Suppose $c \in \mathbb{R}$ is such that there is no tropical monomial in $f(x)$ that differs from any tropical monomial in $g(x)$ by c, the level set $\mathcal{B}$ is contained in tropical hypersurface,*

$$\mathcal{B} \subseteq \mathcal{T}(\max\{f(x),\, g(x) + c\}) = \mathcal{T}(c \odot g \oplus f).$$

Proof. We show that the bounds on the numbers of connected positive (i.e., above c) and negative (i.e., below c) regions are as claimed in (i).

The tropical hypersurface of f divide the space $\mathbb{R}^d$ into $\deg(f)$ convex regions $\mathcal{D}(f) = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{\deg(f)}\}$. On each $\mathcal{D}_i$, f is linear. Meanwhile, g is piecewise linear and convex over $\mathbb{R}^d$. Therefore, $f \oslash g = f - g$ is piecewise linear and concave on each of R_i. Since the level set $\{x : f(x) - g(x) = c\}$ and the superlevel set $\{x : f(x) - g(x) \geq c\}$ must be convex by the concavity of $f - g$, there is at most one positive region in each $\mathcal{D}_i$. Therefore, the total number of connected positive regions cannot exceed $\deg(f)$.

Similarly, the tropical curve of g partitions $\mathbb{R}^d$ into $\deg(g)$ convex regions on each of which $f \oslash g$ is convex. The same argument shows that the number of connected negative regions does not exceed $\deg(g)$.

For (ii) By rearranging terms, the level set becomes

$$\mathcal{B} = \big\{x \in \mathbb{R}^d : f(x) = g(x) + c\big\}.$$

Since $f(x)$ and $g(x) + c$ are both tropical polynomial, we have

$$f(x) = b_1 x^{\alpha_1} \oplus \cdots \oplus b_r x^{\alpha_r},$$
$$g(x) + c = c_1 x^{\beta_1} \oplus \cdots \oplus c_s x^{\beta_s},$$

with the appropriate multiindices $\{\alpha_i\}_{i=1}^r$, $\{\beta_i\}_{i=1}^s$, and real coefficients $\{b_i\}_{i=1}^r$, $\{c_i\}_{i=1}^s$. By the assumption on the monomials, we have that $x_0 \in \mathcal{B}$ only if there exist i, j so that $\alpha_i \neq \beta_j$ and $b_i x_0^{\alpha_i} = c_j x_0^{\beta_j}$. This completes the proof since if we combine the monomials of $f(x)$ and $g(x) + c$ by (tropical) summing them into a single tropical polynomial, $\max\{f(x),\, g(x) + c\}$, the above implies that on the level set, the value of the combined tropical polynomial is attained by at least two monomials and therefore $x_0 \in \mathcal{T}(\max\{f(x),\, g(x) + c\})$. □

2.4 Neural networks

In this section, we discuss neural networks that are slightly more general than ReLU-activated ones. However, we will restrict our attention to feedforward neural networks. We will use this short section to define different components in a neural network, primarily for the purpose of fixing notations and specifying the assumptions that we retain throughout this work.

2.4.1 Neural networks on DAGs

Viewed abstractly, a feedforward neural network is a vector-valued map

$$\nu : \mathbb{R}^d \to \mathbb{R}^p, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \mapsto \begin{bmatrix} \nu_1(x_1, \ldots, x_d) \\ \vdots \\ \nu_p(x_1, \ldots, x_d) \end{bmatrix} = z.$$

The architecture of the network is defined by a directed acyclic graph (DAG) $\mathcal{G} = (V, E)$ where E is the set of directed edges and V is the set of *nodes* (or *neuron*s) including d input nodes and p output nodes. For a node $v \in V$, we will use $\mathcal{N}_{\text{in}}(v) \coloneqq \{u \in V : (u \to v) \in E\}$ to denote the set of immediate predecessors and use $\mathcal{N}_{\text{out}}(v) \coloneqq \{u \in V : (v \to u) \in E\}$ to denote the set of immediate successors. Such a feedforward neural network is parameterized by *weights* assigned to edges $W = \{w_e : e \in E\}$ and *bias* associated with nodes $B = \{b_v : v \in V\}$. Given an input $x \in \mathbb{R}^d$, every node in a network takes input from its immediate predecessors, computes its output according a formula which will be introduced shortly and sends it to all its immediate successors. Let $\nu_v(x)$ be the output from node v. For any node v which is not an input node, the output of v is defined by

$$\nu_v(x) \coloneqq \sigma_v\Big(\sum_{u \in \mathcal{N}_{\text{in}}(v)} w_{u \to v} \nu_u(x) + b_v \Big),$$

where $\sigma_v(\cdot)$ is an *activation* function which is pre-defined for every node. The input nodes simply output the values they receive. We denote the length of the longest directed path in the network, or *depth*, by L.

Some popular choices of nonlinear activations σ_v in modern applications include:

- rectified linear unit (ReLU),
$$\sigma(x) = \max\{x, 0\};$$
- sigmoid function,
$$\sigma(x) = \frac{1}{1 + e^{-x}};$$
- hyperbolic tangent function,
$$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

The final output of a neural network $\nu(x)$ is usually fed into a *score function* $s : \mathbb{R}^p \to \mathbb{R}^m$ that is application specific. When used as an m-category classifier, s may be chosen, for example, to be a soft-max or sigmoidal function with

$$s_j(y) = \frac{e^{y_j}}{\sum_{k=1}^{p} e^{y_k}}, \quad j = 1, \ldots, m.$$

$s_j(\nu(x))$ is interpreted as the probability for the input x to belong to category j. For a two-category classification problem, m may be taken to be 1 (instead of 2) and so $s : \mathbb{R}^p \to \mathbb{R}$ is a scalar-valued function. The score, $s(\nu(x))$ is then interpreted as the probability for the input x to belong to one class and $1 - s(\nu(x))$ as that of belonging to the other class.

The score function is quite often regarded as the last layer of a neural network but this is purely a matter of convenience and we will not make this assumption. In fact, when a neural network is used for regression, there is no score function, or, equivalently, the score function is set to be the identity map, so that the range of the output has infinite support.

We will make the following mild assumptions on the architecture of our feedforward neural networks in this work, and will explain next why they are indeed mild.

(A) the weights w_e, $e \in E$ are integer-valued;

(B) the bias b_v, $v \in V$ are real-valued;

(C) the activation functions σ_v, $v \in V$ take the form

$$\sigma_v(x) \coloneqq \max\{x, t_v\},$$

where $t_v \in \mathbb{R} \cup \{-\infty\}$ is called a *threshold*.

Henceforth all neural networks in our subsequent discussions will be assumed to satisfy (A)–(C). Note that the activation function in (C) includes the ReLU as a special case but allows us to treat both ReLU and identity map on an equal footing: setting $t_v = 0$ gives the usual ReLU whereas setting $t_v = -\infty$, i.e., the tropical zero vector, gives the identity map.

While there is no loss of generality in (B), there is also little loss of generality in (A), i.e., in restricting the weights w_e from real numbers to integers, as:

(i) real weights can be approximated arbitrarily closely by rational weights;

(ii) one may then 'clear denominators' in these rational weights by multiplying them by the least common multiple of their denominators to obtain integer weights;

(iii) keeping in mind that scaling all weights and biases by the same positive constant has no bearing on the workings of a neural network.

2.4.2 Multilayer feedforward neural networks

One common architecture of feedforward neural networks is the multilayer feedforward neural network in which all nodes are arranged into multiple layers and edges exist only between

nodes on consecutive layers. In a multilayer feedforward network, the input nodes form the first layer while output nodes form the last layer. Edges are always directed to nodes closer to the output. If each node is connected to every node in the subsequent layer, the network is called *fully connected*. All multilayer feedforward neural networks that we will discuss later are all fully connected as a multilayer network with a missing edge can be treated as a fully connected network with zero weight on the corresponding edge.

Multilayer feedforward neural networks are arguably one of the simplest types of neural network but they capture many important properties of deep neural networks and also serve as crucial building blocks in numerous applications. The function represented by a multilayer network is given by a composition of functions

$$\nu = \sigma^{(L)} \circ \rho^{(L)} \circ \sigma^{(L-1)} \circ \rho^{(L-1)} \dots \circ \sigma^{(1)} \circ \rho^{(1)}.$$

The *preactivation* functions $\rho^{(1)}, \dots, \rho^{(L)}$ are affine transformations to be determined by *training* on given data. The *activation* functions $\sigma^{(1)}, \dots, \sigma^{(L)}$ are chosen and fixed in advanced. The output of the of lth layer will be denoted

$$\nu^{(l)} \coloneqq \sigma^{(l)} \circ \rho^{(l)} \circ \sigma^{(l-1)} \circ \rho^{(l-1)} \dots \circ \sigma^{(1)} \circ \rho^{(1)}.$$

This is a vector-valued map $\nu^{(l)} : \mathbb{R}^d \to \mathbb{R}^{n_l}$ where n_l is the dimension (number of nodes) of the lth layer. Let $n_0 = d$ and $\nu^{(0)}(x) \coloneqq x$. Also, let $n_L = p$. Then

$$\nu^{(l)} = \sigma^{(l)} \circ \rho^{(l)} \circ \nu^{(l-1)}, \quad l = 1, \dots, L.$$

Similar to the general DAG networks, the length of the longest path, which is equal to number of layers L, is the *depth* of the neural network.

The affine function $\rho^{(l)} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$ is parameterized by a *weight* matrix $A^{(l)} \in$

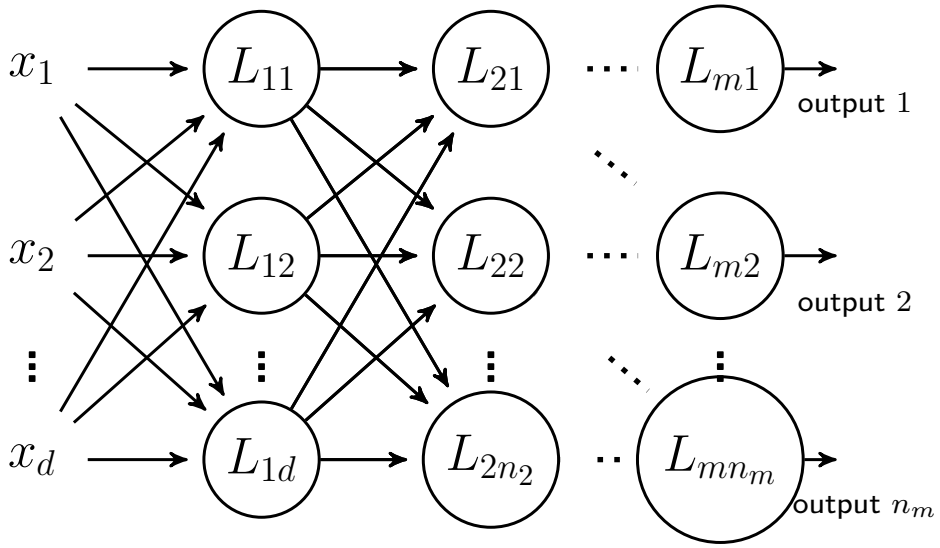


Figure 2.4: A general form of a multilayer feedforward neural network $\nu : \mathbb{R}^d \to \mathbb{R}^p$ with L layers.

$\mathbb{Z}^{n_l \times n_{l-1}}$ and a *bias* vector $b^{(l)} \in \mathbb{R}^{n_l}$:

$$\rho^{(l)}(\nu^{(l-1)}) \coloneqq A^{(l)}\nu^{(l-1)} + b^{(l)}.$$

The (i,j)th entry of the matrix $A^{(l)}$ will be denoted by $a_{ij}^{(l)}$, $i = 1, \ldots, n_l$, $j = 1, \ldots, n_{l-1}$; the ith entry of $b^{(l)}$ will be denoted by $b_i^{(l)}$, $i = 1, \ldots, n_l$. These are collectively called the *parameters* of the lth layer.

One may choose any activation function employed by a DAG network for a multilayer network as well, except that, for a vector input $x \in \mathbb{R}^d$, $\sigma(x)$ is understood to be in coordinatewise sense; so $\sigma : \mathbb{R}^d \to \mathbb{R}^d$. In practice, the activation functions are chosen to be of the same type in all layers, e.g., $\sigma^{(l)} : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$ are all ReLU's.

2.5 Tropical algebra of neural networks

We now describe our tropical formulation of a feedforward neural network satisfying (A)–(C). We will show that every such feedforward neural network is a tropical rational map.

2.5.1 Tropical characterization of feedforward neural networks defined on DAGs

We start by showing a tropical characterization for feedforward neural network defined on general DAGs. Formulation for multilayer feedforward neural network will follow naturally.

In general, a neural network function is nonconvex while tropical polynomials are always convex piece-wise linear functions. Since most nonconvex functions are a difference of two convex functions Hartman et al. (1959), a reasonable guess is that a feedforward neural network is the difference of two tropical polynomials, i.e., a tropical rational function. This is indeed the case, as we will see from the following.

Lemma 2.5.1. *Suppose v is a non-input node in a feedforward neural network under assumptions* (A)–(C). *Assume the outputs from all of v's immediate predecessors are tropical rational functions. i.e., for any $u \in \mathcal{N}_{in}(v)$, $\nu_u(x)$ can be written as $f_u(x) - g_u(x)$ where f_u and g_u are tropical polynomials. Meanwhile, let*

$$w^+_{u\to v} := \max\{w_{u\to v}, 0\}, \quad w^-_{u\to v} := \max\{-w_{u\to v}, 0\}.$$

Then the output from v is a tropical rational function $\nu_v(x) = f_v(x) - g_v(x)$ where

$$\begin{aligned}
f_v(x) &= \max\{h_v(x),\, g_v(x) + t_v\},\\
g_v(x) &= \sum_{u\in\mathcal{N}_{in}(v)} \big(w^-_{u\to v} f_u(x) + w^+_{u\to v} g_u(x)\big),\\
h_v(x) &= \sum_{u\in\mathcal{N}_{in}(v)} \big(w^+_{u\to v} f_u(x) + w^-_{u\to v} g_u(x)\big) + b_v.
\end{aligned}$$

In tropical arithmetic, these are

$$\begin{aligned}
f_v(x) &= h_v(x) \oplus g_v(x) \odot t_v,\\
g_v(x) &= \bigodot_{u\in\mathcal{N}_{in}(v)} \Big[f_u(x)^{w^-_{u\to v}}\Big] \odot \bigodot_{u\in\mathcal{N}_{in}(v)} \Big[g_u(x)^{w^+_{u\to v}}\Big],\\
h_v(x) &= \bigodot_{u\in\mathcal{N}_{in}(v)} \Big[f_u(x)^{w^+_{u\to v}}\Big] \odot \bigodot_{u\in\mathcal{N}_{in}(v)} \Big[g_u(x)^{w^-_{u\to v}}\Big] \odot b_v.
\end{aligned}$$

Proof. The output of v can be expressed as

$$\begin{aligned}
\nu_v(x) &= \sigma_v\Big(\sum_{u\in\mathcal{N}_{\text{in}}(v)} w_{u\to v}\nu_u(x) + b_v\Big)\\
&= \max\Big\{\sum_{u\in\mathcal{N}_{\text{in}}(v)} \big(w^+_{u\to v} - w^-_{u\to v}\big)(f_u(x) - g_u(x)) + b_v, t_v\Big\}\\
&= \max\Big\{\sum_{u\in\mathcal{N}_{\text{in}}(v)} \big(w^+_{u\to v} f_u(x) + w^-_{u\to v} g_u(x)\big) + b_v,\\
&\qquad \sum_{u\in\mathcal{N}_{\text{in}}(v)} \big(w^-_{u\to v} f_u(x) + w^+_{u\to v} g_u(x)\big) + t_v\Big\}\\
&\qquad\qquad - \sum_{u\in\mathcal{N}_{\text{in}}(v)} \big(w^-_{u\to v} f_u(x) + w^+_{u\to v} g_u(x)\big)
\end{aligned}$$

□

Note that the integer weights $w_{u\to v}$ have gone into the powers of tropical monomials

in g and h, which is why we require our weights to be integer-valued, although as we have explained at the end of Section 2.4, this requirement imposes little loss of generality. Provided with this lemma and the fact that outputs from input nodes are clearly tropical polynomials, we obtain the following theorem by induction.

Theorem 2.5.2 (Tropical characterization of neural networks defined on DAGs)**.** *A family of feedforward neural network under assumptions* (A)–(C) *is a function* $\nu : \mathbb{R}^d \to \mathbb{R}^p$ *whose coordinates are tropical rational functions of the input, i.e.,*

$$\nu(x) = F(x) \oslash G(x) = F(x) - G(x)$$

where F *and* G *are tropical polynomial maps. Thus* ν *is a tropical rational map.*

2.5.2 Tropical characterization of multilayer feedforward neural networks

Multilayer feedforward network is a special case of feedforward networks defined on DAGs. Therefore Lemma 2.5.1 and Theorem 2.5.2 apply. We show the tropical characterization of multilayer feedforward neural network in Proposition 2.5.3 and Corollary 2.5.4.

Proposition 2.5.3. *Let* $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{R}^m$ *be the parameters of the* $(l+1)$*th layer, and let* $t \in (\mathbb{R} \cup \{-\infty\})^m$ *be the threshold vector in the* $(l+1)$*th layer. If the nodes of the* l*th layer are given by tropical rational functions,*

$$\nu^{(l)}(x) = F^{(l)}(x) \oslash G^{(l)}(x) = F^{(l)}(x) - G^{(l)}(x),$$

i.e., each coordinate of $F^{(l)}$ *and* $G^{(l)}$ *is a tropical polynomial in* x*, then the nodes of the preactivation and the output of the* $(l+1)$*th layer are given respectively by tropical rational*

functions

$$\rho^{(l+1)}(x) \circ \nu^{(l)}(x) = H^{(l+1)}(x) - G^{(l+1)}(x),$$
$$\nu^{(l+1)}(x) = \sigma \circ \rho^{(l+1)} \circ \nu^{(l)}(x) = F^{(l+1)}(x) - G^{(l+1)}(x),$$

where

$$\begin{aligned} F^{(l+1)}(x) &= \max\big\{H^{(l+1)}(x),\, G^{(l+1)}(x) + t\big\}, \\ G^{(l+1)}(x) &= A_+ G^{(l)}(x) + A_- F^{(l)}(x), \\ H^{(l+1)}(x) &= A_+ F^{(l)}(x) + A_- G^{(l)}(x) + b. \end{aligned}$$

In tropical arithmetic, the recurrence above takes the form

$$\begin{aligned} F_i^{(l+1)} &= H_i^{(l+1)} \oplus (G_i^{(l+1)} \odot t_i), \\ G_i^{(l+1)} &= \left[\bigodot_{j=1}^{n} (F_j^{(l)})^{a_{ij}^-}\right] \odot \left[\bigodot_{j=1}^{n} (G_j^{(l)})^{a_{ij}^+}\right], \\ H_i^{(l+1)} &= \left[\bigodot_{j=1}^{n} (F_j^{(l)})^{a_{ij}^+}\right] \odot \left[\bigodot_{j=1}^{n} (G_j^{(l)})^{a_{ij}^-}\right] \odot b_i. \end{aligned} \tag{2.4}$$

Here the subscript i indicates the ith coordinate.

Corollary 2.5.4 (Tropical characterization of multilayer neural networks)**.** *A family of multilayer feedforward neural network under assumptions* (A)–(C) *is a function $\nu : \mathbb{R}^d \to \mathbb{R}^p$ whose coordinates are tropical rational functions of the input, i.e.,*

$$\nu(x) = F(x) \oslash G(x) = F(x) - G(x)$$

where F and G are tropical polynomial maps. Thus ν is a tropical rational map.

The case $p = 1$ is particularly important for classification problems and we will next

discuss this special case. By setting $t^{(0)} = \cdots = t^{(L-1)} = 0$ and $t^{(L)} = -\infty$, we obtain the following corollary.

Corollary 2.5.5. *Let $\nu : \mathbb{R}^d \to \mathbb{R}$ be an ReLU activated feedforward neural network with integer weights and linear output. Then ν is a tropical rational function.*

A more remarkable fact is that the converse of Corollary 2.5.5 also holds.

Theorem 2.5.6 (Equivalence of neural networks and tropical rational functions)**.**

(i) *Let $\nu : \mathbb{R}^d \to \mathbb{R}$. Then ν is a tropical rational function if and only if ν is a feedforward neural network with integer weights and ReLU activation.*

(ii) *A tropical rational function $f \oslash g$, as an L-layer neural network, has*

$$L \leq \max\{\lceil \log_2 r_f \rceil, \lceil \log_2 r_g \rceil\} + 2,$$

where r_f and r_g are the number of monomials in the tropical polynomials f and g respectively.

Proof. It remains to establish the "only if" part. We will write $\sigma_t(x) \coloneqq \max\{x, t\}$. Any tropical monomial $b_i x^{\alpha_i}$ is clearly such a neural network as

$$b_i x^{\alpha_i} = (\sigma_{-\infty} \circ \rho_i)(x) = \max\{\alpha_i^\top x + b_i, -\infty\}.$$

If two tropical polynomials p and q are represented as neural networks with l_p and l_q layers respectively,

$$p(x) = \big(\sigma_{-\infty} \circ \rho_p^{(l_p)} \circ \sigma_0 \circ \ldots \sigma_0 \circ \rho_p^{(1)}\big)(x),$$
$$q(x) = \big(\sigma_{-\infty} \circ \rho_q^{(l_q)} \circ \sigma_0 \circ \ldots \sigma_0 \circ \rho_q^{(1)}\big)(x),$$

then $(p \oplus q)(x) = \max\{p(x), q(x)\}$ can also be written as a neural network with $\max\{l_p, l_q\}+1$ layers:

$$(p \oplus q)(x) = \sigma_{-\infty}\big([\sigma_0 \circ \rho_1](y(x)) + [\sigma_0 \circ \rho_2](y(x)) - [\sigma_0 \circ \rho_3](y(x))\big),$$

where $y : \mathbb{R}^d \to \mathbb{R}^2$ is given by $y(x) = (p(x), q(x))$ and $\rho_i : \mathbb{R}^2 \to \mathbb{R}$, $i = 1, 2, 3$, are linear functions defined by

$$\rho_1(y) = y_1 - y_2, \quad \rho_2(y) = y_2, \quad \rho_3(y) = -y_2.$$

Thus, by induction, any tropical polynomial can be written as a neural network with ReLU activation. Observe also that if a tropical polynomial is the tropical sum of r monomials, then it can be written a neural network with no more than $\lceil \log_2 r \rceil + 1$ layers.

Next we consider a tropical rational function $(p \oslash q)(x) = p(x) - q(x)$ where p and q are tropical polynomials. Under the same assumptions, we can represent $p \oslash q$ as

$$(p \oslash q)(x) = \sigma_{-\infty}\big([\sigma_0 \circ \rho_4](y(x)) - [\sigma_0 \circ \rho_5](y(x)) + [\sigma_0 \circ \rho_6](y(x)) - [\sigma_0 \circ \rho_7](y(x))\big)$$

where $\rho_i : \mathbb{R}^2 \to \mathbb{R}^2$, $i = 4, 5, 6, 7$, are linear functions defined by

$$\rho_4(y) = y_1, \quad \rho_5(y) = -y_1, \quad \rho_6(y) = -y_2, \quad \rho_7(y) = y_2.$$

Therefore $p \oslash q$ is also a neural network with at most $\max\{l_p, l_q\} + 1$ layers.

Finally, if f and g are tropical polynomials that are respectively tropical sums of r_f and r_g monomials, then the discussions above show that $(f \oslash g)(x) = f(x) - g(x)$ is an ReLU neural network with at most $\max\{\lceil \log_2 r_f \rceil, \lceil \log_2 r_g \rceil\} + 2$ layers. □

By construction, a tropical rational function is a continuous piecewise linear function.

The continuity of a piecewise linear function automatically implies that each of the pieces on which it is linear is a polyhedral region. As we saw in Section 2.3, a tropical polynomial $f : \mathbb{R}^d \to \mathbb{R}$ gives a tropical hypersurface that divides $\mathbb{R}^d$ into *convex* polyhedral regions defined by linear inequalities with integer coefficients: $\{x \in \mathbb{R}^d : Ax \leq b\}$ with $A \in \mathbb{Z}^{m\times d}$ and $b \in \mathbb{R}^m$. A tropical rational function $f \oslash g : \mathbb{R}^d \to \mathbb{R}$ must also divide $\mathbb{R}^d$ into polyhedral regions on each of which $f \oslash g$ is linear, although these regions are *nonconvex* in general. We will show that the converse also holds — any continuous piecewise linear function with integer coefficients is a tropical rational function.

Proposition 2.5.7. *Let $\nu : \mathbb{R}^d \to \mathbb{R}$. Then ν is a continuous piecewise linear function with integer coefficients if and only if ν is a tropical rational function.*

Proof. It remains to establish the "if" part. Let $\mathbb{R}^d$ be partitioned into N polyhedral region on each of which ν restricts to a linear function

$$\ell_i(x) = a_i^\top x + b_i, \quad a_i \in \mathbb{Z}^d, \quad b_i \in \mathbb{R}, \quad i = 1, \ldots, L,$$

i.e., for any $x \in \mathbb{R}^d$, $\nu(x) = \ell_i(x)$ for some $i \in \{1, \ldots, L\}$. It follows from Tarela and Martinez (1999) that we can find N subsets of $\{1, \ldots, L\}$, denoted by S_j, $j = 1, \ldots, N$, so that ν has a representation

$$\nu(x) = \max_{j=1,\ldots,N} \min_{i\in S_j} \ell_i.$$

It is clear that each ℓ_i is a tropical rational function. Now for any tropical rational functions

p and q,

$$\begin{aligned}\min\{p, q\} &= -\max\{-p, -q\} \\ &= 0 \oslash [(0 \oslash p) \oplus (0 \oslash q)] \\ &= [p \odot q] \oslash [p \oplus q].\end{aligned}$$

Since $p \odot q$ and $p \oplus q$ are both tropical rational functions, so is their tropical quotient. By induction, $\min_{i \in S_j} \ell_i$ is a tropical rational function for any $j = 1, \ldots, N$, and therefore so is their tropical sum ν. □

Corollary 2.5.5, Theorem 2.5.6, and Proposition 2.5.7 collectively imply the equivalence of

(i) tropical rational functions with real coefficients,

(ii) continuous piecewise linear functions with integer coefficients,

(iii) neural networks satisfying assumptions (A)–(C).

An immediate advantage of the first characterization is that the set of tropical rational functions $\mathbb{T}(x_1, \ldots, x_d)$ has a semifield structure as we had pointed out in Section 2.2, a fact that we have implicitly used in the proof of Proposition 2.5.7. But more importantly, it is not the algebra but the algebraic geometry that this perspective brings — some rudimentary aspects of which we will see in the next four sections.

We would like to point out that an equivalence between ReLU-activated L-layer neural networks with *real* weights and d-variate continuous piecewise functions with *real* coefficients, and where $L \leq \lceil \log_2(d+1) \rceil + 1$, may be found in (Arora et al., 2018, Theorem 2.1). Note that our bound for L in Theorem 2.5.6(ii), which is in term of the number of monomials, is qualitatively different from this bound.

2.6 Tropical geometry of neural networks

Section 2.5 defines neural networks as objects in tropical algebra, a perspective that permits us to now study them via tropical algebraic geometry. Our tropical characterization in Section 2.5 applies to general feedforward networks defined on DAGs. In this section, we will focus on the analysis of multilayer feedforward neural network. As mentioned earlier, multilayer feedforward neural network captures many important properties of deep neural networks. Among other things, we will see that, in an appropriate sense, zonotopes form the geometric building blocks for neural networks (Section 2.6.2) and that the geometry of the decision boundary (Section 2.6.1) grows vastly more complex as the number of layers of the neural network increases, explaining why "deeper is better" (Section 2.6.3).

2.6.1 Decision boundaries of a neural network

We will focus on the case of two-category classification (also known as binary classification) for clarity. Suppose we would like to distinguish between images of, say, CATS and DOGS. As explained in Section 2.4, a neural network $\nu : \mathbb{R}^d \to \mathbb{R}^p$ together with a choice of score function $s : \mathbb{R}^p \to \mathbb{R}$ give us a classifier that takes an image encoded as $x \in \mathbb{R}^d$ and gives a score $s(\nu(x)) \in \mathbb{R}$ that represents the likelihood of x belonging to one class, say, CAT. If this value exceeds some decision threshold c, then x is a CAT image, and otherwise it is a DOG image. The space of all images is thereby partitioned into two disjoint subsets according to the outcome of such a prediction rule. The boundary in $\mathbb{R}^d$ between the two subsets is called the *decision boundary*. Connected regions that produce value above threshold and connected regions with the value below threshold will be called the *positive regions* and *negative regions* respectively.

Finding a mathematical characterization of decision boundaries is an important topic in neural networks and other areas of artificial intelligence. In the following, we use tropical geometry and insights from Section 2.5.2 to present a novel characterization of decision

boundaries for our family of neural networks.

By Theorem 2.5.6, a neural network $\nu : \mathbb{R}^d \to \mathbb{R}$ that satisfies assumptions (A)–(C) is a tropical rational function $f \oslash g$. Its decision boundary is a level set as defined in Proposition 2.3.7, which also gives us the following. By Theorem 2.5.6, a neural network $\nu : \mathbb{R}^d \to \mathbb{R}$ that satisfies assumptions (A)–(C) is a tropical rational function $f \oslash g$. Its decision boundary is a level set as defined in Proposition 2.3.7, which also gives us the following.

Corollary 2.6.1 (Tropical characterization of decision boundary)**.** *Let $\nu : \mathbb{R}^d \to \mathbb{R}$ be an L-layered neural network satisfying assumptions* (A)–(C) *and with $t^{(L)} = -\infty$. And let $\mathcal{B} = \{x \in \mathbb{R}^d \ : \ \nu(x) = s^{-1}(c)\}$ the decision boundary of ν with injective score function $s : \mathbb{R} \to \mathbb{R}$ and a decision threshold c in its range. If $\nu = f \oslash g$ where f and g are tropical polynomials, then*

(i) *the decision boundary $\mathcal{B}$ divides $\mathbb{R}^d$ into at most* $\deg(f)$ *connected positive regions and at most* $\deg(g)$ *connected negative regions;*

(ii) *and it satisfies*

$$\mathcal{B} \subseteq \mathcal{T}(s^{-1}(c) \odot g \oplus f) = \mathcal{T}(\max\{f(x),\, g(x) + s^{-1}(c)\}).$$

We would like to ultimately bound the number of positive, negative, and linear regions of a neural network ν. This requires us to examine the tropical geometry of ν more carefully in Section 2.6.2 before we derive these bounds in Section 2.6.3

2.6.2 Zonotopes as geometric building blocks of neural networks

From Section 2.3, we know that the number of regions that a tropical hypersurface $\mathcal{T}(f)$ divides the space into equals the number of vertices in the dual subdivision of the Newton

polygon associated with the tropical polynomial f. This allows us to bound the number of connected positive or negative regions of a neural network by bounding the number of vertices in the dual subdivision of the Newton polygon.

We will follow the notations in Proposition 2.5.3. The recurrent relation (2.4) describes how the tropical polynomials occurring in the $(l+1)$th layer depend on those in the lth layer. We may use this to answer the following:

Question. *How are the tropical hypersurfaces of the polynomials in the $(l+1)$th layer related to those in the lth layer?*

We observe that the polynomials in the $(l+1)$th layer are constructed from those on the lth layer via three operations: tropical sum, tropical product, and tropical exponentiation. Hence the question boils down to how these three operations transform the tropical hypersurfaces, which we have studied in Propositions 2.3.2 and 2.3.1. Thus we may deduce the next result.

Lemma 2.6.2. *Let $F_i^{(l)}$, $G_i^{(l)}$, $H_i^{(l)}$ be the tropical polynomials produced by the ith node in the lth layer of a neural network, i.e., they are defined by* (2.4)*. Then $\mathcal{P}\big(F_i^{(l)}\big)$, $\mathcal{P}\big(G_i^{(l)}\big)$, $\mathcal{P}\big(H_i^{(l)}\big)$ are subsets of $\mathbb{R}^{d+1}$ given as follows:*

(i) *$\mathcal{P}\big(G_i^{(1)}\big)$ and $\mathcal{P}\big(H_i^{(1)}\big)$ are points.*

(ii) *$\mathcal{P}\big(F_i^{(1)}\big)$ is a line segment.*

(iii) *$\mathcal{P}\big(G_i^{(2)}\big)$ and $\mathcal{P}\big(H_i^{(2)}\big)$ are zonotopes.*

(iv) *For $l \geq 1$,*

$$\mathcal{P}\big(F_i^{(l)}\big) = \mathrm{Conv}\big[\mathcal{P}\big(G_i^{(l)} \odot t_i^{(l)}\big) \cup \mathcal{P}\big(H_i^{(l)}\big)\big]$$

if $t_i^{(l)} \in \mathbb{R}$, and $\mathcal{P}\big(F_i^{(l)}\big) = \mathcal{P}\big(H_i^{(l)}\big)$ if $t_i^{(l)} = -\infty$.

(v) *For $l \geq 1$, $\mathcal{P}\big(G_i^{(l+1)}\big)$ and $\mathcal{P}\big(H_i^{(l+1)}\big)$ are weighted Minkowski sums of*

$$\mathcal{P}\big(F_1^{(l)}\big), \ldots, \mathcal{P}\big(F_{n_l}^{(l)}\big), \mathcal{P}\big(G_1^{(l)}\big), \ldots, \big(G_{n_l}^{(l)}\big),$$

given by

$$\mathcal{P}\big(G_i^{(l+1)}\big) = \sum_{j=1}^{n_l} a_{ij}^{-} \mathcal{P}\big(F_j^{(l)}\big) + \sum_{j=1}^{n_l} a_{ij}^{+} \mathcal{P}\big(G_j^{(l)}\big),$$
$$\mathcal{P}\big(H_i^{(l+1)}\big) = \sum_{j=1}^{n_l} a_{ij}^{+} \mathcal{P}\big(F_j^{(l)}\big) + \sum_{j=1}^{n_l} a_{ij}^{-} \mathcal{P}\big(G_j^{(l)}\big) + \big\{b_i e\big\},$$

where a_{ij} is the (i,j)th entry of the weight matrix $A^{(l+1)} \in \mathbb{Z}^{n_{l+1} \times n_l}$, b_i is the ith coordinate of the bias vector $b^{(l+1)} \in \mathbb{R}^{n_{l+1}}$, and $e = (0, \ldots, 0, 1) \in \mathbb{R}^{d+1}$.

An insight that we may deduce from Lemma 2.6.2 is that zonotopes play the role of building blocks in the tropical geometry of neural networks — $\mathcal{P}\big(F_i^{(l)}\big)$ and $\mathcal{P}\big(G_i^{(l)}\big)$ are all Minkowski sums of zonotopes. The study of zonotopes forms a rich subject in convex geometry and, in particular, are intimately related to hyperplane arrangements Greene and Zaslavsky (1983); Guibas et al. (2003); McMullen (1971); Holtz and Ron (2011). While the discussion here connects neural networks to this extensive body of work, its full implication remains to be explored.

2.6.3 Deeper is better: complexity of decision boundary

As we discussed in Section 2.1, we would like to track how the complexity of the function represented by a neural network changes through the layers and thereby understand the role of the number of layers — why a deeper neural network is better than a shallow one. We may rely on a few related quantities to quantify the "complexity" of the function: (a) number of linear regions, (b) number of positive regions and (c) number of negative regions. Here (b)

and (c) are specific to the scenario of binary classification. While all three quantities capture some aspects of how complex the function is, (a) has been adopted as the primary measure in studies of deep neural networks Montufar et al. (2014). The tropical geometric framework developed earlier allows us to study these three quantities by investigate the number of vertices on the polytopes associated the neural network, and further determine recursive relations for upper bounds on these quantities thereby obtain their orders of magnitudes.

Lemma 2.6.3. *Let $\sigma^{(l)}, \rho^{(l)}$ be the affine transformation and the activation of the lth layer of neural network.*

Set $V = \sigma^{(l)} \circ \rho^{(l)} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$. Assume $d \leq n_l$, then

$$\overline{\deg}(V|_{\mathbb{A}^d}) \leq \sum_{i=0}^{d} \binom{n_l}{i}$$

where $\overline{\deg}(V|_{\mathbb{A}^d})$ is as in Definition 2.3.6.

Proof. By definition, the convex degree of $V|_{\mathbb{A}^d}$ is defined as the maximum convex degree of tropical rational map $U : \mathbb{R}^d \to \mathbb{R}^{n_l}$ of the form

$$U_j(x) \coloneqq \sigma_j^{(l)} \circ \rho^{(l)} \circ (b_1 \odot x^{\alpha_1}, \ldots, b_{n_{l-1}} \odot x^{\alpha_{n_{l-1}}}),\ j = 1, \ldots, n_l.$$

For a general affine transformation $\rho^{(l)}$, $\rho^{(l)}(b_1 \odot x^{\alpha_1}, \ldots, b_{n_{l-1}} \odot x^{\alpha_{n_{l-1}}})$ evaluates to $(b'_1 \odot x^{\alpha'_1}, \ldots, b'_{n_l} \odot x^{\alpha'_{n_l}})$ for some $\{\alpha'_i, b'_i\}$. This yields $U_j(x) = \sigma_j^{(l)}(b'_1 \odot x^{\alpha'_1}, \ldots, b'_{n_l} \odot x^{\alpha'_{n_l}})$. Define $W : \mathbb{R}^d \to \mathbb{R}^{n_l}$ by $W(x) = (b'_1 \odot x^{\alpha'_1}, \ldots, b'_{n_l} \odot x^{\alpha'_{n_l}})$. Then we can write $U_j(x)$ as $U_j = \sigma_j^{(l)} \circ W$. This places us in the setting of Theorem 2.3.6. So we have $\overline{\deg}(V|_{\mathbb{A}^d}) = \overline{\deg}(\sigma^{(l)}|_{\mathbb{A}^d}) \cdot \overline{\deg}(W)$. Due to the linearity of W, we have $\overline{\deg}(W) = 1$ and further $\overline{\deg}(V|_{\mathbb{A}^d}) = \overline{\deg}(\sigma^{(l)}|_{\mathbb{A}^d})$. Now, the above is equivalent to the convex degree of a single layered neural network with n_l neurons. We calculate this convex degree next. Let $\nu(x) : \mathbb{R}^d \to \mathbb{R}^{n_l}$ be a single layered neural network with n_l neurons. Let $\gamma = (c_1, \ldots, c_{n_l})$ be a non-negative

general exponent for $\nu(x)$. We have

$$\bigodot_{j=1}^{n_l} \nu_j^{c_j} = \bigodot_{j=1}^{n_l} \left[\left(\bigodot_{i=1}^{d} b_i \odot x^{a_{ji}^+} \right) \oplus \left(\bigodot_{i=1}^{d} x^{a_{ji}^-} \right) \odot t_j \right]^{c_j} - \bigodot_{j=1}^{n_l} \left[\bigodot_{i=1}^{d} (x^{a_{ji}^-}) \right]^{c_j} \tag{2.5}$$

The last term is linear in x and we can drop it without affecting the calculation of convex degree. What remains is to bound the number of linear regions in the tropical polynomial

$$h = \bigodot_{j=1}^{n_l} \left[\left(\bigodot_{i=1}^{d} b_i \odot x^{a_{ji}^+} \right) \oplus \left(\bigodot_{i=1}^{d} x^{a_{ji}^-} \right) \odot t_j \right]^{c_j}.$$

We will find the bound by counting vertices in the polytope $\mathcal{P}(h)$. By Propositions 2.3.2 and 2.3.1 the polytope $\mathcal{P}(h)$ is given by the Minkowski sum

$$\sum_{j=1}^{n_l} c_j \mathcal{P}\left(\left(\bigodot_{i=1}^{d} b_i \odot x^{a_{ji}^+} \right) \oplus \left(\bigodot_{i=1}^{d} x^{a_{ji}^-} \right) \odot t_j \right).$$

So we investigate

$$\mathcal{P}\left(\left(\bigodot_{i=1}^{d} b_i \odot x^{a_{ji}^+} \right) \oplus \left(\bigodot_{i=1}^{d} x^{a_{ji}^-} \right) \odot t_j \right)$$

which, by Proposition 2.3.2 again, is given by $\mathrm{Conv}(\mathcal{V}(\mathcal{P}(p(x))) \cup \mathcal{V}(\mathcal{P}(q(x))))$ with

$$p(x) = \left(\bigodot_{i=1}^{d} b_i \odot x^{a_{ji}^+} \right) \text{ and } q(x) = \left(\bigodot_{i=1}^{d} x^{a_{ji}^-} \right) \odot t_j.$$

$p(x)$ and $q(x)$ are tropical monomials of x. Therefore $\mathcal{P}(p)$ and $\mathcal{P}(q)$ are points in $\mathbb{R}^{d+1}$. Further, $\mathrm{Conv}(\mathcal{V}(\mathcal{P}(p)) \cup \mathcal{V}(\mathcal{P}(q)))$ is a line in $\mathbb{R}^{d+1}$. Hence $\mathcal{P}(h)$ is a zonotope constructed by Minkowski some of n_l line segments in $\mathbb{R}^{d+1}$. Finally, the proof is completed by using Corollary 2.3.4. □

Theorem 2.6.4. *Let $\nu : \mathbb{R}^d \to \mathbb{R}^{n_L}$ be L layered neural network satisfying assumptions* (A)–(C). *And let $G^{(l)}$, $F^{(l)}$, $H^{(l)}$ and $\nu^{(l)}$ be defined as in Proposition 2.5.3. Assume $n_l \geq d$,*

$l = 1, 2, \ldots, L$. *Then*

(i) $\deg(G^{(1)}) = \deg(H^{(1)}) = \overline{\deg}(\nu^{(0)}) = 1$;

(ii) $\overline{\deg}(\nu^{(l+1)}) \leq \overline{\deg}(\nu^{(l)}) \cdot \sum_{i=0}^{d} \binom{n_{l+1}}{i}$;

(iii) *Write* $W^{(l)} = \left[H^{(l)}; G^{(l)}\right] \in \mathfrak{H}(d, 2n_l)$ *for the concatenation of* $G^{(l)}$ *and* $H^{(l)}$, *then* $\overline{\deg}(W^{(1)}) = 1$ *and*

$$\overline{\deg}(W^{(l+1)}) \leq \overline{\deg}(W^{(l)}) \cdot \sum_{i=0}^{d} \binom{n_l}{i}.$$

Proof. For (i) we have $G^{(1)} = A_{-}^{(1)} x$ and $H^{(1)} = A_{+}^{(1)} x + b^{(1)}$ both are linear in x, therefore $\deg(G^{(1)}) = \deg(H^{(1)}) = 1$. The bound on $\overline{\deg}(\nu^{(1)})$ follows from the proof of Lemma 2.6.3. To show (ii), recall that $\nu^{(l)} = (\sigma^{(l)} \circ \rho^{(l)}) \circ \nu^{(l-1)}$. And this inequality can be obtained by Theorem 2.3.6 and Lemma 2.6.3.

The base case in (iii) is immediately since $H^{(1)}, G^{(1)}$ are linear. For the induction step, substitute $G^{(l)}, H^{(l)}$ for $F^{(l)}$ in the recurrence (2.4), to obtain

$$G_j^{(l+1)} = \left[\bigodot_{i=1}^{n_l} \left(H_i^{(l)} \oplus (G_i^{(l)} \odot t_i)\right)^{a_{ji}^{-}} \right] \odot \left[\bigodot_{i=1}^{n_l} (G^{(l)})^{a_{ji}^{+}} \right],$$
$$H_j^{(l+1)} = \left[\bigodot_{i=1}^{n_l} \left(H_i^{(l)} \oplus (G_i^{(l)} \odot t_i)\right)^{a_{ji}^{+}} \right] \odot \left[\bigodot_{i=1}^{n_l} (G^{(l)})^{a_{ji}^{-}} \right] \odot b_j,$$

where $a_{ij}^{+}, a_{ij}^{-}, b_j$ are parameters of $l+1$th layer and t_i is threshold of lth layer, $i = 1, \ldots n_l$ and $j = 1, \ldots, n_{l+1}$. To find the convex rank of $W^{(l+1)}$ we look at $(W^{(l+1)})^{\alpha}$ for a general exponent α. After some basic algebra, we can bound $\overline{\deg}(W^{(l+1)})^{\alpha}$ by the convex rank of the composition $(V \circ W^{(l)})(x) \in \mathfrak{R}(d, 1)$ where $V \in \mathfrak{H}(2n_l, 1)$ is given by

$$V(y) \coloneqq \left(\bigodot_{i=1}^{n_l} (y_i \oplus y_{i+n_l} \odot t_i)^{r_{ji}} \right) \odot \left(\bigodot_{i=1}^{n_l} (y_{i+n_l})^{s_{ji}} \right), \quad \text{for } y \in \mathbb{R}^{2n_l},$$

for arbitrary exponents $r_{ij}, s_{ij} \in \mathbb{Z}$, $i = 1, \ldots, n_l, j = 1, \ldots, n_{l+1}$. Similarly to the the proof

in Lemma 2.6.3, the convex degree of V restricted to d dimensional affine space is

$$\overline{\deg}(V|_{\mathbb{A}^d}) \leq \sum_{i=0}^{d} \binom{n_l}{i}$$

And so by Theorem 2.3.6

$$\overline{\deg}((V \circ W^{(l)})(x)) \leq \sum_{i=0}^{d} \binom{n_l}{i} \overline{\deg}(W^{(l)}).$$

the proof is complete since

$$\overline{\deg}(W^{(l+1)}) \leq \overline{\deg}((V \circ W^{(l)})(x)).$$

□

Observe that $\deg\big(H^{(l)}\big) \leq \deg\big(\big[H^{(l)}; G^{(l)}\big]\big)$ and similarly for $G^{(l)}$, we are now at the position where Theorem 2.6.4 and Corollary 2.6.1 yield

Corollary 2.6.5. *Let $\nu : \mathbb{R}^d \to \mathbb{R}$ be L-layered real valued feedforward neural network as in* (A)–(C). *Set $t^{(L)} = -\infty$ and $n_l \geq d$, $l = 1, 2, \ldots, L-1$. Then*

1. *$\nu(x)$ has at most*

$$\prod_{l=1}^{L-1} \sum_{i=0}^{d} \binom{n_l}{i}$$

linear regions. In particular, when $n_l = n$ for $l = 1, \ldots, L-1$, the number of linear regions of $\nu^{(L)}(x)$ cannot exceed $\mathcal{O}\big(n^{d(L-1)}\big)$.

2. *For any constant $c \in \mathbb{R}$, the decision boundary $\{x \in \mathbb{R}^d : \nu(x) = c\}$ divides the space $\mathbb{R}^d$ into at most*

$$\prod_{l=1}^{L-1} \sum_{i=0}^{d} \binom{n_l}{i}$$

connected positive (negative) regions.

Proof. (1) follows from Theorem 2.6.4 (ii) immediately. For (2), since $W^{(l)}$ is the concatenation of $H^{(l)}$ and $G^{(l)}$, the number of linear regions in $W^{(l)}$ is an upper bound on the number of linear regions in $H^{(l)}$ and $G^{(l)}$. Meanwhile, by construction, we have $\nu(x) = f(x) - g(x)$ with $f(x) = H^{(L)}$ and $g(x) = G^{(L)}$. Together with Corollary 2.6.1 (i), we reach the bound on the number of connected positive (negative) regions. □

The analysis in this section implies that a deeper network is able to create tropical polynomials with more linear regions and decision boundaries with more complicated geometry. In particular, we conjecture that the number of layers for a feedforward network plays a similar role as the degree of regular polynomials: polynomials with a higher degree leads to algebraic curves with more complicated geometry while tropical polynomials computed by a deeper network produces tropical curves that divide the space into more linear regions.

2.6.4 Example

For concreteness, we illustrate our preceding discussions in Section 2.5 with a two-layer example. Let $\nu : \mathbb{R}^2 \to \mathbb{R}$ be with $n_0 = 2$ input nodes, $n_1 = 5$ nodes in the first layer, and $n_2 = 1$ nodes in the output:

$$y = \nu^{(1)}(x) = \max\left\{ \begin{bmatrix} -1 & 1 \\ 1 & -3 \\ 1 & 2 \\ -4 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 2 \\ 0 \\ -2 \end{bmatrix}, 0 \right\},$$

$$\nu^{(2)}(y) = \max\{y_1 + 2y_2 + y_3 - y_4 - 3y_5, 0\}.$$

We first express $\nu^{(1)}$ and $\nu^{(2)}$ as tropical rational maps,

$$\nu^{(1)} = F^{(1)} \oslash G^{(1)}, \quad \nu^{(2)} = f^{(2)} \oslash g^{(2)},$$

where

$$\begin{aligned} y &\coloneqq F^{(1)}(x) = H^{(1)}(x) \oplus G^{(1)}(x), \\ z &\coloneqq G^{(1)}(x) = \begin{bmatrix} x_1 \\ x_2^3 \\ 0 \\ x_1^4 \\ 0 \end{bmatrix}, \qquad H^{(1)}(x) = \begin{bmatrix} 1 \odot x_2 \\ (-1) \odot x_1 \\ 2 \odot x_1 x_2^2 \\ x_2 \\ (-2) \odot x_1^3 x_2^2 \end{bmatrix}, \end{aligned}$$

and

$$\begin{aligned} f^{(2)}(x) &= g^{(2)}(x) \oplus h^{(2)}(x), \\ g^{(2)}(x) &= y_4 \odot y_5^3 \odot z_1 \odot z_2^2 \odot z_3 \\ &= (x_2 \oplus x_1^4) \odot ((-2) \odot x_1^3 x_2^2 \oplus 0)^3 \odot x_1 \odot (x_2^3)^2, \\ h^{(2)}(x) &= y_1 \odot y_2^2 \odot y_3 \odot z_4 \odot z_5^3 \\ &= (1 \odot x_2 \oplus x_1) \odot ((-1) \odot x_1 \oplus x_2^3)^2 \odot (2 \odot x_1 x_2^2 \oplus 0) \odot x_1^4. \end{aligned}$$

The monomials occurring in ${G_j}^{(1)}$ and ${H_j}^{(1)}$ have the form $c x_1^{a_1} x_2^{a_2}$. Therefore $\mathcal{P}\big(G_j^{(1)}\big)$ and $\mathcal{P}\big(H_j^{(1)}\big)$, $j = 1, \ldots, 5$, are points in $\mathbb{R}^3$.

Since $F^{(1)} = G^{(1)} \oplus H^{(1)}$, $\mathcal{P}(F_j^{(1)})$ is the convex hull of two points, hence a line segment in $\mathbb{R}^3$. The Newton polygons (which is equal to their subdivisions in this case) associated with $F_j^{(1)}$ are obtained by projecting these line segments back to the plane of (a_1, a_2). See the figure on the left of Fig 2.5.

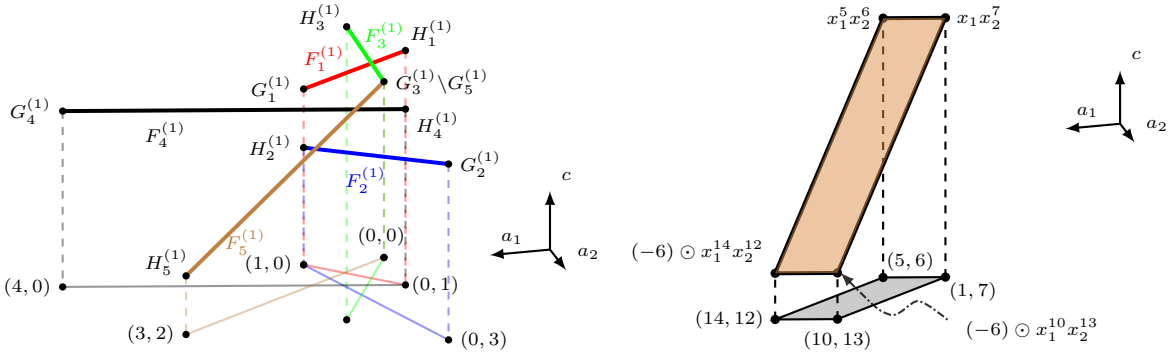


Figure 2.5: Illustration of polytopes and dual subdivisions associated with neural network functions. Left: $\mathcal{P}(F^{(1)})$ and dual subdivision of $F^{(1)}$. Right: $\mathcal{P}(g^{(2)})$ and dual subdivision of $g^{(2)}$. In both figures, dual subdivisions have been translated along $-c$ direction (downwards) and separated from the polytopes for illustrative purposes.

These line segments in $\mathcal{P}(F_j^{(1)})$'s and vertices in $\mathcal{P}(G_j^{(1)})$'s serve as the building blocks of $\mathcal{P}(h^{(2)})$ and $\mathcal{P}(g^{(2)})$. $\mathcal{P}(h^{(2)})$ and $\mathcal{P}(g^{(2)})$ are constructed by taking the weighted Minkowski sum of polytopes from $\{\mathcal{P}(F_j^{(1)}) : j = 1, \cdots, 5\} \cup \{\mathcal{P}(G_j^{(1)}) : j = 1, \cdots, 5\}$,

$$\mathcal{P}(h^{(2)}) = \mathcal{P}(F_4^{(1)}) + 3\mathcal{P}(F_5^{(1)}) + \mathcal{P}(G_1^{(1)}) + 2\mathcal{P}(G_2^{(1)}) + \mathcal{P}(G_3^{(1)}),$$
$$\mathcal{P}(g^{(2)}) = \mathcal{P}(F_1^{(1)}) + 2\mathcal{P}(F_2^{(1)}) + \mathcal{P}(F_3^{(1)}) + \mathcal{P}(G_4^{(1)}) + 3\mathcal{P}(G_5^{(1)}).$$

See the right panel in Fig 2.5 for the illustration of $\mathcal{P}(g^{(2)})$ and the subdivision of Newton polygon associated with $g^{(2)}$. $\mathcal{P}(h^{(2)})$ is shown in Fig 2.6.

Lastly, $\mathcal{P}(f^{(2)})$ is constructed by taking the convex hull of the union of $\mathcal{P}(g^{(2)})$ and $\mathcal{P}(h^{(2)})$. See the right panel in Fig 2.6. The dual subdivision of Newton polygon associated with $\mathcal{P}(f^{(2)})$ is obtained by projecting the upper faces of $\mathcal{P}(f^{(2)})$ to the plane of (a_1, a_2).

2.7 Conclusion

We formulated feedforward neural networks with rectified linear units as tropical rational functions. This formulation establishes the connection between tropical geometry and neural

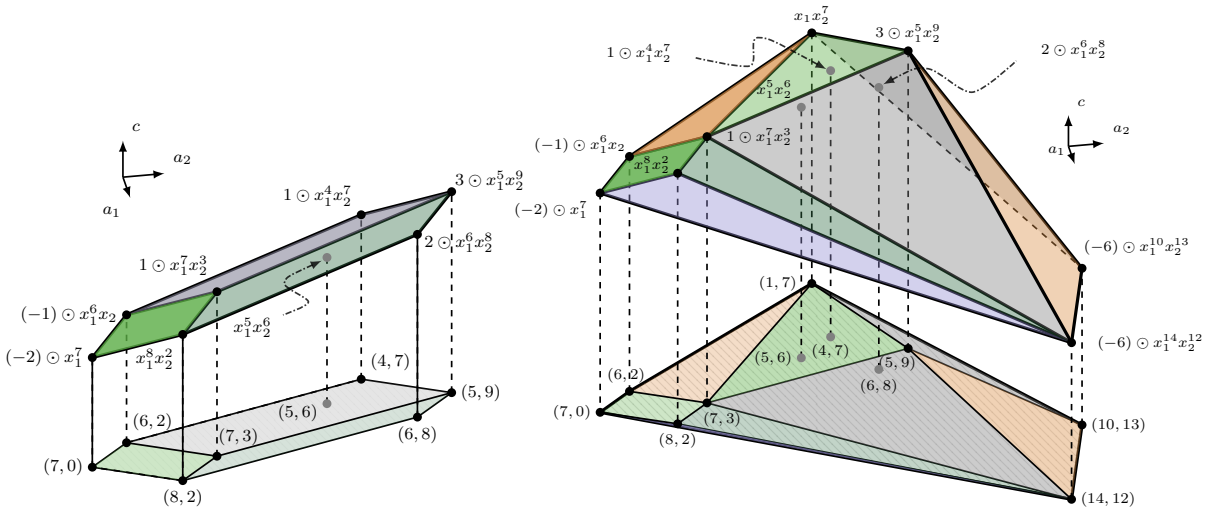


Figure 2.6: Illustration of polytopes and dual subdivisions associated with tropical polynomials of the second layer. Left: The polytope associated with $h^{(2)}$ and its subdivision. Right: $\mathcal{P}(f^{(2)})$ and dual subdivision of $f^{(2)}$. In both figures, dual subdivisions have been translated along $-c$ direction (downwards) and separated from the polytopes for illustrative purposes.

network. A direct implication is that a neural network with one hidden layer has deep connections to zonotopes which serve as building blocks for a deeper network. Further, such a relation is helpful to understand the family of functions representable by a deep neural network and provides a new view on the complexity and structure of neural networks. We showed that the study on decision boundaries and number of linear regions of a neural network can be transferred to the study on tropical hypersurfaces, subdivision of Newton polygon and the family of polytopes constructed from zonotopes. As an application, we showed that the number of linear regions and number of connected positive (negative) regions in classification problems are all at most polynomial in number of nodes on each layer of the network. Although our analysis is rather basic, we hope it will inspire and foster research for understanding neural networks from this new perspective.

Chapter 3

Knowledge Graph Embedding and Question Answering

3.1 Introduction

There is a growing interest in incorporating external memory into neural networks. For example, memory networks (Weston et al., 2014; Sukhbaatar et al., 2015) are equipped with static memory slots that are content or location addressable. Neural Turing machines (Graves et al., 2014) implement memory slots that can be read and written as in Turing machines (Turing, 1938) but through differentiable attention mechanism.

Each memory slot in these models stores a vector corresponding to a continuous representation of the memory content. In order to recall a piece of information stored in memory, attention is typically employed. Attention mechanism introduced by Bahdanau et al. (2014) uses a network that outputs a discrete probability mass over memory items. A memory read can be implemented as a weighted sum of the memory vectors in which the weights are given by the attention network. Reading out a single item can be realized as a special case in which the output of the attention network is peaked at the desired item. The attention network may depend on the current context as well as the memory item itself. The attention model is called location-based and content-based, if it depends on the location in the memory and the stored memory vector, respectively.

Knowledge bases, such as WordNet and Freebase, can also be stored in memory either

through an explicit knowledge base embedding (Bordes et al., 2011; Nickel et al., 2011; Socher et al., 2013) or through a feedforward network (Bordes et al., 2015).

When we embed entities from a knowledge base in a continuous vector space, if the capacity of the embedding model is appropriately controlled, we expect semantically similar entities to be close to each other, which will allow the model to generalize to unseen facts. However the notion of proximity may strongly depend on the type of a relation. For example, Benjamin Franklin was an engineer but also a politician. We would need different metrics to capture his proximity to other engineers and politicians of his time.

In this work, we propose a new attention model for content-based addressing. Our model scores each item $\boldsymbol{v}_{\texttt{item}}$ in the memory by the (logarithm of) multivariate Gaussian likelihood as follows:

$$\begin{aligned}\text{score}(\boldsymbol{v}_{\texttt{item}}) &= \log \phi(\boldsymbol{v}_{\texttt{item}} | \boldsymbol{\mu}_{\texttt{context}}, \boldsymbol{\Sigma}_{\texttt{context}}) \\ &= -\frac{1}{2}(\boldsymbol{v}_{\texttt{item}} - \boldsymbol{\mu}_{\texttt{context}})\boldsymbol{\Sigma}^{-1}_{\texttt{context}}(\boldsymbol{v}_{\texttt{item}} - \boldsymbol{\mu}_{\texttt{context}}) + \text{const.} \end{aligned} \tag{3.1}$$

where `context` denotes all the variables that the attention depends on. For example, "American engineers in the 18th century" or "American politicians in the 18th century" would be two contexts that include Benjamin Franklin but the two attentions would have very different shapes.

Compared to the (normalized) inner product used in previous work (Sukhbaatar et al., 2015; Graves et al., 2014) for content-based addressing, the Gaussian model has the additional control of the spread of the attention over items in the memory. As we show in Figure 3.1, we can view the conventional inner-product-based attention and the proposed Gaussian attention as addressing by an affine energy function and a quadratic energy function, respectively. By making the addressing mechanism more complex, we may represent many entities in a relatively low dimensional embedding space. Since knowledge bases are typically extremely sparse, it is more likely that we can afford to have a more complex attention model

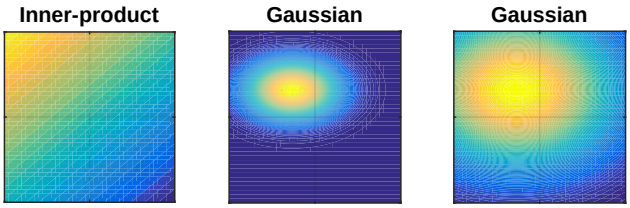


Figure 3.1: Comparison of the conventional content-based attention model using inner product and the proposed Gaussian attention model with the same mean but two different covariances.

than a large embedding dimension.

We apply the proposed Gaussian attention model to question answering based on knowledge bases. At the high-level, the goal of the task is to learn the mapping from a question about objects in the knowledge base in natural language to a probability distribution over the entities. We use the scoring function (3.1) for both embedding the entities as vectors, and extracting the conditions mentioned in the question and taking a conjunction of them to score each candidate answer to the question.

The ability to compactly represent a set of objects makes the Gaussian attention model well suited for representing the uncertainty in a multiple-answer question (e.g., "who are the children of Abraham Lincoln?"). Moreover, traversal over the knowledge graph (see Guu et al., 2015) can be naturally handled by a series of Gaussian convolutions, which generalizes the addition of vectors. In fact, we model each relation as a Gaussian with mean and variance parameters. Thus a traversal on a relation corresponds to a translation in the mean and addition of the variances.

The proposed question answering model is able to handle not only the case where the answer to a question is associated with an atomic fact, which is called simple Q&A (Bordes et al., 2015), but also questions that require composition of relations (path queries in Guu et al. (2015)) and conjunction of queries. An example flow of how our model deals with a question "Who plays forward for Borussia Dortmund?" is shown in Figure 3.2 in Section 3.4.

This chapter is structured as follows. In Section 3.3, we describe how the Gaussian scoring function (3.1) can be used to embed the entities in a knowledge base into a continuous vector space. We call our model TransGaussian because of its similarity to the TransE model proposed by Bordes et al. (2013). Then in Section 3.4, we describe our question answering model. In Section 3.6, we carry out experiments on WorldCup2014 dataset we collected. The dataset is relatively small but it allows us to evaluate not only simple questions but also path queries and conjunction of queries. The proposed TransGaussian embedding with the question answering model achieves significantly higher accuracy than the vanilla TransE embedding or TransE trained with compositional relations Guu et al. (2015) combined with the same question answering model.

3.2 Background and related work

3.2.1 Distributed Representation of Words

Distributed representations or low-dimensional vector embeddings of words (e.g. Mikolov et al. (2013b); Pennington et al. (2014)) have been widely used in the area of natural language processing. Different from treating every word in the vocabulary as a unique unit, embedding words in low-dimensional is capable of modelling similarity and relationships between words. The work by Mikolov et al. (2013c) showed that word embeddings can capture meaningful syntactic and semantic relationships between words. For example, the difference between the learned word vectors *"king - man"* is very close to *"queen - woman"* while *"apple - apples"* produces a vector near *"car - cars"*. Low-dimensional embeddings have been adapted in various applications in natural language processing and lead to outstanding performance. It is also commonly used together with neural network to address more sophisticated tasks such as sentiment analysis and machine translation.

As an alternative to point vector representations, Vilnis and McCallum (2014) advocates

for density-based distributed embeddings of words and represents a word as a Gaussian distribution. They discuss many advantages of the Gaussian embedding; for example, it is arguably a better way of handling asymmetric relations and entailment. Their work is similar to our Gaussian attention model. However their density-based embedding was presented in the word2vec(Mikolov et al., 2013a)-style word embedding setting and the Gaussian embedding was used to capture the diversity in the meaning of a word. Our Gaussian attention model extends their work to a more general setting in which any memory item can be addressed through a concept represented as a Gaussian distribution over the memory items. Under the same spirit as word embedding, vector representations are also widely used to represent other types of objects such as sentences, paragraphs and documents (Le and Mikolov, 2014).

3.2.2 Knowledge graph and embeddings

Large scale knowledge graphs (or knowledge bases) such as Freebase (Bollacker et al., 2008), WikiData (Vrandečić and Krötzsch, 2014) and WordNet (Miller, 1995) provide enormous structured information in the world and are extremely useful to tasks such as automated question answering, information retrievel, document understanding, etc. Yet, it is well-known that large scale knowledge bases are highly incomplete. By using the same idea as word embedding, finding low-dimensional representation for entities and relations serves as an efficient way to extract the patterns and discover unseen facts in the knowledge graphs. In this work, we work with knowledge graphs stored in the form of directed multi-graphs. Every vertex in the graph represents an *entity* e.g. "the United States", "New York City", "Chicago Bulls". Every directed edge in the graph is assigned a label which indicates the *relation* between the entities it connects e.g. an edge goes from "New York City" to "the United States" can be labeld "a city of", "Chicago Bulls" is a "basketball team based in" "the United State".

As one of the pioneering work of knowledge graph embedding, Bordes et al. (2013) proposed to use low-dimensional vectors to represent entities and relations in a knowledge graph. Given a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$, their algorithm learns d-dimensional vectors $\{v_e : e \in \mathcal{E}\} \subset \mathbb{R}^d$ and $\{\delta_r : r \in \mathcal{R}\} \subset \mathbb{R}^d$ such that $\|v_h + \delta_r - v_t\|$ (with either 1-norm or Euclidean norm) is small if and only if there is an directed edge of relation r goes from entity h to entity t. Intuitively, walking along an edge in the knowledge graph is modelled as a translation along a vector in the embedding space. Hence their model is named *TransE*.

There have been several variations of embedding models for knowledge graph (Nickel et al., 2011; Wang et al., 2014; Nickel et al., 2015; Socher et al., 2013; Trouillon et al., 2016; Joulin et al., 2017). In RESCAL (Nickel et al., 2011), a relation is modelled by a matrix transformation hence walking along an edge becomes transforming an entity vector by the corresponding relation matrix. To improve the modelling of some relation properties such as reflexivity, one-to-many, many-to-one and many-to-many, the TransH embedding Wang et al. (2014) models each relation as a translation on a hyperplane. HolE (Nickel et al., 2015) and ComplEx (Trouillon et al., 2016) [1] model entities and relations as vectors in complex vector space $\mathbb{C}^d$ and achieved the state-of-the-art performance in predicting unseen facts on knowledge graph (also known as link-prediction). The neural tensor model (NTN) by Socher et al. (2013) scores the likelihood of a fact "e_1 and e_2 has relation r" being true by using an expressive bilinear tensor model. More recently, Joulin et al. (2017) build up a much lighter weighted model aiming at only capturing the co-occurences of entities and relations in the knowledge graph. Their model is very similar to TransE but with different scoring function and loss function. Surprisingly, their model obtained performance competitive to state-of-the-art while it took drastically less time to train. Abstractly, these models all share the same form: entities in the knowledge graph are presented as vectors (either in $\mathbb{R}^d$ or in $\mathbb{C}^d$) while a score function score(s, r, o) is employed to indicate how much the fact "s and o

1. HolE and ComplEx are proved to be mathematically equivalent by Hayashi and Shimbo (2017). Trouillon and Nickel (2017) conducted a detailed comparison of these two models and their performance in practice.

Table 3.1: Lists of models for knowledge graph embedding. In this table, $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$ is the hyperbolic tangent function; $\text{logist}(x) = e^x/(1+e^x)$ is the logistic function; $(x \star y)_k = \sum_{i=0}^{d-1} x_i y_{(k+i) \bmod d}$ is the circular correlation; and $\langle a, b, c \rangle = \sum_k a_k b_k c_k$ is the multilinear product.

Model	Entity	Relation	score(s, r, o)
RESCAL (Nickel et al., 2011)	$v_e \in \mathbb{R}^d$	$M_r \in \mathbb{R}^{d \times d}$	$v_o^\top M_r v_s$
TransE (Bordes et al., 2013)	$v_e \in \mathbb{R}^d$	$\delta_r \in \mathbb{R}^d$	$- \|v_s + \delta_r - v_o\|$
NTN (Socher et al., 2013)	$v_e \in \mathbb{R}^d$	$u_r \in \mathbb{R}^k$, $W_r \in \mathbb{R}^{d \times d \times k}$, $M_{r,1}, M_{r,2} \in \mathbb{R}^{k \times d}$, $b_r \in \mathbb{R}^k$	$u_r^\top \tanh\big(v_s^\top W_r v_o + M_{r,1} v_s + M_{r,2} v_o + b_r\big)$
TransH (Wang et al., 2014)	$v_e \in \mathbb{R}^d$	$w_r, \delta_r \in \mathbb{R}^d$	$-\|(I - w_r w_r^\top) v_s + \delta_r - (I - w_r w_r^\top) v_o\|_2^2$
HolE (Nickel et al., 2015)	$v_e \in \mathbb{R}^d$	$u_r \in \mathbb{R}^d$	$\text{logist}(u_r^\top (v_s \star v_o))$
ComplEx (Trouillon et al., 2016)	$v_e \in \mathbb{C}^d$	$w_r \in \mathbb{C}^d$	$\text{Re}\big(\langle w_r, v_s, \overline{v_o} \rangle\big)$
fastText (Joulin et al., 2017)	$v_e \in \mathbb{R}^d$	$v_r \in \mathbb{R}^d$	$\frac{1}{2}\langle v_s + v_r, v_o \rangle$

have relation r" is likely to be true. As a summary, Table 3.1 illustrates the parameterization and the score functions of the embedding models under this general form.

3.2.3 Compositionality of knowledge graph embeddings

A compositional relation is a relation that is composed as a series of relations in $\mathcal{R}$, for example, `grand_father_of` can be composed as first applying the `parent_of` relation and then the `father_of` relation, which can be seen as a traversal over a path on the knowledge graph. We will write the series of k relations on a path as $r_1/r_2/\cdots/r_k$. Guu et al. (2015) raised the concept of *composable* embedding model: an embedding model is called composable if its scoring function score(s, r, o) can be expressed in the form:

$$\text{score}(s, r, o) = \mathbb{M}(\mathbb{T}_r(v_s), v_o)$$

for some membership operator $\mathbb{M} : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ and traversal operator $\mathbb{T} : \mathbb{R}^d \mapsto \mathbb{R}^d$. For example, TransE model has $\mathbb{M}(u, v) = \|u - v\|$, $\mathbb{T}_r(v) = v + \delta_r$ where $\delta_r \in \mathbb{R}^d$ is the vector parameterizing the relation r. For RESCAL, the membership function is $\mathbb{M}(u, v) = v^\top u$ and the traversal operator is $\mathbb{T}_r(v) = M_r v$. Thus, the traversal operator for a composed relation

$r_1/r_2/\cdots/r_k$ becomes

$$\mathbb{T}_{r_1/r_2/\cdots/r_k} \coloneqq \mathbb{T}_{r_k} \circ \cdots \circ \mathbb{T}_{r_2} \circ \mathbb{T}_{r_1}$$

and the score about a fact involving a composed relation "s is related to o through the composed relation $r_1/r_2/\cdots/r_k$" can be computed by

$$\text{score}(s, r_1/r_2/\cdots/r_k, o) = \mathbb{M}(\mathbb{T}_{r_1/r_2/\cdots/r_k}(v_s), v_o).$$

Guu et al. (2015) has shown that training TransE with *compositional relations* by sampling paths from the knowledge graph can make it competitive to more complex models, although TransE is much simpler compared to, for example, neural tensor networks (NTN, Socher et al. (2013)) and TransH Wang et al. (2014). The authors also pointed out that some embedding models such as NTN Socher et al. (2013) are not naturally composable. We will should that our proposed model is composable and hence can be used for answering questions which require walking down a path in the knowledge graph.

3.2.4 Question answering on knowledge graph

Large scale knowledge graph provides highly structured source of knowledge and can be very useful for open-domain question answering. However, finding the appropriate way to incorporate such knowledge into question answering system remains a challenging problem. Bordes et al. (2014, 2015) proposed a question-answering model that embeds both questions and their answers to a common continuous vector space. Their method in Bordes et al. (2015) can combine multiple knowledge bases and even generalize to a knowledge base that was not used during training. However their method is limited to the simple question answering setting in which the answer of each question associated with a triplet in the knowledge base. In contrast, our method can handle both composition of relations and conjunction of conditions, which are both naturally enabled by the proposed Gaussian attention model.

Neelakantan et al. (2015a) proposed a method that combines relations to deal with compositional relations for knowledge base completion. Their key technical contribution is to use recurrent neural networks (RNNs) to *encode* a chain of relations. When we restrict ourselves to path queries, question answering can be seen as a sequence transduction task (Graves, 2012; Sutskever et al., 2014) in which the input is text and the output is a series of relations. Another interesting connection to our work is that they take the maximum of the inner-product scores (see also Weston et al., 2013; Neelakantan et al., 2015b), which are computed along multiple paths connecting a pair of entities. Representing a set as a collection of vectors and taking the maximum over the inner-product scores is a natural way to represent a set of memory items. The Gaussian attention model we propose in this work, however, has the advantage of differentiability and composability.

3.3 The TransGaussian model

In this section, we describe the proposed TransGaussian model based on the Gaussian attention model (3.1). While it is possible to train a network that computes the embedding in a single pass (Bordes et al., 2015) or over multiple passes (Li et al., 2015), it is more efficient to offload the embedding as a separate step for question answering based on a large static knowledge base.

Let $\mathcal{E}$ be the set of entities and $\mathcal{R}$ be the set of relations. A knowledge base is a collection of triplets (s, r, o), where we call $s \in \mathcal{E}$, $r \in \mathcal{R}$, and $o \in \mathcal{E}$, the subject, the relation, and the object of the triplet, respectively. Each triplet encodes a *fact*. For example, (`Albert_Einstein`, `has_profession`, `theoretical_physicist`). All the triplets given in a knowledge base are assumed to be true. However generally speaking a triplet may be true or false. Thus knowledge base embedding aims at training a model that predict if a triplet is true or not given some parameterization of the entities and relations (Bordes et al., 2011, 2013; Nickel et al., 2011; Socher et al., 2013; Wang et al., 2014).

In this work, we associate a vector $\boldsymbol{v}_s \in \mathbb{R}^d$ with each entity $s \in \mathcal{E}$, and we associate each relation $r \in \mathcal{R}$ with two parameters, $\boldsymbol{\delta}_r \in \mathbb{R}^d$ and a positive definite symmetric matrix $\boldsymbol{\Sigma}_r \in \mathbb{R}^{d \times d}_{++}$.

Given subject s and relation r, we can compute the score of an object o to be in triplet (s, r, o) using the Gaussian attention model as (3.1) with

$$\text{score}(s, r, o) = \log \phi(\boldsymbol{v}_o | \boldsymbol{\mu}_{\texttt{context}}, \boldsymbol{\Sigma}_{\texttt{context}}), \tag{3.2}$$

where $\boldsymbol{\mu}_{\texttt{context}} = \boldsymbol{v}_s + \boldsymbol{\delta}_r$, $\boldsymbol{\Sigma}_{\texttt{context}} = \boldsymbol{\Sigma}_r$. Note that if $\boldsymbol{\Sigma}_r$ is fixed to the identity matrix, we are modeling the relation of subject $\boldsymbol{v}_s$ and object $\boldsymbol{v}_o$ as a translation $\boldsymbol{\delta}_r$, which is equivalent to the TransE model (Bordes et al., 2013). We allow the covariance $\boldsymbol{\Sigma}_r$ to depend on the relation to handle one-to-many relations (e.g., `profession_has_person` relation) and capture the shape of the distribution of the set of objects that can be in the triplet. We call our model **TransGaussian** because of its similarity to TransE (Bordes et al., 2013).

Parameterization For computational efficiency, we will restrict the covariance matrix $\boldsymbol{\Sigma}_r$ to be diagonal in this work. Furthermore, in order to ensure that $\boldsymbol{\Sigma}_r$ is strictly positive definite, we employ the exponential linear unit (ELU, Clevert et al., 2015) and parameterize $\boldsymbol{\Sigma}_r$ as follows:

$$\boldsymbol{\Sigma}_r = \begin{pmatrix} \text{ELU}(m_{r,1})+1+\epsilon & & \\ & \ddots & \\ & & \text{ELU}(m_{r,d})+1+\epsilon \end{pmatrix}$$

where $m_{r,j}$ $(j = 1, \ldots, d)$ are the unconstrained parameters that are optimized during training and ϵ is a small positive value that ensure the positivity of the variance during numerical

computation. The ELU is defined as

$$\mathrm{ELU}(x) = \begin{cases} x, & x \geq 0, \\ \exp(x) - 1, & x < 0. \end{cases}$$

Loss function Suppose we have a set of triplets $\mathcal{T} = \{(s_i, r_i, o_i)\}_{i=1}^{N}$ from the knowledge base. Let $\mathcal{N}(s, r, o)$ denote the set of incorrect triplets by replacing either the subject or the object in the triplet (s, r, o) with every entity that leads to a false triplet. Our objective function utilizes a loss function to measure the difference between the scores of true triplets and those of false triplets. We will experiment with the following two loss functions:

1. Margin loss:

$$\ell(s, s') \coloneqq \max\{\mu - s + s', 0\}$$

   where μ is the margin parameter.

2. Ratio loss:

$$\ell(s, s') \coloneqq -\log \frac{e^s}{e^s + e^{s'}}$$

The objective function can be written as follows:

$$\min_{\substack{\{\boldsymbol{v}_e : e \in \mathcal{E}\}, \\ \{\boldsymbol{\delta}_r, \boldsymbol{M}_r, : r \in \bar{\mathcal{R}}\}}} \frac{1}{N} \sum_{(s,r,o) \in \mathcal{T}} \mathbb{E}_{(s',r,o') \sim \mathcal{N}(s,r,o)} \ell(\mathrm{score}(s, r, o), \mathrm{score}(s', r, t')) \tag{3.3}$$

$$+ \lambda \Big[\sum_{e \in \mathcal{E}} \|\boldsymbol{v}_e\|_2^2 + \sum_{r \in \bar{\mathcal{R}}} \big(\|\boldsymbol{\delta}_r\|_2^2 + \|\boldsymbol{M}_r\|_F^2 \big) \Big], \tag{3.4}$$

where, $N = |\mathcal{T}|$, $\boldsymbol{M}_r$ denotes the diagonal matrix with $m_{r,j}$, $j = 1, \ldots, d$ on the diagonal. Here, we treat an inverse relation as a separate relation and denote by $\bar{\mathcal{R}} = \mathcal{R} \cup \mathcal{R}^{-1}$ the set of all the relations including both relations in $\mathcal{R}$ and their inverse relations; a relation $\tilde{r}$ is the inverse relation of r if $(s, \tilde{r}, o)$ implies (o, r, s) and vice versa. Moreover, $\mathbb{E}_{t' \sim \mathcal{N}(s,r,o)}$

denotes the expectation with respect to the uniform distribution over the set of incorrect triplets, which we approximate with 10 random samples in the experiments. Finally, the last terms are ℓ_2 regularization terms for the embedding parameters.

3.3.1 Compositional relations

TransGaussian model can naturally handle and propagate the uncertainty over such a chain of relations by convolving the Gaussian distributions along the path. That is, the score of an entity o to be in the τ-step relation $r_1/r_2/\cdots/r_\tau$ with subject s, which we denote by the triplet $(s, r_1/r_2/\cdots/r_\tau, o)$, is given as

$$\text{score}(s, r_1/r_2/\cdots/r_\tau, o) = \log \phi(\boldsymbol{v}_o | \boldsymbol{\mu}_{\texttt{context}}, \boldsymbol{\Sigma}_{\texttt{context}}), \tag{3.5}$$

with $\boldsymbol{\mu}_{\texttt{context}} = \boldsymbol{v}_s + \sum_{t=1}^{\tau} \boldsymbol{\delta}_{r_t}$, $\boldsymbol{\Sigma}_{\texttt{context}} = \sum_{t=1}^{\tau} \boldsymbol{\Sigma}_{r_t}$, where the covariance associated with each relation is parameterized in the same way as in the previous subsection.

Training with compositional relations Let $\mathcal{P} = \left\{(s_i, r_{i_1}/r_{i_2}/\cdots/r_{i_{l_i}}, o_i)\right\}_{i=1}^{N'}$ be a set of randomly sampled paths from the knowledge graph. Here relation r_{i_k} in a path can be a relation in $\mathcal{R}$ or an inverse relation in $\mathcal{R}^{-1}$. With the scoring function (3.5), the generalized training objective for compositional relations can be written identically to (3.3) except for replacing $\mathcal{T}$ with $\mathcal{T} \cup \mathcal{P}$ and replacing N with $N' = |\mathcal{T} \cup \mathcal{P}|$.

3.4 Question answering on embedded knowledge graph by using recurrent neural network

Given a set of question-answer pairs, in which the question is phrased in natural language and the answer is an entity in the knowledge base, our goal is to train a model that learns the mapping from the question to the correct entity. Our question answering model consists

of three steps, entity recognition, relation composition, and conjunction. We first identify a list of entities mentioned in the question (which is assumed to be provided by an oracle in this work). If the question is "Who plays Forward for Borussia Dortmund?" then the list would be [`Forward`, `Borussia_Dortmund`]. The next step is to predict the path of relations on the knowledge graph starting from each entity in the list extracted in the first step. In the above example, this will be (smooth versions of) `/Forward/position_played_by/` and `/Borussia_Dortmund/has_player/` predicted as series of Gaussian convolutions. In general, we can have multiple relations appearing in each path. Finally, we take a product of all the Gaussian attentions and renormalize it, which is equivalent to Bayes' rule with independent observations (paths) and a noninformative prior.

3.4.1 Entity recognition

We assume that there is an oracle that provides a list containing all the entities mentioned in the question, because (1) a domain specific entity recognizer can be developed efficiently (Williams et al., 2015) and (2) generally entity recognition is a challenging task and it is beyond the scope of this work to show whether there is any benefit in training our question answering model jointly with a entity recognizer. We assume that the number of extracted entities can be different for each question.

3.4.2 Relation composition

We train a long short-term memory (LSTM, Hochreiter and Schmidhuber, 1997) network that emits an output $\boldsymbol{h}_t$ for each token in the input sequence. Then we compute the attention over the hidden states for each recognized entity e as

$$p_{t,e} = \text{softmax}\left(f(\boldsymbol{v}_e, \boldsymbol{h}_t)\right) \quad (t = 1, \ldots, T),$$

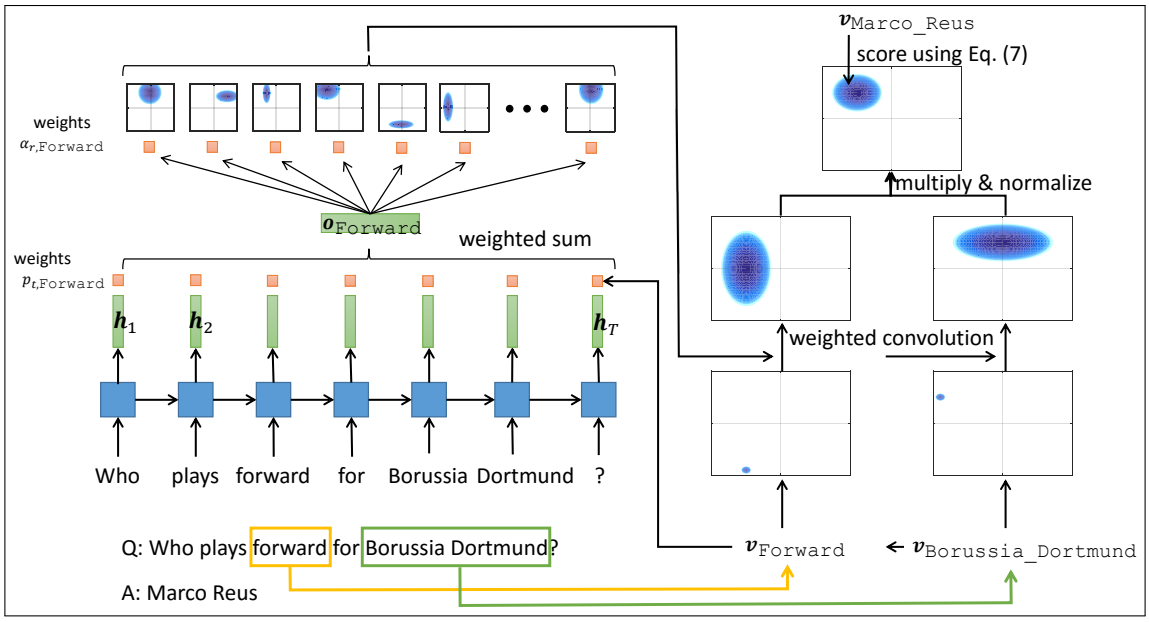


Figure 3.2: A schematic illustration of question answering with Gaussian attention. The input to the system is a question in natural language. Two entities `Forward` and `Borussia_Dortmund` are identified in the question and associated with point mass distributions centered at the corresponding entity vectors. An LSTM encodes the input into a sequence of output vectors of the same length. Then we take average of the output vectors weighted by attention $p_{t,e}$ for each recognized entity e to predict the weight $\alpha_{r,e}$ for relation r associated with entity e. We form a Gaussian attention over the entities for each entity e by convolving the corresponding point mass with the (pre-trained) Gaussian embeddings of the relations weighted by $\alpha_{r,e}$ according to Eq. (3.7). The final prediction is produced by taking the product and normalizing the Gaussian attentions.

where $\boldsymbol{v}_e$ is the vector associated with the entity e. We use a two-layer perceptron for f in our experiments, which can be written as follows:

$$f(\boldsymbol{v}_e, \boldsymbol{h}_t) = \boldsymbol{u}_f^\top \mathrm{ReLU}\left(\boldsymbol{W}_{f,v}\boldsymbol{v}_e + \boldsymbol{W}_{f,h}\boldsymbol{h}_t + \boldsymbol{b}_1\right) + b_2,$$

where $\boldsymbol{W}_{f,v} \in \mathbb{R}^{L\times d}$, $\boldsymbol{W}_{f,h} \in \mathbb{R}^{L\times H}$, $\boldsymbol{b}_1 \in \mathbb{R}^L$, $\boldsymbol{u}_f \in \mathbb{R}^L$, $b_2 \in \mathbb{R}$ are parameters. Here $\mathrm{ReLU}(x) = \max(0, x)$ is the rectified linear unit. Finally, softmax denotes softmax over the T tokens.

Next, we use the weights $p_{t,e}$ to compute the weighted sum over the hidden states $\boldsymbol{h}_t$ as

$$\boldsymbol{o}_e = \sum_{t=1}^{T} p_{t,e} \boldsymbol{h}_t. \tag{3.6}$$

Then we compute the weights $\alpha_{r,e}$ over all the relations as $\alpha_{r,e} = \mathrm{ReLU}\left(\boldsymbol{w}_r^\top \boldsymbol{o}_e\right) \quad (\forall r \in \mathcal{R} \cup \mathcal{R}^{-1})$. Here the rectified linear unit is used to ensure the positivity of the weights. Note however that the weights should not be normalized, because we may want to use the same relation more than once in the same path. Making the weights positive also has the effect of making the attention sparse and interpretable because there is no cancellation.

For each extracted entity e, we view the extracted entity and the answer of the question to be the subject and the object in some triplet (e, p, o), respectively, where the path p is inferred from the question as the weights $\alpha_{r,e}$ as we described above. Accordingly, the score for each candidate answer o can be expressed using (3.1) as:

$$\mathrm{score}_e(\boldsymbol{v}_o) = \log \phi(\boldsymbol{v}_o | \boldsymbol{\mu}_{e,\alpha,\mathrm{KB}}, \boldsymbol{\Sigma}_{e,\alpha,\mathrm{KB}}) \tag{3.7}$$

with $\boldsymbol{\mu}_{e,\alpha,\mathrm{KB}} = \boldsymbol{v}_e + \sum_{r \in \bar{\mathcal{R}}} \alpha_{r,e} \boldsymbol{\delta}_r$, $\boldsymbol{\Sigma}_{e,\alpha,\mathrm{KB}} = \sum_{r \in \bar{\mathcal{R}}} \alpha_{r,e}^2 \boldsymbol{\Sigma}_r$, where $\boldsymbol{v}_e$ is the vector associated with entity e and $\bar{\mathcal{R}} = \mathcal{R} \cup \mathcal{R}^{-1}$ denotes the set of relations including the inverse relations.

3.4.3 Conjunction

Let $\mathcal{E}(q)$ be the set of entities recognized in the question q. The final step of our model is to take the conjunction of the Gaussian attentions derived in the previous step. This step is

simply carried out by multiplying the Gaussian attentions as follows:

$$\begin{aligned}\text{score}(\boldsymbol{v}_o|\mathcal{E}(q),\Theta) &= \log \prod_{e\in\mathcal{E}(q)} \phi(\boldsymbol{v}_o|\boldsymbol{\mu}_{e,\alpha,\text{KB}},\boldsymbol{\Sigma}_{e,\alpha,\text{KB}}) \\ &= -\frac{1}{2}\sum_{e\in\mathcal{E}(q)} \left(\boldsymbol{v}_o-\boldsymbol{\mu}_{e,\alpha,\text{KB}}\right)^\top \boldsymbol{\Sigma}_{e,\alpha,\text{KB}}^{-1}(\boldsymbol{v}_o-\boldsymbol{\mu}_{e,\alpha,\text{KB}}) + \text{const.}, \end{aligned} \tag{3.8}$$

which is again a (logarithm of) Gaussian scoring function, where $\boldsymbol{\mu}_{e,\alpha,\text{KB}}$ and $\boldsymbol{\Sigma}_{e,\alpha,\text{KB}}$ are the mean and the covariance of the Gaussian attention given in (3.7). Here Θ denotes all the parameters of the question-answering model.

3.4.4 Training the question answering model

Suppose we have a knowledge base $(\mathcal{E},\mathcal{R},\mathcal{T})$ and a trained TransGaussian representation $\left(\left\{v_e\right\}_{e\in\mathcal{E}},\left\{(\boldsymbol{\delta}_r,\boldsymbol{\Sigma}_r)\right\}_{r\in\bar{\mathcal{R}}}\right)$, where $\bar{\mathcal{R}}$ is the set of all relations including the inverse relations. During training time, we assume the training set is a supervised question-answer pairs $\{(q_i,\mathcal{E}(q_i),a_i) : i=1,2,\ldots,m\}$. Here, q_i is a question formulated in natural language, $\mathcal{E}(q_i)\subset\mathcal{E}$ is a set of knowledge base entities that appears in the question, and $a_i\in\mathcal{E}$ is the answer to the question. For example, on a knowledge base of soccer players, a valid training sample could be

$$\begin{aligned}\{\quad q&\text{: ``Who plays forward for Borussia Dortmund?''},\\ \mathcal{E}(q)&\text{: [}\texttt{Forward}\text{, }\texttt{Borussia\_Dortmund}\text{]},\\ a&\text{: [}\texttt{Marco\_Reus}\text{]}\}.\end{aligned}$$

Note that the answer to a question is not necessarily unique and we allow a_i to be any of the true answers in the knowledge base. During test time, our model is shown $(q_i,\mathcal{E}(q_i))$ and the task is to find a_i. We denote the set of answers to q_i by $A(q_i)$.

To train our question-answering model, we minimize the objective function

$$\frac{1}{m}\sum_{i=1}^{m}\Big(\mathop{\mathbb{E}}_{t'\sim\mathcal{N}(q_i)}\big[[\mu-\text{score}(\boldsymbol{v}_{a_i}|\mathcal{E}(q_i),\Theta)+\text{score}(\boldsymbol{v}_{t'}|\mathcal{E}(q_i),\Theta)]_+\big]+\nu\sum_{e\in\mathcal{E}(q_i)}\sum_{r\in\bar{\mathcal{R}}}\big|\alpha_{r,e}\big|\Big)+\lambda\|\Theta\|_2^2$$

where $\mathbb{E}_{t'\sim\mathcal{N}(q_i)}$ is expectation with respect to a uniform distribution over of all incorrect answers to q_i, which we approximate with 10 random samples. We assume that the number of relations implied in a question is small compared to the total number of relations in the knowledge base. Hence the coefficients $\alpha_{r,e}$ computed for each question q_i are regularized by their ℓ_1 norms.

3.5 Experiments: knowledge graph completion

The task of knowledge graph completion tests knowledge graph models on their ability of generalizing to unseen facts. Here, we apply our TransGaussian model to knowledge completion tasks and show that it has competitive performance. We first introduce the experimental setup. Evaluation of TransGaussian model follows.

3.5.1 Experimental setup

Tasks The most common tasks of knowledge graph completion include two kinds: *triplet classification* and *link prediction*. In both tasks, the training algorithm takes a set of training triplets $\mathcal{T}=\{(s,r,o):s,o\in\mathcal{E},r\in\mathcal{R}\}$ and learn the representation of entities and relations. All triplets in the training set are considered true facts (or positive examples). The two tasks differ in their test phase:

- In triplet classification, the model needs to predict if an unseen triplet is true or false during test time;

- In link prediction, the model needs to predict the objects for a pair of subject and relation (s, r). Such a prediction can be considered as a ranking problem: the model usually ranks all entities $e \in \mathcal{E}$ based on the score of triplets (s, r, e). Likewise, the model may be asked to predict the subject for a pair of object and relation during test time.

Evaluation metrics In the case of triplet classification, the test set contains both true triplets and false triplets. A model is evaluated by its classification accuracy on the test set. For link prediction, the test set consists of only true triplets. And the model predicts the subject (or object) of every triplet by only looking at its object (or subject) and relation. Notice that a pair of subject and relation may have more than one valid object in the knowledge graph. For example, a soccer team has multiple players and a singer may have released multiple albumns. Hence, when the model is making a prediction for the object of a triplet (s, r, o) in the test set, we evaluate the quality of the ranking after removing all other entities that can serve as the object. Likewise, we do the same for subject prediction. A rank after other positive entities are removed is sometimes called a *filtered rank* in the literature. Two measures are applied to this filtered rank:

- *Mean rank*, the average rank of the target entity across all test triplets;
- *Mean reciprocal rank (MRR)*, the average of the reciprocal of the ranks;
- *Hits@10*, the percentage of time when the target entity appears among the top 10 in the rank.

Datasets For triplet classification, we use the datasets *WN11* and *FB13* from Socher et al. (2013). *WN11* is a subset of WordNet. There are 11 different relations and 38,696 unique entities each of which is a synset of words from WordNet. *FB13* is a subset of triplets from FreeBase under the *People* domain and includes 13 relations and 75,043 entities. In the test

Table 3.2: Statistics of datasets for knowledge graph completion.

Task	Dataset	# entities	# relations	Number of triplets (Train / Val / Test)
Triplet classification	WN11	38,696	11	112,581 / 2,609 / 10,544
	FB13	75,043	13	316,232 / 5,908 / 23,733
Link prediction	WN18	40,943	18	141,442 / 5,000 / 5,000
	FB15K	14,951	1,345	483,142 / 50,000 / 59,071

sets of WN11 and FB13, there are euqal number of positive and negative examples. For link prediction, we conduct experiments on *WN18*, a subset of WordNet with 18 relations and 40,943 entities, and *FB15k* a subset FreeBase with 1,345 relations and 14,951 entities. Both datasets are taken from Bordes et al. (2013). Table 3.2 gives the statistics of these datasets.

Training configurations During every iteration, we replace the object of every positive triplet with 10 entities randomly selected from the knowledge graph to generate 10 negative examples. We generate another 10 negative examples by corrupting the subject for each triplet as well. Adam (Kingma and Ba, 2014) was employed as the optimizer for training the TransGaussian model. All hyperparameters were tuned on the validation set. For the task of triplet classification, we also experimented with using word embedding as done by Socher et al. (2013). The name of an entity in a knowledge graph may contain one or more words. Socher et al. (2013) proposed to represent the entity vector by averaging its word vectors. For example, $v_{\texttt{united\_states}} = 0.5(v_{\texttt{united}} + v_{\texttt{states}})$. Under this setting, the training algorithm aims to learn the word vectors while the representation of the relations remains the same.

3.5.2 Experimental results

We report our experimental results on triplet classification of a 100 dimensional TransGaussian embedding trained with the margin loss in Table 3.3. Columns labeled with "EV" and "WV" show results from using entity vectors and word vectors for entity representation re-

Table 3.3: Triplet classification: accuracies (%).

Model	WN11		FB13	
	EV	WV	EV	WV
NTN (Socher et al., 2013)	70.4	86.2	87.2	90.0
TransE (unif.) (Wang et al., 2014)	75.85	-	70.9	-
TransE (bern.) (Wang et al., 2014)	75.87	-	81.5	-
TransH (unif.) (Wang et al., 2014)	77.68	-	76.5	-
TransH (bern.) (Wang et al., 2014)	78.80	-	83.3	-
TransGaussian	75.40	76.60	86.95	89.20

spectively. TransGaussian was able to achieve results comparable to the baseline models. We see that using word vector representation improves the classification accuracy by roughly 1% to 2%.

Results on link prediction task are presented in Table 3.4. We experimented with both margin loss and ratio loss for this task. The results show that TransGaussian performs better than TransE and TransH in terms of Hits@10. However, TransGaussian has a larger mean rank. Comparing results for two losses functions, we found that ratio loss performs better than margin loss overall. The benefit of using ratio loss is more significant in mean reciprocal rank on WN18. Both ComplEx and HolE achieve the best mean reciprocal rank among all models and stay the state of art on these two benchmarks. TransGaussian is able to achieve comparable performance in Hits@10 but is outperformed by ComplEx by a large margin. This implies that ComplEx produces a better ranking for the top entities than TransGaussian. We show the experimental results on question answering in the next section.

Table 3.4: Link prediction on knowledge base. For TransGaussian, two loss functions margin loss (margin) and ratio loss (ratio) were used.

Model	WN18			FB15k		
	Mean Rank	MRR	Hits@10 (%)	Mean rank	MRR	Hits@10 (%)
TransE (Bordes et al., 2013)	251	-	89.2	125	-	47.1
TransH (unif.) (Wang et al., 2014)	303	-	86.7	84	-	58.5
TransH (bern.) (Wang et al., 2014)	388	-	82.3	87	-	64.4
HolE (Nickel et al., 2015)	-	0.938	94.9	-	0.524	73.9
ComplEx (Trouillon and Nickel, 2017)	-	0.941	94.7	-	0.692	84.0
TransGaussian (margin, 50 dim)	484	0.537	88.6	79	0.425	66.9
TransGaussian (margin, 100 dim)	577	0.537	89.6	88	0.495	74.7
TransGaussian (margin, 150 dim)	567	0.550	90.1	115	0.548	79.3
TransGaussian (ratio, 50 dim)	565	0.663	94.0	77	0.437	69.9
TransGaussian (ratio, 100 dim)	646	0.668	92.8	70	0.534	78.3
TransGaussian (ratio, 150 dim)	642	0.654	93.1	68	0.551	79.5

3.6 Experiments: question answering with TransGaussian

As a demonstration of the proposed framework of question answering on knowledge graph, we perform question and answering on a dataset of soccer players. In this work, we consider two types of questions. A *path query* is a question that contains only one named entity from the knowledge base and its answer can be found from the knowledge graph by walking down a path consisting of a few relations. A *conjunctive query* is a question that contains more than one entities and the answer is given as the conjunction of all path queries starting from each entity.

3.6.1 WorldCup2014 dataset

We build a knowledge base of football players that participated in FIFA World Cup 2014 [2]. The original dataset consists of players' information such as nationality, positions on the

2. The original dataset can be found at https://datahub.io/dataset/fifa-world-cup-2014-all-players.

Table 3.5: Atomic relations in WorldCup2014 dataset. Here, `wears_number` indicates players' jersey numbers in the national teams. PLAYER, CLUB, NUMBER, etc, denote the type of entities that can appear as the left or right argument for each relation. Some relations share the same type as the right argument, e.g., `plays_for_country` and `is_in_country`.

Relation	Types of subjects and objects
`plays_in_club`	PLAYER $\rightarrow$ CLUB
`plays_position`	PLAYER $\rightarrow$ POSITION
`is_aged`	PLAYER $\rightarrow$ NUMBER
`wears_number`	PLAYER $\rightarrow$ NUMBER
`plays_for_country`	PLAYER $\rightarrow$ COUNTRY
`is_in_country`	CLUB $\rightarrow$ COUNTRY

field and ages etc. We picked a few attributes and constructed 1127 entities and 6 atomic relations. The entities include 736 players, 297 professional soccer clubs, 51 countries, 39 numbers and 4 positions. And the six atomic relations are listed in Table 3.5.

Given the entities and relations, we transformed the dataset into a set of 3977 triplets. A list of sample triplets can be found in Table 3.6. Based on these triplets, we created two sets of question answering tasks which we call *path query* and *conjunctive query* respectively. The answer of every question is always an entity in the knowledge base and a question can involve one or two triplets. The questions are generated as follows.

Path queries. Among the paths on the knowledge graph, there are some natural composition of relations, e.g., `plays_in_country` (PLAYER $\rightarrow$ COUNTRY) can be decomposed as the composition of `plays_in_club` (PLAYER$\rightarrow$ CLUB) and `is_in_country` (CLUB $\rightarrow$ COUNTRY). In addition to the atomic relations, we manually picked a few meaningful compositions of relations and formed *query templates*, which takes the form "find $e \in \mathcal{E}$, such that (s, p, e) is true", where s is the subject and p can be an atomic relation or a path of relations. To formulate a set of path-based question-answer pairs, we manually created one or more *question templates* for every query template (see Table 3.8) Then, for a particular instantiation of a query template with subject and object entities, we randomly select a

question template to generate a question given the subject; the object entity becomes the answer of the question. See Table 3.9 for the list of composed relations, sample questions, and answers. Note that all atomic relations in this dataset are many-to-one while these composed relations can be one-to-many or many-to-many as well.

Conjunctive queries. To generate question-and-answer pairs of conjunctive queries, we first picked three pairs of relations and used them to create query templates of the form "Find $e \in \mathcal{E}$, such that both (s_1, r_1, e) and (s_2, r_2, e) are true." (see Table 3.8). For a pair of relations r_1 and r_2, we enumerated all pairs of entities s_1, s_2 that can be their subjects and formulated the corresponding query in natural language using question templates as in the same way as path queries. See Table 3.10 for a list of sample questions and answers.

As a result, we created 8003 question-and-answer pairs of path queries and 2208 pairs of conjunctive queries which are partitioned into train / validation / test subsets. We refer to Table 3.7 for more statistics about the dataset. Templates for generating the questions are list in Table 3.8.

Table 3.6: Sample atomic triplets.

Subject	Relation	Object
david_villa	`plays_for_country`	spain
lionel_messi	`plays_in_club`	fc_barcelona
antoine_griezmann	`plays_position`	forward
cristiano_ronaldo	`wears_number`	7
fulham_fc	`is_in_country`	england
lukas_podolski	`is_aged`	29

3.6.2 Experimental setup

To perform question and answering under our proposed framework, we first train the TransGaussian model on WorldCup2014 dataset. In addition to the atomic triplets, we randomly

Table 3.7: Statistics of the WorldCup2014 dataset.

Item	Count
Entities	1127
Atomic relations	6
Atomic triplets	3977
Relations (atomic and compositional) in path queries	12
Question and answer pairs in path queries (train / validation / test)	5313 / 760 / 1686
Types of questions in conjunctive queries	3
Question and answer pairs in conjunctive queries (train / validation / test)	1564 / 224 / 420
Unique words	1800

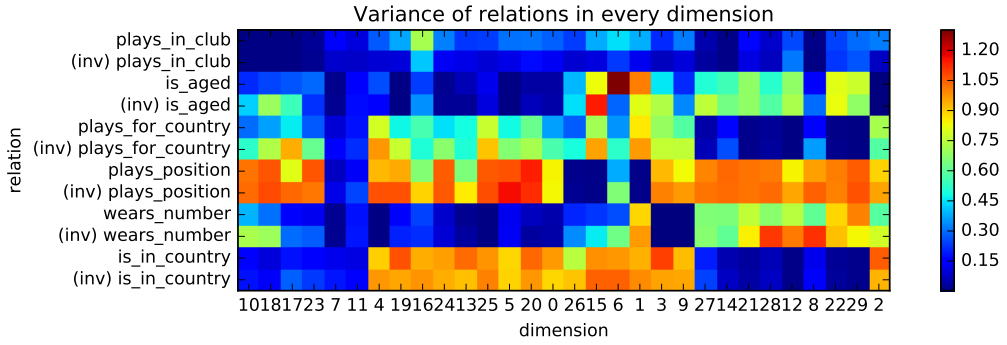


Figure 3.3: Variance of each trained relation. Each row shows the diagonal values in the variance matrix associated with a relation. Columns are permuted to reveal the block structure.

sampled 50000 paths with length 1 or 2 from the knowledge graph and trained a TransGaussian model compositionally as described in Set 3.3.1. An inverse relation is treated as a separate relation. Following the naming convention from Guu et al. (2015), we denote this trained embedding by *TransGaussian (COMP)*. We found that the learned embedding possess some interesting properties. Some dimensions of the embedding space dedicate to represent a particular relation. Players are clustered by their attributes when entities' embeddings are projected to the corresponding lower dimensional subspaces. We elaborate and illustrate such properties in Figure 3.3 and 3.4.

Table 3.8: Templates of questions. In the table, (player), (club), (position) are placeholders of named entities with associated type. (country_1) is a placeholder for a country name while (country_2) is a placeholder for the adjectival form of a country.

#	Query template	Question template
1	Find $e \in \mathcal{E}$: ((player), plays_in_club,e) is true	which club does (player) play for ? which professional football team does (player) play for ? which football club does (player) play for ?
2	Find $e \in \mathcal{E}$: ((player), plays_position, e) is true	what position does (player) play ? what position does (player) play on the field? what is the position of (player) ?
3	Find $e \in \mathcal{E}$: ((player), is_aged, e) is true	how old is (player) ? what is the age of (player) ?
4	Find $e \in \mathcal{E}$: ((player), wears_number, e) is true	what is the jersey number of (player) ? what number does (player) wear ?
5	Find $e \in \mathcal{E}$: ((player), plays_for_country, e) is true	what is the nationality of (player) ? which national football team does (player) play for ? which national soccer team does (player) play for ? where is (player) from ? which country is (player) from ?
6	Find $e \in \mathcal{E}$: ((club), is_in_country, e) is true	which country is the football team (club) based in ? where is the football team (club) located ? which country is (club) located in ? which country is the football team (club) located in ? which country is the professional football team (club) located in ? which country is (club) based in ?
7	Find $e \in \mathcal{E}$: ((club), plays_in_club^{-1}, e) is true	name a player from (club) ? who plays for (club) ? who plays at the soccer club (club) ? who is from the professional football team (club) ? who plays professionally at (club) ?
8	Find $e \in \mathcal{E}$: ((country_1), plays_for_country^{-1}, e) is true	which player is from (country_1) ? name a player from (country_1) ? who is from (country_1) ? who plays for the (country_1) national football team ?
9	Find $e \in \mathcal{E}$: ((position), plays_position^{-1}, e) is true	name a player who plays (position) ? who plays (position) ? name a football player who plays (position) ? which football player plays (position) ?
10	Find $e \in \mathcal{E}$: ((country_1), is_in_country^{-1}, e) is true	which soccer club is based in (country_1) ? which football club is based in (country_1) ? which football club is located in (country_1) ? which professional football team is located in (country_1) ? name a soccer club in (country_1) ? name a football club in (country_1) ?
11	Find $e \in \mathcal{E}$: ((player), plays_in_club / is_in_country, e) is true	which country does (player) play professionally in ? where is the football club that (player) plays for ?
12	Find $e \in \mathcal{E}$: ((country_1), plays_for_country^{-1} / plays_in_club, e) is true	which professional football team do players from (country_1) play for ? name a soccer club that has a player from (country_1) ? name a professional football team that has a player from (country_1) ? name a soccer club that has a (country_2) player ? name a professional football team that has a (country_2) player ? which professional team has a (country_2) player ? which professional soccer team has a (country_2) player ? which professional football team has a player from (country_1) ?
13	Find $e \in \mathcal{E}$: ((position), plays_position^{-1}, e) is true and ((club), plays_in_club^{-1}, e) is true	who plays (position) for (club)? who are the (position) at (club) ? name a (position) that plays for (club) ?
14	Find $e \in \mathcal{E}$: ((position), plays_position^{-1}, e) is true and ((country_1), plays_for_country^{-1}, e) is true	who plays (position) for (country_1) ? who are the (position) on (country_1) national team ? name a (position) from (country_1) ? which (country_2) footballer plays (position) ? who is a (country_2) (position) ? name a (country_2) (position) ?
15	Find $e \in \mathcal{E}$: ((club), plays_in_club^{-1}, e) is true and ((country_1), plays_for_country^{-1}, e) is true	who are the (country_2) players at (club) ? which (country_2) footballer plays for (club) ? name a (country_2) player at (club) ? which player in (club) is from (country_1) ?

Baseline methods We also trained a TransGaussian model only on the atomic triplets and denote such a model by *TransGaussian (SINGLE)*. Since no inverse relation was involved

Table 3.9: (Composed) relations and sample questions in path queries.

#	Relation	Type	Sample question	Sample answer
1	plays_in_club	many-to-one	which club does alan pulido play for ? which professional football team does klaas jan huntelaar play for ?	tigres_uanl fc_schalke_04
2	plays_position	many-to-one	what position does gonzalo higuain play ?	forward
3	is_aged	many-to-one	how old is samuel etoo ? what is the age of luis suarez ?	33 27
4	wears_number	many-to-one	what is the jersey number of mario balotelli ? what number does shinji okazaki wear ?	9 9
5	plays_for_country	many-to-one	which country is thomas mueller from ? what is the nationality of helder postiga ?	germany portugal
6	is_in_country	many-to-one	which country is the soccer team fc porto based in ?	portugal
7	plays_in_club^{-1}	one-to-many	who plays professionally at liverpool fc ? name a player from as roma ?	steven_gerrard miralem_pjanic
8	plays_for_country^{-1}	one-to-many	which player is from iran ? name a player from italy ?	masoud_shojaei daniele_de_rossi
9	plays_position^{-1}	one-to-many	name a player who plays goalkeeper ? who plays forward ?	gianluiqi_buffon raul_jimenez
10	is_in_country^{-1}	one-to-many	which soccer club is based in mexico ? name a soccer club in australia ?	cruz_azul_fc melbourne_victory_fc
11	plays_in_club / is_in_country	many-to-one	where is the club that edin dzeko plays for ? which country does sime vrsaljko play professionally in ?	england italy
12	plays_for_country^{-1} / plays_in_club	many-to-many	name a soccer club that has a player from australia ? name a soccer club that has a player from spain ?	crystal_palace_fc fc_barcelona

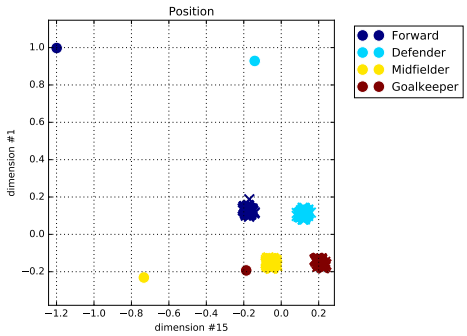
Table 3.10: Conjunctive queries and sample questions.

#	Relations	Sample questions	Entities in questions	Sample answer
13	plays_position^{-1} and plays_in_club^{-1}	who plays forward for fc barcelona ? who are the midfielders at fc bayern muenchen ?	forward , fc_barcelona midfielder, fc_bayern_muenchen	lionel_messi toni_kroos
14	plays_position^{-1} and plays_for_country^{-1}	who are the defenders on german national team ? which mexican footballer plays forward ?	defender , germany defender , mexico	per_mertesacker raul_jimenez
15	plays_in_club^{-1} and plays_for_country^{-1}	which player in paris saint-germain fc is from argentina ? who are the korean players at beijing guoan ?	paris_saint-germain_fc , argentina beijing_guoan , korea	ezequiel_lavezzi ha_daesung

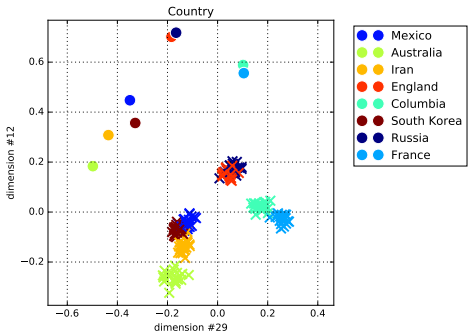
when *TransGaussian (SINGLE)* was trained, to use this embedding in question answering tasks, we represent the inverse relations as follows: for each relation r with mean $\boldsymbol{\delta}_r$ and variance $\boldsymbol{\Sigma}_r$, we model its inverse r^{-1} as a Gaussian attention with mean $-\boldsymbol{\delta}_r$ and variance equal to $\boldsymbol{\Sigma}_r$.

We also trained TransE models on WorldCup2014 dataset by using the code released by the authors of Guu et al. (2015). Likewise, we use *TransE (SINGLE)* to denote the model trained with atomic triplets only and use *TransE (COMP)* to denote the model trained with the union of triplets and paths. Note that TransE can be considered as a special case of TransGaussian where the variance matrix is the identity and hence, the scoring formula Eq. (3.8) is applicable to TransE as well.

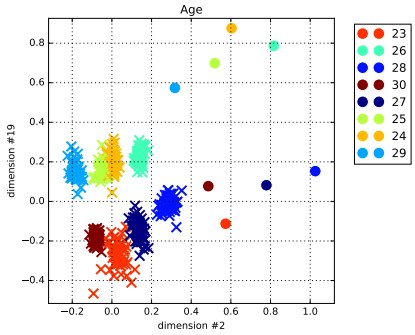
Training configurations For all models, dimension of entity embeddings was set to 30. The hidden size of LSTM was set to 80. Word embeddings were trained jointly with the



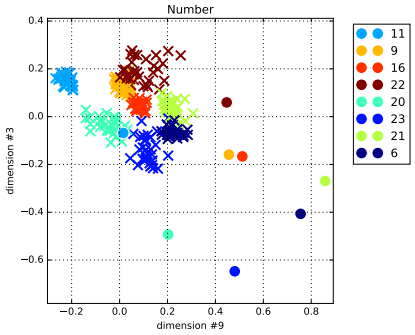
(a) `plays_position`



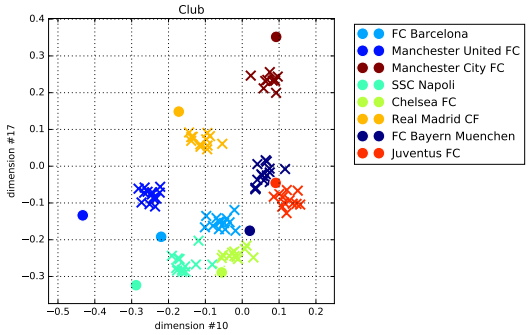
(b) `plays_for_country`



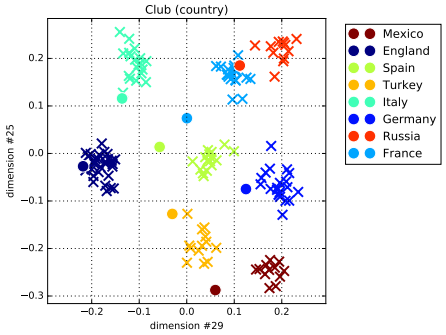
(c) `is_aged`



(d) `wears_number`



(e) `plays_in_club`



(f) `is_in_country`

Figure 3.4: Visualization of TransGaussian entity embeddings. Crosses are the subjects and circles are the objects of a relation. Specifically, crosses are players in (a)-(e) and professional football clubs in (f).

question answering model and dimension of word embedding was set to 40. We employed Adam (Kingma and Ba, 2014) as the optimizer. All parameters were tuned on the validation set. Under the same setting, we experimented with two cases: first, we trained models for path queries and conjunctive queries separately; Furthermore, we trained a single model that

addresses both types queries. We present the results of the latter case in the next subsection while the results of the former are included in the Appendix.

Evaluation metrics During test time, our model receives a question in natural language and a list of knowledge base entities contained in the question. Then it predicts the mean and variance of a Gaussian attention formulated in Eq. (3.8) which is expected to capture the distribution of all positive answers. We rank all entities in the knowledge base by their scores under this Gaussian attention. Next, for each entity which is a correct answer, we check its rank relative to all incorrect answers and call this rank the filtered rank. For example, if a correct entity is ranked above all negative answers except for one, it has filtered rank two. We compute this rank for all true answers and report *mean filtered rank* and *H@1* which is the percentage of true answers that have filtered rank 1.

3.6.3 Experimental results

We present the results of joint learning in Table 3.11. These results show that TransGaussian works better than TransE in general. In fact, *TransGaussian (COMP)* achieved the best performance in almost all aspects. Most notably, it achieved the highest H@1 rates on challenging questions such as "where is the club that edin dzeko plays for?" (#11, composition of two relations) and "who are the defenders on german national team?" (#14, conjunction of two queries).

The same table shows that TransGaussian benefits remarkably from compositional training. For example, compositional training improved TransGaussian's H@1 rate by near 60% in queries on players from a given countries (#8) and queries on players who play a particular position (#9). It also boosted TransGaussian's performance on all conjunctive quries (#13–#15) significantly.

To understand *TransGaussian (COMP)*'s weak performance on answering queries on the

Table 3.11: Results of joint learning with path queries and conjunction queries on WorldCup2014.

#	Sample question	TransE (SINGLE) H@1 (%)	TransE (SINGLE) Mean Filtered Rank	TransE (COMP) H@1 (%)	TransE (COMP) Mean Filtered Rank	TransGaussian (SINGLE) H@1 (%)	TransGaussian (SINGLE) Mean Filtered Rank	TransGaussian (COMP) H@1 (%)	TransGaussian (COMP) Mean Filtered Rank
1	which club does alan pulido play for?	96.40	1.04	97.84	1.02	94.96	1.06	99.28	1.01
2	what position does gonzalo higuain play?	97.99	1.02	99.33	1.01	100.00	1.00	100.00	1.00
3	how old is samuel etoo?	98.71	1.01	97.42	1.03	100.00	1.00	100.00	1.00
4	what is the jersey number of mario balotelli?	98.68	1.01	98.03	1.03	100.00	1.00	100.00	1.00
5	which country is thomas mueller from ?	96.43	1.05	95.71	1.06	99.29	1.01	100.00	1.00
6	which country is the soccer team fc porto based in ?	96.92	1.05	98.46	1.02	76.92	1.69	100.00	1.00
7	who plays professionally at liverpool fc?	97.95	1.03	82.19	1.61	90.41	1.14	97.95	1.02
8	which player is from iran?	93.28	1.34	60.08	3.32	100.00	1.00	100.00	1.00
9	name a player who plays goalkeeper?	98.76	1.01	100.00	1.00	50.31	1.50	100.00	1.00
10	which soccer club is based in mexico?	88.71	1.39	100.00	1.00	90.32	1.35	98.39	1.02
11	where is the club that edin dzeko plays for ?	35.37	5.00	55.78	2.69	25.17	69.30	91.84	1.36
12	name a soccer club that has a player from australia ?	17.09	30.23	29.91	10.88	6.84	49.96	60.68	18.32
	Overall (Path Query)	85.77	3.46	82.68	2.25	79.72	10.46	96.26	2.24
13	who plays forward for fc barcelona?	90.80	1.23	61.35	2.97	98.16	1.02	99.39	1.01
14	who are the defenders on german national team?	67.48	1.78	39.02	7.47	95.93	1.04	100.00	1.00
15	which player in ssc napoli is from argentina?	97.01	1.04	65.67	1.85	96.27	1.07	100.00	1.00
	Overall (Conj. Query)	85.95	1.33	56.19	3.93	96.90	1.04	99.76	1.00

professional football club located in a given country (#10) and queries on professional football club that has players from a particular country (#12), we tested its capability of modeling the composed relation by feeding the correct relations and subjects during test time. It turns out that these two relations were not modeled well by *TransGaussian (COMP)* embedding, which limits its performance in question answering. (See Table 3.12 for quantitative evaluations.) The same limit was found in the other three embeddings as well.

Note that all the models compared in Table 3.11 uses the proposed Gaussian attention model because TransE is the special case of TransGaussian where the variance is fixed to one. Thus the main differences are whether the variance is learned and whether the embedding was trained compositionally.

3.7 Conclusion

In this chapter, we have proposed the Gaussian attention model which can be used in a variety of contexts where we can assume that the distance between the memory items in the latent space is compatible with some notion of semantics. We have shown that the proposed

Table 3.12: Evaluation of embedding of WorldCup2014. We evaluate the embeddings by feeding the correct entities and relations from a path or conjunctive query to an embedding model and using its scoring function to retrieve the answers from the embedded knowledge base.

		TransE (SINGLE)		**TransE (COMP)**		**TransGaussian (SINGLE)**		**TransGaussian (COMP)**	
#	Relation	H@1 (%)	Mean Filtered Rank	H@1 (%)	Mean Filtered Rank	H@1 (%)	Mean Filtered Rank	H@1 (%)	Mean Filtered Rank
1	plays_in_club	99.86	1.00	98.37	1.02	95.65	1.05	100.00	1.00
2	plays_position	100.00	1.00	99.32	1.01	100.00	1.00	100.00	1.00
3	is_aged	98.78	1.01	97.69	1.02	100.00	1.00	99.86	1.00
4	wears_number	98.64	1.01	97.42	1.05	100.00	1.00	100.00	1.00
5	plays_for_country	100.00	1.00	98.10	1.04	100.00	1.00	100.00	1.00
6	is_in_country	100.00	1.00	100.00	1.00	97.31	1.04	100.00	1.00
7	plays_in_club^{-1}	100.00	1.00	95.52	1.07	92.80	1.09	99.32	1.01
8	plays_for_country^{-1}	100.00	1.00	87.50	1.41	100.00	1.00	99.86	1.00
9	plays_position^{-1}	100.00	1.00	100.00	1.00	100.00	1.00	99.86	1.00
10	is_in_country^{-1}	100.00	1.00	100.00	1.00	95.62	1.11	100.00	1.00
11	plays_in_club / is_in_country	36.01	4.84	56.66	2.89	11.14	153.27	93.61	1.71
12	plays_for_country^{-1} / plays_in_club	16.45	39.28	45.47	12.40	11.83	94.19	76.34	24.87
	Overall (Path relations)	87.85	4.04	89.61	2.03	84.06	21.96	97.64	2.73
13	plays_position^{-1} and plays_in_club^{-1}	83.56	1.35	55.30	2.36	97.15	1.03	98.10	1.02
14	plays_position^{-1} and plays_for_country^{-1}	69.02	2.00	44.16	4.37	100.00	1.00	99.59	1.01
15	plays_in_club^{-1} and plays_for_country^{-1}	97.01	1.04	64.54	1.61	97.01	1.04	97.42	1.03
	Overall (Conj. relations)	83.20	1.46	54.66	2.78	98.05	1.02	98.37	1.02

Gaussian scoring function can be used for knowledge base embedding achieving competitive accuracy. We have also shown that our embedding model can naturally propagate uncertainty when we compose relations together. Our embedding model also benefits from compositional training proposed by Guu et al. (2015). Furthermore, we have demonstrated the power of the Gaussian attention model in a challenging question answering problem which involves both composition of relations and conjunction of queries. Future work includes experiments on natural question answering datasets and end-to-end training including the entity extractor. If we use RNNs as a *decoder*, our model would be able to handle non-commutative composition of relations, which the current weighted convolution cannot handle well.

Chapter 4

Low Coherence Frames via Alternating Projections and von Neumann Algebras

4.1 Introduction

Frames are generalization of orthonormal systems and have various application in areas such as signal and image processing, data compression and coding theory. In the literature, people have been interested in frames with low coherence (Strohmer and Heath, 2003; Candès et al., 2009; Donoho et al., 2006; Mixon et al., 2011). In this work, we consider the problem of constructing group frames, frames that are generated by a cyclic vector and a group representation. In particular, we are interested in using non-commutative groups. Inspired by Barbieri et al. (2014) who inspect the relation between frame bound and generating vector in the space of von Neumann space, we inspect the worse-case coherence of a cyclic vector in the space of operators. The problem of existence of group frame with a certain coherence is then transformed to the existence problem of intersection between a convex set and a special subset of rank-one operators in the space of matrices (operators). Before introducing our framework for constructing group frames, we first give a brief review of the necessary background.

4.1.1 Some background of tight frames and lower bound on coherence

We consider construction of frame in d dimensional complex vector space $\mathbb{C}^d$. Throughout this chapter, we use the definition of inner product that is linear in the first argument and conjugate linear in the second, i.e. for two vectors $u, v \in \mathbb{C}^d$, $\langle u,\, v\rangle_{\mathbb{C}^d} := v^* u$; for two matrices $X, Y \in \mathbb{C}^{d\times d}$, $\langle X,\, Y\rangle_{\mathbb{C}^{d\times d}} := \operatorname{trace}(Y^* X)$. The subscript of the angled brackets indicates the space where the inner product is taken. This subscript is omitted wherever it is clear from the context.

Recall that a set of m vectors $\Phi = \{u_1, u_2, \ldots, u_m\} \subset \mathbb{C}^d$ form a frame if and only there there exist constants $A, B > 0$ such that, for any $x \in \mathbb{C}^d$, we have

$$A\|x\|_2^2 \leq \sum_{i=1}^{m} |\langle x,\, u_i\rangle|^2 \leq B\|x\|_2^2.$$

A and B are called frame bounds. When $A = B$, Φ is called a tight frame and if all u_i's have unit norm, it is called a unit norm tight frame.

The (worst-case) coherence of a frame is defined by

$$\mu(\Phi) := \max_{\substack{u,v\in\Phi \\ u\neq v}} \frac{|\langle u,\, v\rangle|}{\|u\|_2 \cdot \|v\|_2}.$$

When the dimension d and number of vectors m are fixed, a frame that minimizes the worst-case coherence is called grassmannian frames (Strohmer and Heath, 2003). For a given pair of d and m, a well-known lower bound for the optimal coherence that a frame with m vectors in a d dimensional complex vector space is $\mu_{m,d} := \sqrt{\frac{m-d}{d(m-1)}}$. This bound is called the Welch bound. It worth noticing that when $m > d^2$, Welch bound is known to be not sharp. For

example, Zorlein and Bossert (2015) (and the reference within) shows a better lower bound

$$\mu(\Phi) \geq \begin{cases} \sqrt{\frac{m-d}{d(m-1)}} \text{ (Welch bound)}, & \text{for } m \leq d^2, \\ \max\left\{\sqrt{\frac{1}{n}}, \sqrt{\frac{2m-d^2-d}{(d+1)(m-d)}}, 1-2m^{-\frac{1}{d-1}}\right\}, & \text{for } d^2 < m \leq 2(d^2-1), \\ \max\left\{\sqrt{\frac{2m-d^2-d}{(d+1)(m-d)}}, 1-2m^{-\frac{1}{d-1}}\right\}, & \text{for } m > 2(d^2-1). \end{cases} \tag{4.1}$$

When $m \leq d^2$, complex frames that reach Welch bound are still only known for a few cases. If vectors in a unit norm tight frame satisfy the condition that absolute values of inner products between any two pairs vectors in the frame are the same, i.e., $\forall i \neq j, \left|\langle u_i, u_j \rangle\right| = C$ for a constant C, then we call it an equiangular unit norm tight frame. Equiangular frames do not exist for every pair of d and m. The work by Sustik et al. (2007); Waldron (2009); Fickus et al. (2012); Renes et al. (2003) has drawn some conditions on the existence of equiangular frames and developed methods for constructing such frames. For example, it is known that equiangular frame can only exists if $m \leq \frac{d(d+1)}{2}$ for a real frame and $m \leq d^2$ for a complex frame. Sustik et al. (2007) provides more detailed conditions on both real and complex equiangular tight frames. On the other hand, Zauner's conjecture, which was origionally posed in Zauner's dissertation, says that the condition for complex equiangular frame is sharp:

Zauner's conjecture (Zauner (1999)) For every $d \geq 2$, there exist equiangular unit norm complex tight frame with d^2 vectors.

Zauner's conjecture has been proved only for a few dimensions. For a good survey and up-to-dated results on equiangular tight frames, we would refer to Fickus and Mixon (2015).

4.1.2 Group frame

We consider frame generated under the action of group representations. In another word, we are interested in frames of the form $\{\pi(g)\psi : g \in G\}$, where G is a finite group, π is a d dimensional representation of G and ψ is a d dimensional complex vector. Frames generated under group actions are called group frames. We denote such a frame by $\mathcal{F}_\pi(\psi)$. To make $\mathcal{F}_\pi(\psi)$ a frame, it is necessary to pick ψ such that $\mathcal{F}_\pi(\psi)$ span the entire space $\mathbb{C}^d$. Thus ψ is necessarily a cyclic vector. Casazza and Kutyniok (2013) and Vale and Waldron (2004) have some characteristics on the tightness of group frames. It is worth to mention the following theorem.

Theorem 4.1.1. *(Casazza and Kutyniok (2013), Theorem 5.4) Let G be a finite group which acts on $\mathcal{H}$ as unitary transformations, and let*

$$\mathcal{H} = V_1 \oplus V_2 \oplus \cdots \oplus V_M$$

be an orthonormal direct sum of irreducible $\mathbb{F}G$-modules for which repeated summands are absolutely irreducible. Then $\Phi = \{\rho(g)v : g \in G\}$, $v = v_1 + v_2 + \cdots + v_M$, $v_j \in V_j$ is a tight G-frame if and only if

$$\frac{\|v_j\|^2}{\|v_k\|^2} = \frac{\dim V_j}{\dim V_k}, \forall j, k,$$

and $\langle \sigma v_j, v_k \rangle = 0$ when V_j is $\mathbb{F}G$-isomorphic to V_k via $\sigma : V_j \to V_k$. By Schur's lemma there is at most one σ to check.

It is known that, for a given pair of d and m, there is a finite number of tight frames of m vectors for $\mathbb{C}^d$ (up to unitary equivalence) which are given by the orbit of an abelian group of $d \times d$ matrices (Casazza and Kutyniok (2013)). See Vale and Waldron (2004); Han and Larson (2000); Thill (2016) for introductions and reviews on group frames. Xia et al.

(2005) discovered the connection between complex equiangular tight frames and difference sets of Abelian groups. More specifically, they use difference sets for choosing rows of the Fourier matrix associated with an Abelian group and proved that the rows form a frame with low coherence. Thill et al. (2014) dealt with non-Abelian group and constructed frames by choosing subsets of rows in generalized Fourier matrices. Meanwhile, Thill and Hassibi (2015) proposed a framework for constructing frame with low coherence by controlling number of distinct inner products between frame vectors.

4.1.3 Numerical methods

Analytic constructions of frames that reach the optimal lower bound are only known for very a few cases. People have been resort to numerical methods for searching frames with low coherence. The work of Scott and Grassl (2010) produced frames with d^2 vectors in d dimensional complex vector space that satisfy conditions for equiangular tight frames within machine precision up to $d = 67$. They worked with group frames generated from the Heisenberg group and their numerical method searches for a cyclic vector by directly minimizing the sum of absolute values of inner products raised to the fourth power

$$\min_{\psi\in\mathbb{C}^d,\|\psi\|_2=1} \frac{1}{d} \sum_{p\in\mathbb{Z}_d\times\mathbb{Z}_d} \left|\psi^* D_p \psi\right|^4 = \sum_{j,k\in\mathbb{Z}_d} \left| \sum_{l\in\mathbb{Z}_d} \overline{\psi(j+l)}\psi(l)\overline{\psi(k+l)}\psi(j+k+l) \right|^2$$

where $\left\{D_p : p \in \mathbb{Z}_d \times \mathbb{Z}_d\right\}$ is a set of unitary matrices from the representation of Heisenberg group. See Scott and Grassl (2010) for details of the definitions.

Zorlein and Bossert (2015) designed an algorithm for searching optimal complex spherical code which equivalently searches for frames with low coherence. Their search is based on

minimizing a generalized potential function

$$\min_{\substack{\psi_j \in \mathbb{C}^d, \|\psi_j\|_2 = 1 \\ j=1,\ldots,m}} \sum_{l=1}^{m} \sum_{k<l} \|\psi_l - \psi_k\|_2^{-(\nu-2)}$$

where $\nu > 2$ is an integer that is large enough.

Tropp et al. (2005) work with the Gram matrix $\Phi^*\Phi$. They use an alternating projection method to search for a low-rank Gram matrix with bounded off-diagonal entries. Tsiligianni et al. (2014) proposed alternating projection methods under a similar idea, but their algorithms involve projection onto more than two sets.

Our work aims to design finite dimensional complex group frames by searching for a cyclic vector that leads to low coherence. It is motivated by the idea of transforming the inner products between vectors to inner products between operators under the spirit of Barbieri et al. (2014). We will first formulate the structure of group frames in von Neumann algebra associated with the generating group and formulate the coherence in terms of intersection between two sets of operators. Then, we propose an alternating projection method to search for such an intersection.

4.2 Formulation and algorithms

In the work by Barbieri et al. (2014), the authors characterize frames generated from the action of a countable discrete group G on a single element ψ on a Hilbert space $\mathcal{H}$. Suppose T is a unitary representation of G defined on $\mathcal{H}$ and let $L^1(\mathfrak{N}(G))$ be the L^1-space associated to the group von Neumann algebra. The core of their characterization is a bracket map

$$[\cdot, \cdot] : \mathcal{H} \times \mathcal{H} \to L^1(\mathfrak{N}(G))$$

which satisfies

$$\langle \varphi,\, T(\gamma)\psi \rangle_{\mathcal{H}} = \tau\left([\varphi, \psi]\, \lambda(\gamma)^*\right)$$

for every $\varphi, \psi \in \mathcal{H}$ and $\gamma \in G$. Here, $\lambda(\cdot)$ is the regular representation on the group algebra and $\tau(\cdot)$ is a trace defined on $\mathfrak{N}(G)$.

We apply the same idea to unitary representations of finite groups and characterize the worst-case coherence of a finite group frame in terms of von Neumann algebra associated with the group. Note that when it comes to finite groups, group algebra can be expressed in terms of finite vector space and operators over the group algebra can be expressed as square matrices.

Besides, when the group has a non-trivial center and the representation of $g \in Z(G)$ is a scalar multiply the identity matrix, $|\langle \psi, \pi(g)\psi \rangle| = 1$ for any unit norm vector ψ. In such a case, it makes sense to only the quotient group $G/Z(G)$ in constructing the frame, avoiding including two vectors that only differ by a phase. For example, when $\{\pi(g) : g \in Z(G)\} \subset \{c \cdot I : c \in \mathbb{C}, |c| = 1\}$, one may pick a set of representatives from the set of cosets $\{gZ(G) : g \in G\}$ and only use these operators to construct a frame. Unitary operators associated with such a subset is called a group-like unitary system by Gabardo et al. (2003). The discussion in the rest of this section assumes that the group has a trivial center and every group element is used in the construction of the frame. Yet, as one can see through the discussion, it is not be difficult to generalize our framework to groups with non-trivial center as well.

4.2.1 von Neumann algebra and decomposition of operators

Let G be a finite group of order n. Let $\mathbb{C}(G)$ be the space of functions over G which is isomorphic to the n dimensional vector space $\mathbb{C}^n$. We index each dimension of $\mathbb{C}^n$ with a

unique group element and use $\{\delta_g : g \in G\}$ to denote its natural basis. Let L be the left regular representation of G defined by

$$L : G \to \mathcal{U}(\mathbb{C}(G)) \cong \mathcal{U}(\mathbb{C}^n)$$
$$\gamma \mapsto \left(\delta_g \mapsto \delta_{\gamma g}\right).$$

Thus, each operator $L(g)$ can be considered as an n-by-n permutation matrix. We index rows and columns of n-by-n matrices by group elements as well. Denote the group von Neumann algebra for G by $\mathfrak{N}(G)$ which is the closure of linear span of operators $\{L(g) : g \in G\}$ in $\mathbb{C}^{n\times n}$. i.e.

$$\mathfrak{N}(G) := \text{span}\left(\{L(g) : g \in G\}\right) = \left\{\sum_{g\in G} c_g L(g) : c_g \in \mathbb{C}\right\} \subset \mathbb{C}^{n\times n}.$$

Let $\mathfrak{N}(G)^\perp$ be its orthogonal complement in $\mathbb{C}^{n\times n}$.

Since $\langle L(\gamma),\, L(g)\rangle_{\mathbb{C}^{n\times n}} = 0$ whenever $\gamma \neq g$, $\left\{\frac{1}{\sqrt{n}}L(g) : g \in G\right\}$ is an orthonormal basis of $\mathfrak{N}(G)$. Therefore, any $A \in \mathbb{C}^{n\times n}$ can be decomposed as

$$\begin{aligned} A &= \sum_{g\in G} \left\langle A,\, \frac{1}{\sqrt{n}}L(g)\right\rangle_{\mathbb{C}^{n\times n}} \frac{1}{\sqrt{n}}L(g) + \mathbf{Proj}_{\mathfrak{N}(G)^\perp}(A) \\ &= \frac{1}{n}\Big(\text{trace}(A) + \sum_{g\in G\setminus\{e\}} \langle A,\, L(g)\rangle_{\mathbb{C}^{n\times n}}\, L(g)\Big) + \mathbf{Proj}_{\mathfrak{N}(G)^\perp}(A) \end{aligned} \tag{4.2}$$

where $\mathbf{Proj}_{\mathfrak{N}(G)^\perp}$ is the projection to $\mathfrak{N}(G)^\perp$.

Suppose π is a subrepresentation of L in the space $\mathbb{C}^d$. It is well-known that the left regular representation of a finite group is equivalent to the direct sum of all irreducible representations of G with certain multiplicity. This means there exists an unitary matrix P_π that simultaneously transforms $L(g)$ into a block diagonal matrix with $\pi(g)$ on its top left

block for all $g \in G$. i.e., $P_\pi L(g) P_\pi^*$ takes the form

$$P_\pi L(g) P_\pi^* = \begin{bmatrix} \pi(g) & \\ & \rho(g) \end{bmatrix}, \quad \text{for any } g \in G. \tag{4.3}$$

4.2.2 Coherence of group frames

As stated in (4.3), we can decompose the representation of an regular representation into the direct sum of stable subspaces associated with the irreducibles. Likewise, we can regard the representation space of an irreducible π as a stable subspace of $\mathbb{C}^N$ which is the representation space of the regular representation.

Proposition 4.2.1. *For any $g \in G$ and $\phi, \psi \in \mathbb{C}^d$,*

$$\langle \phi,\, \pi(g)\psi \rangle_{\mathbb{C}^d} = \left\langle P_\pi^* \begin{bmatrix} \phi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi,\, L(g) \right\rangle_{\mathbb{C}^{n\times n}}.$$

Proof.

$$\begin{aligned}
\langle \phi,\, \pi(g)\psi \rangle_{\mathbb{C}^d} &= \left\langle \begin{bmatrix} \phi \\ 0 \end{bmatrix}, \begin{bmatrix} \pi(g) & \\ & \rho(g) \end{bmatrix} \begin{bmatrix} \psi \\ 0 \end{bmatrix} \right\rangle_{\mathbb{C}^n} \\
&= \left\langle P_\pi^* \begin{bmatrix} \phi \\ 0 \end{bmatrix},\, L(g) P_\pi^* \begin{bmatrix} \psi \\ 0 \end{bmatrix} \right\rangle_{\mathbb{C}^n} \\
&= \operatorname{trace}\left(P_\pi^* \begin{bmatrix} \phi \\ 0 \end{bmatrix} \begin{bmatrix} \psi^* & 0 \end{bmatrix} P_\pi L(g)^* \right) \\
&= \left\langle P_\pi^* \begin{bmatrix} \phi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi,\, L(g) \right\rangle_{\mathbb{C}^{n\times n}}.
\end{aligned}$$

□

By this proposition, we can express the inner product of any two vector in the group frame as an inner product between two matrices,

$$\langle \psi,\, \pi(g)\psi \rangle_{\mathbb{C}^d} = \left\langle P_\pi^* \begin{bmatrix} \psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi,\, L(g) \right\rangle_{\mathbb{C}^{n\times n}}. \tag{4.4}$$

Now, consider Φ_π as a mapping from $\mathbb{C}^d$ to $\mathbb{C}^{n\times n}$

$$\begin{aligned} \Phi_\pi : \mathbb{C}^d &\to \mathbb{C}^{n\times n} \\ \psi &\mapsto P_\pi^* \begin{bmatrix} \psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi. \end{aligned}$$

This mapping plays a similar role as the bracket mapping $[\cdot,\cdot]$ defined in Barbieri et al. (2014). For any cyclic vector ψ, let

$$c_g(\psi) := \langle \psi,\, \pi(g)\psi \rangle = \left\langle P_\pi^* \begin{bmatrix} \psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi,\, L(g) \right\rangle = \langle \Phi_\pi(\psi),\, L(g) \rangle. \tag{4.5}$$

By using (4.2), we can uniquely expand $\Phi_\pi(\psi)$ as

$$\Phi_\pi(\psi) = \frac{1}{n} \sum_{g\in G} c_g(\psi) L(g) + \mathbf{Proj}_{\mathfrak{N}(G)^\perp} \left(\Phi_\pi(\psi) \right). \tag{4.6}$$

Based on the definition of $c_g(\psi)$, we have the following proposition.

Proposition 4.2.2. *Given a unit norm cyclic vector $\psi \in \mathbb{C}^d$, the worst-case coherence of the dictionary $\{\pi(g)\psi : g \in G\}$ is equal to $\max_{g\in G\setminus\{e\}} \left| c_g(\psi) \right|$.*

Denote the range of Φ_π by $\mathcal{R}\left(\Phi_\pi\right)$. $\mathcal{R}\left(\Phi_\pi\right)$ is a subset of rank-one positive semi-definite Hermitian matrices in $\mathbb{C}^{n\times n}$. Meanwhile, let $\mathcal{S}_d$ be the unit sphere in $\mathbb{C}^d$ and $\mathcal{R}\left(\Phi_\pi, \mathcal{S}_d\right)$ be the range of Φ_π applied to all unit-norm vectors, then $\mathcal{R}\left(\Phi_\pi, \mathcal{S}_d\right) := \{\Phi_\pi(\psi) : \|\psi\|_2 = 1\}$

which is a *bounded* subset of $\mathcal{R}(\Phi_\pi)$. Since we are looking for a unit norm cyclic vector that generates a dictionary with bounded coherence, it is useful to define the following convex subset $\Gamma_c \subset \mathbb{C}^{n\times n}$.

Definition 4.2.1. *Given $C > 0$,*

- *define the set $V_c \subset \mathfrak{N}(G)$ as*

$$V_c := \left\{ \frac{1}{n}\left(I + \sum_{g\in G\backslash\{e\}} c_g L(g) \right) : |c_g| \leq C, \forall g \in G\backslash\{e\} \right\};$$

- *define the set $\Gamma_c \subset \mathbb{C}^{n\times n}$ as*

$$\Gamma_c := \left\{ X + Y : X \in V_c,\, Y \in \mathfrak{N}(G)^\perp \right\}.$$

Notice that any matrix in Γ_c has trace equal to one and therefore $\mathcal{R}(\Phi_\pi) \cap \Gamma_c = \mathcal{R}(\Phi_\pi, \mathcal{S}_d) \cap \Gamma_c$. Immediately we have:

Proposition 4.2.3. *Based on the definitions above, we have the following facts:*

- *Given $C > 0$, if $\mathcal{R}(\Phi_\pi) \cap \Gamma_C \neq \varnothing$, then there exists a unit-norm cyclic vector which, under the action of π, generates a dictionary with coherence smaller than or equal to C.*

- *The worst-case coherence is the minimum C such that $\mathcal{R}(\Phi_\pi) \cap \Gamma_C \neq \varnothing$.*

4.2.3 Alternating projection

The previous subsection states that finding a cyclic vector that generates a frame with worst-case coherence no larger than C is equivalent to finding the intersection between $\mathcal{R}(\Phi_\pi)$ (or $\mathcal{R}(\Phi_\pi, \mathcal{S}_d)$) and Γ_C. It is natural to consider the alternating projection method for finding such an intersection. The algorithm boils down to solving the following problems,

- **Problem 1** Given an $X \in \mathcal{R}(\Phi_\pi)$, solve $\arg\min_{A\in\Gamma_C} \|A - X\|_F$;

- **Problem 2** Given an $A \in \Gamma_C$, solve $\arg\min_{\psi\in\mathbb{C}^d} \|A - \Phi_\pi(\psi)\|_F$.

Propositions below give the solutions to the problems.

Proposition 4.2.4. *Given $C > 0$ and $\psi \in \mathbb{C}^d$, let $c_g(\psi) = \langle \psi, \pi(g)\psi\rangle$ be the same as defined by (4.5). For each $g \in G\backslash\{e\}$, let*

$$\tilde{c}_g(\psi) = \begin{cases} C \cdot \frac{c_g(\psi)}{|c_g(\psi)|}, & \text{if } |c_g(\psi)| > C, \\ c_g(\psi), & \text{otherwise.} \end{cases}$$

Define the projection from $\mathcal{R}(\Phi)$ to Γ_C by

$$\mathbf{Proj}_{\Gamma_C}\big(\Phi(\psi)\big) := \frac{1}{n}\Big(I + \sum_{g\in G\backslash\{e\}} \tilde{c}_g(\psi)L(g)\Big) + \mathbf{Proj}_{\mathfrak{N}(G)^\perp}\big(\Phi_\pi(\psi)\big). \tag{4.7}$$

Then $\mathbf{Proj}_{\Gamma_C}(\Phi(\psi))$ gives the unique solution to $\arg\min_{A\in\Gamma_C} \|A - \Phi_\pi(\psi)\|_F$. Meanwhile,

$$\min_{A\in\Gamma_C} \|A - \Phi_\pi(\psi)\|_F^2 = \frac{1}{n}\bigg((\|\psi\|_2^2 - 1)^2 + \sum_{g\in G\backslash\{e\}} \big(\tilde{c}_g(\psi) - c_g(\psi)\big)^2\bigg). \tag{4.8}$$

Proof. Given ψ, by 4.6, $\Phi_\pi(\psi)$ can be decomposed as

$$\begin{aligned} \Phi_\pi(\psi) &= \frac{1}{n}\sum_{g\in G} c_g(\psi)L(g) + \mathbf{Proj}_{\mathfrak{N}(G)^\perp}\left(\Phi_\pi(\psi)\right) \\ &= \frac{1}{n}\Big(\|\psi\|_2^2 I + \sum_{g\in G\backslash\{e\}} c_g(\psi)L(g)\Big) + \mathbf{Proj}_{\mathfrak{N}(G)^\perp}\left(\Phi_\pi(\psi)\right). \end{aligned}$$

Suppose $A \in \Gamma_C$ and it has the decomposition

$$A = \frac{1}{n}\Big(I + \sum_{g\in G\backslash\{e\}} a_g L(g)\Big) + A^\perp$$

and we need to find a_g and $A^\perp \in \mathfrak{N}(G)^\perp$ such that $\|A - \Phi_\pi(\psi)\|_F$ is minimized under the constraints that $\left|a_g\right| < C$, for any $g \in G\backslash\{e\}$. Due to the fact that $\{L(g)\}$ is an orthogonal basis of $\mathfrak{N}(G)$, we have

$$\begin{aligned}\|A - \Phi_\pi(\psi)\|_F^2 &= \frac{1}{n}\Big(\big\|(1 - \operatorname{trace}\Phi_\pi(\psi))I\big\|_F^2 + \sum_{g\in G\backslash\{e\}} \big\|(a_g - c_g(\psi))L(g)\big\|_F^2\Big) \\ &\quad + \left\|A^\perp - \mathbf{Proj}_{\mathfrak{N}(G)^\perp}\left(\Phi_\pi(\psi)\right)\right\|_F^2 \\ &= \frac{1}{n}\Big(\Big(1 - \|\psi\|_2^2\Big)^2 + \sum_{g\in G\backslash\{e\}} \left|a_g - c_g(\psi)\right|^2\Big) \\ &\quad + \left\|A^\perp - \mathbf{Proj}_{\mathfrak{N}(G)^\perp}\left(\Phi_\pi(\psi)\right)\right\|_F^2.\end{aligned}$$

The second equality used the fact that $\operatorname{trace}(\Phi(\psi))/\sqrt{n} = \|\psi\|_2^2/\sqrt{n}$. The minimum of the last expression can be reached by setting $A^\perp$ to $\mathbf{Proj}_{\mathfrak{N}(G)^\perp}\left(\Phi_\pi(\psi)\right)$ and setting a_g to $\tilde{c}_g(\psi)$. i.e., the minimum is reached when A is equal to $\mathbf{Proj}_{\Gamma_C}(\psi)$ defined by (4.7).

Formula (4.8) follows from the fact that $\{L(g)/\sqrt{n} : g \in G\}$ is orthonormal. □

Proposition 4.2.5. *Given an $A \in \Gamma_C$, suppose H is the top-left d-by-d block of the matrix $P_\pi A P_\pi^*$. Suppose v is the leading singular vector of $\frac{1}{2}(H + H^*)$ and σ_1 is the largest singular value. Define*

$$\mathbf{Proj}_{\mathcal{R}(\Phi_\pi)}(A) := P_\pi^* \begin{bmatrix} \sigma_1 v v^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi. \tag{4.9}$$

Then $\mathbf{Proj}_{\mathcal{R}(\Phi_\pi)}(A)$ gives a solution to $\arg\min_{X\in\mathcal{R}(\Phi_\pi)} \|A - X\|_F$. When σ_1 is strictly larger than the next singular value, the solution is unique.

Proof. Since Frobenius norm is unitarily invariant, for any $X = P_\pi^* \begin{bmatrix} \sigma\psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi \in \mathcal{R}\left(\Phi_\pi\right)$,

we have

$$\begin{aligned}\left\| A - P_\pi^* \begin{bmatrix} \sigma\psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi \right\|_F &= \left\| P_\pi A P_\pi^* - \begin{bmatrix} \sigma\psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} \right\|_F \\ &= \|H - \sigma\psi\psi^*\|_F \\ &= \left\| \frac{1}{2}(H + H^*) - \sigma\psi\psi^* \right\|_F\end{aligned}$$

and its minimum can be reached by setting ψ and σ to the leading singular vector and singular value of $\frac{1}{2}(H + H^*)$. □

Input: Group G, d dimensional representation π, upper bound of worst-case coherence C;
Output: Cyclic vector $\psi \in \mathbb{C}^d$
Initialize the unit-norm vector ψ randomly; Set $X \leftarrow \Phi_\pi(\psi)$.
while *not converged* **do**
1. Update $A \leftarrow \mathbf{Proj}_{\Gamma_C}(X)$ according to (4.7);
2. Update $X \leftarrow \mathbf{Proj}_{\mathcal{R}(\Phi_\pi, \mathcal{S}_d)}(A)$ according to (4.9).

end
$X \in \mathcal{R}(\Phi_\pi)$ can be decomposed as $X = P_\pi^* \begin{bmatrix} vv^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi$.
When converged, $\|v\|_2 = 1$. Set $\psi = v$ and return ψ.

Algorithm 1: Alternating projection

Next, we show that the alternating projection method in $\mathbb{C}^{n\times n}$ is actually equivalent to running iteration in $\mathbb{C}^d$ in the following way.

Proposition 4.2.6. *Given $C > 0$ and a cyclic vector $\psi \in \mathbb{C}^d$, for each $g \in G\backslash\{e\}$, let $c_g(\psi)$ be the same as* (4.5)*. Set $\Delta c_g(\psi)$ to*

$$\Delta c_g(\psi) = \begin{cases} c_g(\psi)\left(\frac{C}{|c_g(\psi)|} - 1\right), & \text{if } |c_g(\psi)| > C, \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

The following iterations lead to the same solution as the alternating projection method.

Input: Group G, d dimensional representation π, upper bound of worst-case coherence C;
Output: Cyclic vector $\psi \in \mathbb{C}^d$.
Initialize the unit-norm vector $\psi^{(0)}$ randomly;
Set $H^{(0)} = \psi^{(0)}\psi^{(0)*}$;
Initialize $t = 0$.
while *not converged* **do**

1. Update H:
   Compute $\Delta c_g(\psi^{(t)})$ according to (4.10);
   Set $H^{(t+1)} \leftarrow \psi^{(t)}\psi^{(t)*} + \frac{1}{n}\sum_{g\in G\setminus\{e\}} \Delta c_g(\psi^{(t)})\pi(g)$;
2. Update ψ:
   Set $\psi^{(t+1)}$ to the leading singular vector of $\frac{1}{2}\left(H^{(t+1)} + H^{(t+1)*}\right)$.
3. Update $t \leftarrow t + 1$.

end
Return $\psi^{(t)}$.

Algorithm 2: Counterpart of the alternating projection in $\mathbb{C}^d$.

1. *$H \leftarrow \psi\psi^* + \frac{1}{n}\left(1 - \|\psi\psi^*\|_2^2\right) I + \frac{1}{n}\sum_{g\in G\setminus\{e\}} \Delta c_g(\psi)\pi(g)$;*
2. *Set v and σ to the leading singular vector and singular value of $\frac{1}{2}\left(H + H^*\right)$.*
3. *Set ψ to $\sqrt{\sigma}v$.*

Proof. By re-writting (4.7), we find the equivalent form of the projection,

$$\begin{aligned}
\mathbf{Proj}_{\Gamma_C}\left(\Phi(\psi)\right) &= \frac{1}{n}\left(I + \sum_{g\in G\setminus\{e\}} c_g(\psi)L(g)\right) + \mathbf{Proj}_{\mathfrak{N}(G)^\perp}\left(\Phi_\pi(\psi)\right) \\
&\quad + \frac{1}{n}\sum_{g\in G\setminus\{e\}} \Delta c_g(\pi)L(g) \\
&= \Phi(\psi) + \frac{1}{n}\left(1 - \|\psi\psi^*\|_2^2\right) I + \frac{1}{n}\sum_{g\in G\setminus\{e\}} \Delta c_g(\pi)L(g) \\
&= P^*\left(\begin{bmatrix} \psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{n}\left(1 - \|\psi\psi^*\|_2^2\right) I + \frac{1}{n}\sum_{g\in G\setminus\{e\}} \Delta c_g(\pi) \begin{bmatrix} \pi(g) & \\ & \rho(g) \end{bmatrix}\right) P
\end{aligned}$$

Thus, the top-left block of the term in the parenthesis would be equal to $\psi\psi^* + \frac{1}{n}\left(1 - \|\psi\psi^*\|_2^2\right) I + \frac{1}{n}\sum_{g\in G\setminus\{e\}} \Delta c_g(\psi)\pi(g)$ which we denote as H here. And step 2 and 3 is equivalent the up-

date in (4.9) which takes the leading singular vector and singular value of $\frac{1}{2}(H + H^*)$. $\square$

The two-steps updates in Proposition (4.2.6) can be written in a tighter form. Let $\Omega \subset \mathbb{C}^{d\times d}$ be the set of all rank-one Hermitian matrices. Define $\mathbf{Proj}_\Omega$ to be the operator that projects a matrices to Ω. Then for a matrix $H \in \mathbb{C}^{d\times d}$, $\mathbf{Proj}_\Omega(H)$ can be computed efficiently by taking the leading singular vector v and singular value σ of $\frac{1}{2}(H + H^*)$ and computing σvv^* which corresponds to the step (2) and (3) in Proposition (4.2.6). By this definition, the updates in Proposition (4.2.6) can be written as updating a rank-one matrix H in a tighter form:

$$H \leftarrow \mathbf{Proj}_\Omega\bigg(H + (1 - \mathrm{trace}(H))I + \frac{1}{n}\sum_{g\in G\setminus\{e\}} \Delta c_g(H)\pi(g)\bigg). \tag{4.11}$$

where $\Delta c_g(H)$, the analogy of $c_g(\psi)$, is defined by

$$\Delta c_g(H) = \begin{cases} \langle H,\, \pi(g)\rangle \left(\frac{C}{|\langle H,\pi(g)\rangle|} - 1\right), & \text{if } |\langle H,\, \pi(g)\rangle| > C, \\ 0, & \text{otherwise.} \end{cases}$$

This update can be regarded as using projected gradient[1] method with a fixed step-size in solving the following optimization problem.

$$\min_{H\in\mathbb{C}^{d\times d}} \frac{1}{2}(1 - \mathrm{trace}(H))^2 + \sum_{g\in G\setminus\{e\}} \ell_C(H, \pi(g))$$
$$\text{s.t. } H \in \Omega.$$

1. The objective is a real-valued function of complex variables and it is not holomorphic. Hence, its complex derivative does not exist. The gradient here is computed by treating $\mathbb{C}^{d\times d}$ as $\mathbb{R}^{d\times d} \times \mathbb{R}^{d\times d}$.

where $\ell_C(H, \pi(g))$ is defined by

$$\ell_C(H, \pi(g)) = \begin{cases} \frac{1}{2}\left(|\langle H,\, \pi(g)\rangle| - C\right)^2, & \text{if } |\langle H,\, \pi(g)\rangle| > C, \\ 0, & \text{otherwise} \end{cases} \tag{4.12}$$

Note that this objective function of the miminization problem only differs from $\min_{A\in\Gamma_C} \|A - H\|_F^2$ defined in (4.8) by a constant multiplier. Therefore this objective indicates the distance between the consecutive points generated during the iterations of alternating projection. The objective reaches zero if and only if the intersection between the two sets is found.

4.2.4 Convergence of the algorithm

While the alternating projection between two compact convex sets always locates points of minimum distance, it is not necessary the case when one of them is not convex. In our problem, Algorithm (1) projects to Γ_C and $\mathcal{R}(\Phi_\pi)$ alternately. From the construction of the sets, Γ_C is convex. But $\mathcal{R}(\Phi_\pi)$ is a set of rank-one matrices of a particular form and it is non-convex due to the rank constraint. Therefore, even if the two sets intersect, the algorithm is not guaranteed to find a point in the intersection.

However, alternating projection never increases the distance between successive iterates. Suppose Algorithm (1) produces sequences $\{X_j\} \subset \mathcal{R}(\Phi_\pi)$ and $\{A_j\} \subset \Gamma_C$. We may expect the sequence $\|X_j - A_j\|_F$ to converge. If the algorithm is initialized at a point that is close to the intersection, we show that the sequence converges to zero at a linear rate.

Theorem 4.2.7. *Assume the intersection of Γ_C and $\mathcal{R}(\Phi_\pi)$ exists. Let Algorithm 1 start from a point that is close enough to the intersection. Then the algorithm will converge to a point in $\Gamma_C \cap \mathcal{R}(\Phi_\pi)$ at a linear rate.*

Proof. See Section 4.5. □

4.2.5 Variations of the algorithm

In this section, we raise a variation of the algorithm so that it also applies to generation of tight frames from reducible representation. Meanwhile, an acceleration method is also proposed to aquire faster convergence in practice.

Generate tight frame from reducible representations

In this subsection, we extend the algorithm to generating tight frames from reducible group representations that are direct sum of a few inequivalent irreducibles. As stated in Theorem 4.1.1, when the representation π is irreducible, any cyclic vector ψ generates a tight frame. However, when π is not irreducible, ψ needs to satisfy some extra conditions. Therefore, we need to modify our algorithm a bit to produce a tight frame from a reducible representation. Suppose $\pi \cong \rho_1 \oplus \rho_2 \oplus \cdots \oplus \rho_k$ and ρ_i's are inequivalent irreducible unitary representations with dimensions $d_1, \ldots, d_k$ and $\sum_j d_j = d$. Without loss of generality, we may assume that the matrix representations of $\pi(g)$ is block diagonal,

$$\pi(g) = \begin{bmatrix} \rho_1(g) & & \\ & \ddots & \\ & & \rho_k(g) \end{bmatrix}$$

where $\rho_j(g) \in \mathbb{C}^{d_j \times d_j}$. Let $\psi = \begin{bmatrix} \psi_1^\top & \ldots & \psi_k^\top \end{bmatrix}^\top \in \mathbb{C}^d$ be a unit normed vector and $\psi_j \in \mathbb{C}^{d_j}$ be the component corresponding to ρ_j. Then, the necessary and sufficient condition for ψ to generate a tight frame would be

$$\frac{\|\psi_j\|_2^2}{\|\psi_l\|_2^2} = \frac{d_j}{d_l}.$$

Since $\sum_{j=1}^{k} \|\psi_j\|_2^2 = 1$, this is equivalent to

$$\|\psi_j\|_2^2 = \frac{d_j}{d}, \quad \text{for all } j = 1, 2, \ldots, k.$$

Denote the set of vectors that satisfy this condition by

$$\tilde{\mathcal{S}}_d := \left\{ \psi = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_k \end{bmatrix} : \psi_j \in \mathbb{C}^{d_j}, \|\psi_j\|_2^2 = \frac{d_j}{d} \right\}$$

which is a subset of $\mathcal{S}_d = \{\psi : \|\psi\|_2 = 1\}$. Now, the goal becomes searching for the intersection between $\mathcal{R}\big(\Phi_\pi, \tilde{\mathcal{S}}_d\big)$ and Γ_C. Instead of alternately projecting between Γ_C and $\mathcal{R}\big(\Phi_\pi, \tilde{\mathcal{S}}_d\big)$, we can iteratively projecting onto three sets Γ_C, $\mathcal{R}\big(\Phi_\pi, \mathcal{S}_d\big)$ and $\mathcal{R}\big(\Phi_\pi, \tilde{\mathcal{S}}_d\big)$. The projection fromm Γ_C on $\mathcal{R}\left(\Phi_\pi, \mathcal{S}_d\right)$ can be computed in a similar way as (4.9).

Proposition 4.2.8. *Given an $A \in \Gamma_C$, suppose H is the top-left d-by-d block of the matrix $P_\pi A P_\pi^*$. Suppose v is the leading singular vector of $\frac{1}{2}(H + H^*)$. Define*

$$\mathbf{Proj}_{\mathcal{R}(\Phi_\pi, \mathcal{S}_d)}(A) := P_\pi^* \begin{bmatrix} vv^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi. \tag{4.13}$$

Then $\mathbf{Proj}_{\mathcal{R}(\Phi_\pi, \mathcal{S}_d)}(A)$ gives a solution to $\arg\min_{X \in \mathcal{R}(\Phi_\pi, \mathcal{S}_d)} \|A - X\|_F$.

And the projection from $\mathcal{R}\left(\Phi_\pi, \mathcal{S}_d\right)$ to $\mathcal{R}\big(\Phi_\pi, \tilde{\mathcal{S}}_d\big)$ can be computed as the following. Suppose $v = \begin{bmatrix} v_1^\top & \ldots & v_k^\top \end{bmatrix}^\top \in \mathcal{S}_d$. We need to find $\psi = \begin{bmatrix} \psi_1^\top & \ldots & \psi_k^\top \end{bmatrix}^\top \in \tilde{\mathcal{S}}_d$ such that $\|\Phi_\pi(v) - \Phi_\pi(\psi)\|_F^2$ is minimized. By definition of Φ_π, this is equivalent to minimizing

$\|vv^* - \psi\psi^*\|_F^2$ which can be simplified as

$$\begin{aligned}
\|vv^* - \psi\psi^*\|_F^2 &= \langle vv^* - \psi\psi^*,\, vv^* - \psi\psi^* \rangle \\
&= \langle vv^*, vv^* \rangle + \langle \psi\psi^*, \psi\psi^* \rangle - 2\,\langle vv^*, \psi\psi^* \rangle \\
\text{(due to } \|v\|_2 = \|\psi\|_2 = 1) \quad &= 2 - 2\,\langle vv^*, \psi\psi^* \rangle \\
&= 2 - 2\,|v^*\psi|^2 \\
&= 2 - 2\sum_j \left| v_j^* \psi_j \right|^2 .
\end{aligned}$$

Taking into account the constraints $\|\psi_j\|_2^2 = d_j/d$, we can obtain the minimum by setting

$$\psi_j = \sqrt{\frac{d_j}{d}} \frac{v_j}{\|v_j\|_2^2}. \tag{4.14}$$

Incorporate this projection in the framework of Algorithm 2, and we get the algorithm for constructing tight frames from reducible representations. Summary of this algorithm, together with the acceleration scheme discussed below, will be illustrated in Algorithm 3.

An accelerated alternating projection method

Alternating projection often suffers slow convergence. But, due to its wide application, many acceleration schemes have been proposed in the literature, for example Cegielski and Suchocka (2008); Gearhart and Koshy (1989). To find an x in the intersection of two sets A and B by using alternating projection, a general acceleration scheme is

$$x^{(t+1)} = P_A\big(x^{(k)} + \lambda^{(k)}\sigma^{(k)}(P_A P_B x^{(k)} - x^{(k)})\big),$$

where $\lambda^{(k)} \in [0, 2]$ is the relaxation parameter and $\sigma^{(k)} \geq 0$ is the step size. We employ an extrapolation method which is a special case of this scheme. When the alternating projection is between convex sets, the acceleration methods have certain convergence and acceleration

guarantees. However, in our case, the subset of rank-one matrices is not convex. Therefore the theory of accelerated alternating projections does not apply. But, empirically, we observe that the convergence is accelerated significantly.

Suppose $\left(H^{(t)}, \psi^{(t)}\right)$ comes from the t-th iteration of Algorithm 2. Let $\beta \geq 1$ be the parameter of extrapolation. We design the extrapolation step as follows:

- Set $\tilde{H} \leftarrow \beta\psi^{(t+1)}\psi^{(t+1)*} + (1-\beta)\psi^{(t)}\psi^{(t)*}$;
- Set $\tilde{\psi}$ to the leading singular vector of $\frac{1}{2}\left(\tilde{H} + \tilde{H}^*\right)$.
- Normalize $\tilde{\psi}$ for each irreducibles by using (4.14).

If the rank-one Hermitian matrix $\tilde{\psi}\tilde{\psi}^*$ lead to a lower value in the loss function (4.12), we accept this extrapolation and set $\left(H^{(t)}, \psi^{(t)}\right) = \left(\tilde{H}, \tilde{\psi}\right)$. Otherwise, the extrapolation is discarded for this iteration and we take a smaller extrapolation parameter for future iterations. See Algorithm 3 for a summary of the extrapolated algorithm.

4.3 Experiments

4.3.1 Heisenberg group

We first apply our method to Heisenberg group. Let $\mathbb{Z}_d$ be the group of integers modulo d. The Heisenberg group over $\mathbb{Z}_d$ can be written as a group of upper triangular matrices

$$\left\{ \begin{bmatrix} 1 & m & l \\ & 1 & n \\ & & 1 \end{bmatrix} : m, n, l \in \mathbb{Z}_d \right\}$$

under the operation of matrix multiplication.

Input: Group G, d dimensional representation π that is a direct sum of irreducibles ρ_j, each of which has dimension d_j, $j = 1, 2, \ldots, k$. upper bound of worst-case coherence C; extrapolation parameter β.

Output: Cyclic vector $\psi \in \mathbb{C}^d$.

Initialize the unit-norm vector $\psi^{(0)}$ randomly;

Set $H^{(0)} = \psi^{(0)}\psi^{(0)*}$;

Initialize $t = 0$;

while *not converged* **do**

1. Update H:
   - Compute $\Delta c_g(\psi)$ according to (4.10);
   - Set $H^{(t+1)} \leftarrow \psi^{(t)}\psi^{(t)*} + \frac{1}{n}\sum_{g\in G\setminus\{e\}} \Delta c_g(\psi^{(t)})\pi(g)$;
2. Update ψ:
   - Set $\psi^{(t+1)}$ to the leading singular vector of $\frac{1}{2}\left(H^{(t+1)} + H^{(t+1)*}\right)$.
   - Normalize $\psi^{(t+1)}$ for every irreducible by using (4.14).
3. If $t > 0$, extrapolate:
   - Set $\tilde{H} \leftarrow \beta\psi^{(t+1)}\psi^{(t+1)*} + (1-\beta)\psi^{(t)}\psi^{(t)*}$;
   - Set $\tilde{\psi}$ to the leading singular vector of $\frac{1}{2}\left(\tilde{H} + \tilde{H}^*\right)$.
   - Normalize $\tilde{\psi}$ for every irreducible by using (4.14).
   - If $\tilde{\psi}\tilde{\psi}^*$ has a lower loss, i.e. $\ell_C(\tilde{\psi}\tilde{\psi}^*, \pi(g)) < \ell_C(\psi^{(t+1)}\psi^{(t+1)*}, \pi(g))$, set $\left(H^{(t+1)}, \psi^{(t+1)}\right) = \left(\tilde{H}, \tilde{\psi}\right)$.
     Otherwise, decrease extrapolation parameter, set $\beta \leftarrow \frac{1}{2}\beta + \frac{1}{2}$.
4. Update $t \leftarrow t + 1$.

end

Return $\psi^{(t+1)}$.

Algorithm 3: Search for a cyclic vector for a reducible representation.

Let $w_d := e^{2\pi i/d}$ be the d-th primitive root of unit and M, T be the modulation and and translation operators defined on $\mathbb{C}(\mathbb{Z}_d)$:

$$\text{For all } f \in \mathbb{C}(\mathbb{Z}_d), \quad [Mf](t) := w_d^t f(t),$$
$$[Tf](t) := f(t-1).$$

$\mathbf{H}(\mathbb{Z}_d)$ has an irreducible d-dimensional unitary representation $\pi_{\mathbf{H}(\mathbb{Z}_d)}$ defined by

$$\pi_{\mathbf{H}(\mathbb{Z}_d)} : \mathbf{H}(\mathbb{Z}_d) \to \mathcal{U}(\mathbb{C}(\mathbb{Z}_d))$$
$$\begin{bmatrix} 1 & m & l \\ & 1 & n \\ & & 1 \end{bmatrix} \mapsto w_d^l M^n T^{-m}$$

Or, equivalently

$$\left[\pi_{\mathbf{H}(\mathbb{Z}_d)} \left(\begin{bmatrix} 1 & m & l \\ & 1 & n \\ & & 1 \end{bmatrix} \right) f \right] (t) = w_d^{l+nt} f(t+m).$$

Since $\mathbb{C}(\mathbb{Z}_d) \cong \mathbb{C}^d$, the operators M and T can be written as matrices from $\mathbb{C}^{d\times d}$:

$$M = \begin{bmatrix} 1 & & & \\ & w & & \\ & & \ddots & \\ & & & w^{d-1} \end{bmatrix}, \quad T = \begin{bmatrix} & & & 1 \\ & & & \\ & \mathbf{I}_{(d-1)\times(d-1)} & & \end{bmatrix}.$$

$\mathbf{H}(\mathbb{Z}_d)$ has a non-trivial center $\left\{ \begin{bmatrix} 1 & 0 & l \\ & 1 & 0 \\ & & 1 \end{bmatrix} : l \in \mathbb{Z}_d \right\}$ which we denote as $Z\left(\mathbf{H}(\mathbb{Z}_d)\right)$. We would like to find a cyclic vector $\psi \in \mathbb{C}^d$ such that the frame $\{M^n T^m \psi : m, n \in \mathbb{Z}_d\}$ has a low coherence. This unitary operator system $\{M^n T^m : m, n \in \mathbb{Z}_d\}$ can be considered as projective action of Heisenberg group. This frame in $\mathbb{C}^d$ has d^2 vectors. Vectors in this frame can be considered as time-frequency shifts of a generating vector. Hence, this frame is also called a Gabor frame (Gröchenig, 2013).

It is worth mentioning that for some prime $d \geq 5$, Alltop (1980) has constructed a cyclic vector $f \in \mathbb{C}^d$, $f(n) = e^{2\pi i n^3/d}/\sqrt{d}$, $n = 0, 1, \ldots, d-1$, which generates a frame with worst-case coherence $1/\sqrt{d}$. In addition, if we add the natural basis to this frame to get a set of $d(d+1)$ vectors, these vectors can be organized into $d+1$ *mutually unbiased bases* of $\mathbb{C}^d$. It is called mutually unbiased because any two vectors from different basis have inner product with norm equal to $1/\sqrt{d}$ while two distinct vectors from the same basis has inner product equal to 0 (since they are orthogonal). Meanwhile, the worst-case coherence of this set of $d(d+1)$ vectors reaches the lower bound (4.1).

Here, we attempt to construct a frame with d^2 vectors. Due to Welch bound, the optimal coherence we expect is $1/\sqrt{(1+d)}$. If we can find a frame reaches this lower bound, it is necessarily an equiangular tight frame. Zauner's conjecture claims that for every $d \geq 2$, there exists a vector $\psi \in \mathbb{C}^d$ such that $\{M^n T^m \psi : m, n \in \mathbb{Z}_d\}$ is an equiangular tight frame, i.e., the lower bound is achievable.

Assume ψ is a cyclic vector that leads to the lower bound, then we should expect that

$$\begin{aligned} \left| \left\langle \psi, \pi_{\mathbf{H}(\mathbb{Z}_d)} \psi \right\rangle \right| &= 1, \quad \text{for every } g \in Z\left(\mathbf{H}(\mathbb{Z}_d)\right), \\ \left| \left\langle \psi, \pi_{\mathbf{H}(\mathbb{Z}_d)} \psi \right\rangle \right| &= \sqrt{\frac{1}{1+d}}, \quad \text{for every } g \notin Z\left(\mathbf{H}(\mathbb{Z}_d)\right). \end{aligned}$$

By altering the definition of Γ_C a bit, let

$$\hat{\Gamma}_C := \Bigg\{ \frac{1}{n} \left(I + \sum_{g \in Z(\mathbf{H}(\mathbb{Z}_d)) \backslash \{e\}} \theta_g L(g) + \sum_{g \in G \backslash Z(\mathbf{H}(\mathbb{Z}_d))} c_g L(g) \right) + Y$$
$$: \left|\theta_g\right| = 1, \forall g \in Z\left(\mathbf{H}(\mathbb{Z}_d)\right) \backslash \{e\}, \quad \left|c_g\right| \leq \sqrt{\frac{1}{d+1}}, \forall g \in G \backslash Z\left(\mathbf{H}(\mathbb{Z}_d)\right),$$
$$Y \in \mathfrak{N}(\mathbf{H}(\mathbb{Z}_d))^{\perp} \Bigg\}.$$

And Algorithm (3) is adapted to find the intersection between $\hat{\Gamma}_c$ and $\mathcal{R}\left(\Phi_{\pi_{\mathbf{H}(\mathbb{Z}_d)}}, \mathcal{S}_d\right)$. We initialize our algorithm with a random vector drew from a multivariate Gaussian distribution. To find the global optimum, the algorithm was restarted for many time. We illustrated the experimental results in Figure 4.1. The figure shows that when the dimension is smaller than 20, we have found cyclic vectors that lead to equiangular tight frames up to small numerical errors. However, when the dimension increases, the chance to find such vectors is reduced. Scott and Grassl (2010) has shown in numerical experiments that, at least for every dimension smaller than 51, there exists a fiducial vector giving equiangular tight frame in some eigenspace of a particular operator from the Clifford group (see Scott and Grassl (2010) for details). We found that, if our algorithm is initialized with a vector randomly selected in that eigenspace (which we refer as Zauner's subspace)g, the chance of finding ETFs becomes higher. See Figure 4.2 for the rate of success estimated from 50 restarts.

Figure 4.3 shows the convergence of our algorithm with and without extrapolation. It is demonstrative that the extrapolation heuristic work quite well in practice.

It worths noticing that the set of operators $\{M^n T^m : m, n \in \mathbb{Z}_d\}$ are mutually orthogonal. This can be verified by computing their pair-wise inner products directly. Thus $\{M^n T^m : m, n \in \mathbb{Z}_d\}$ in fact is an orthogonal basis of $\mathbb{C}^{d \times d}$. Then the problem of searching for an unit-norm cyclic vector becomes equivalent to searching for a rank-one Hermitian matrix $H \in \mathbb{C}^{d \times d}$ such that $|\langle H, M^n T^m \rangle| = (d+1)^{-1/2}$. Under this observation, let's denote

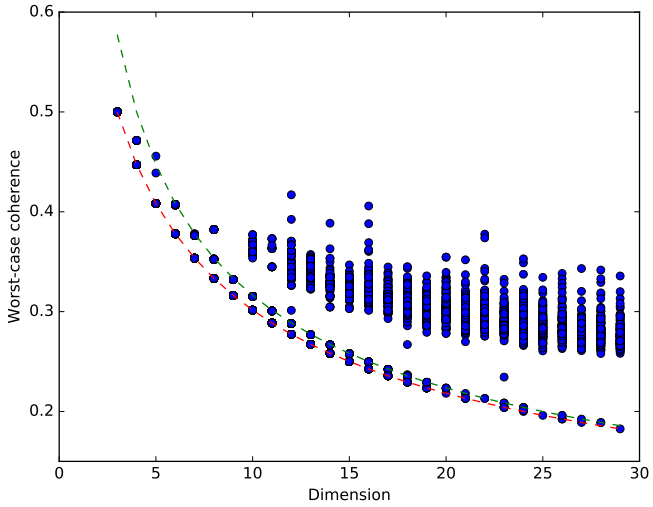


Figure 4.1: Worst-case coherence of frames generated by Heisenberg group. Blue circles: Worse-case coherence of frames generated from alternating projection algorithm. Red dash line: the Welch bound $\sqrt{1/(d+1)}$. Green dash line: the worset-case coherence of mutually unbiased bases $\sqrt{1/d}$.

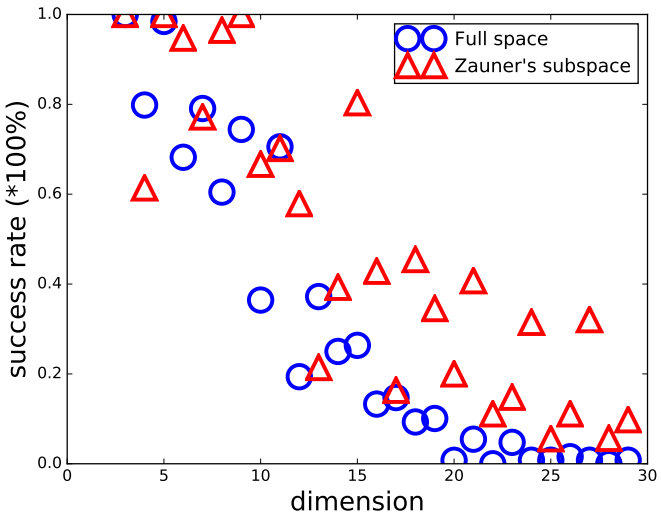


Figure 4.2: Success rate under different initializations. Blue circles: real and imaginary part of the initial vector are drawn from a Gaussian distribution over the full vector space. Red triangles: initial vector is drawn from the Zauner's subspace.

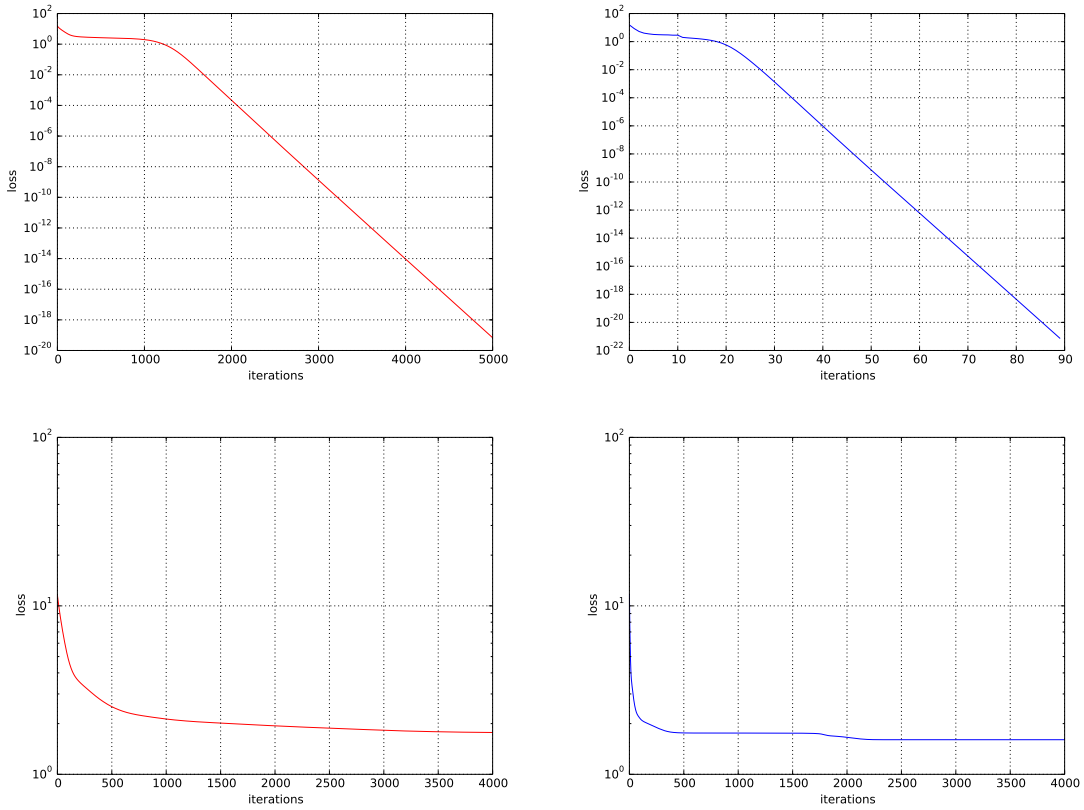


Figure 4.3: Convergence of the proposed algorithm in the case of $d = 17$. Left column: without acceleration. Right column: accelerated with the extrapolation scheme. Upper row: a global minimum is found. Lower row: converge towards a local minimum.

the set $(\mathbb{Z}_d \times \mathbb{Z}_d) \setminus \{(0,0)\}$ as $\mathcal{A}$. Consider the two sets

- $\tilde{\mathcal{R}} := \left\{ H \in \mathbb{C}^{d\times d} : \text{trace}(H) = 1, \text{rank}(H) = 1, H = H^* \right\}$.
- $\tilde{\Gamma} := \left\{ \frac{1}{d} \left(I + \sum_{(m,n)\in\mathcal{A}} c_{(m,n)} M^n T^m \right) : \left| c_{(m,n)} \right| \leq \sqrt{\frac{1}{d+1}}, \text{ for any } (m,n) \in \mathcal{A} \right\}$.

As an alternative to our general algorithm, one can set up a specific alternating projection method to find the intersection between $\tilde{\mathcal{R}}$ and $\tilde{\Gamma}$. As usual, projection onto $\tilde{\mathcal{R}}$ can be computed by using SVD. Projection onto $\tilde{\Gamma}$ can also be easily computed by shrinking the modulus of the coefficients $c_{(m,n)}$, due to the orthogonality of $M^n T^m$'s.

4.3.2 Finite affine group

Let $p > 2$ be a prime number and $\mathbb{F}_p$ be the finite field of order p. Let $\text{Aff}\left(\mathbb{F}_p\right)$ denote the finite affine group $\mathbb{F}_p \rtimes \mathbb{F}_p^\times$, where $\mathbb{F}_p^\times$ is a multiplicative group consisting of all non-zero elements from $\mathbb{F}_p$. The set underlying $\text{Aff}\left(\mathbb{F}_p\right)$ is $\left\{(b,a) : b \in \mathbb{F}_p,\, a \in \mathbb{F}_p^\times\right\}$ and the group operation is defined by

$$(b_1,\, a_1) \cdot (b_2,\, a_2) = (a_1 b_2 + b_1,\, a_1 a_2)\,, \quad \text{for all } b_1, b_2 \in \mathbb{F}_p,\, a_1, a_2 \in \mathbb{F}_p^\times.$$

Given $b \in \mathbb{F}_p$ and $a \in \mathbb{F}_p^\times$, let T_b and D_a be the "translation" and "dilation" operators on $\mathbb{C}\left(\mathbb{F}_q\right)$ defined by,

$$\left(T_b f\right)(t) = f(t-b),$$
$$\left(D_a f\right)(t) = f(a^{-1}t).$$

Let $\pi_{\text{Aff}(\mathbb{F}_p)}$ be a representation of $\text{Aff}\left(\mathbb{F}_p\right)$ on the space $\mathbb{C}\left(\mathbb{F}_p\right)$ defined by,

$$\pi_{\text{Aff}(\mathbb{F}_p)} : \text{Aff}\left(\mathbb{F}_p\right) \to \text{Aut}\left(\mathbb{C}\left(\mathbb{F}_q\right)\right), \tag{4.15}$$
$$(b,\, a) \mapsto T_b D_a.$$

Or, equivalently,

$$\left[\pi_{\mathrm{Aff}(\mathbb{F}_p)}(b,a)f\right](t) = f\left(a^{-1}(t-b)\right).$$

Note that the space of constant functions is invariant under π. $\pi_{\mathrm{Aff}(\mathbb{F}_p)}$ is a direct sum of two irreducible representations with dimension 1 (the trivial representation) and $p-1$ respectively. Similarly, due to $\mathbb{C}(\mathrm{Aff}\left(\mathbb{F}_p\right)) \cong \mathbb{C}^p$, we can write the operators T_b and D_a as matrices. Specifically, for every $b \in Fp$ and $a \in \mathbb{F}_p^\times$, $T_b, D_a \in \mathbb{C}^{p\times p}$. We index the rows and columns with field elements and the (m,n)-th entry of the matrices are

$$(T_b)_{m,n} = \delta_{m=n+b},$$
$$(D_a)_{m,n} = \delta_{m=an}.$$

Given this representation, we would like to search for a unit norm cyclic vector $\psi \in \mathbb{C}^p$ such that the group frame $\left\{T_b D_a \psi : b \in \mathbb{F}_p, a \in \mathbb{F}_p^\times\right\}$ is a tight frame with a low coherence.

The same as the case for Heisenberg group, the algorithm converges to some local minimum every time and we restarted the algorithm for several times. The results are shown in Figure 4.4. Empirically, frames generated from finite affine groups have worst-case coherence bounded a distance away from the Welch bound.

4.4 Discussion

We inspected the coherence of group frames in the space of operators associated with the group von Neumann algebra and constructed an alternating projection method to find a cyclic vector that leads to a frame with low coherence. The alternating projection can be carried out in d dimensional space and resembles a projected gradient descent method. As opposed to gradient descent optimization methods, our algorithm does not require a learning

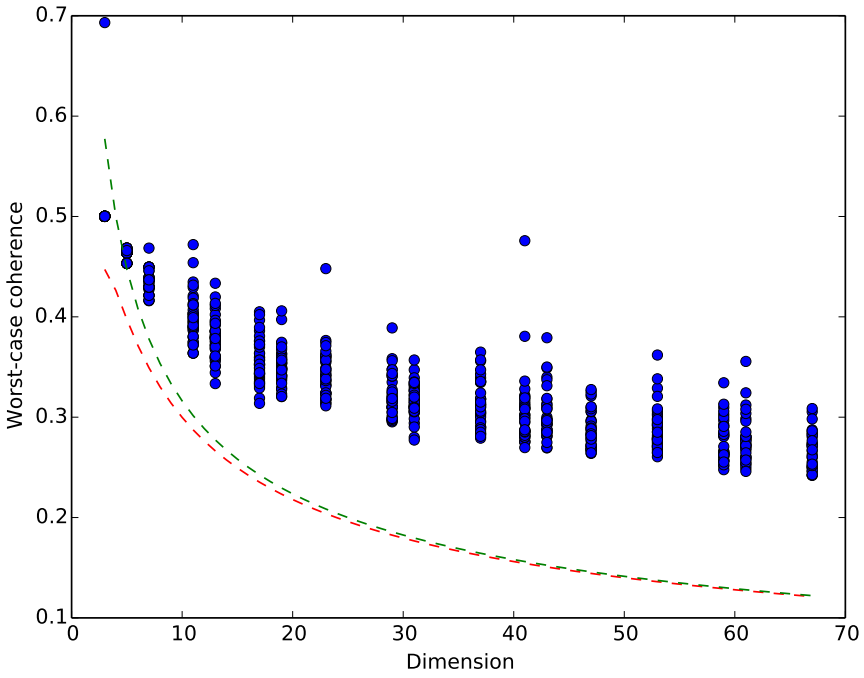


Figure 4.4: Worst-case coherence of frames generated by discrete affine group. Blue circles: Worse-case coherence of frames generated from alternating projection algorithm. Red dash line: the Welch bound $\sqrt{1/(d+1)}$. Green dash line: the worset-case coherence of mutually unbiased bases $\sqrt{1/d}$.

rate. The algorithm has a linear convergence rate locally. The extrapolation heuristic significantly accelerated the convergence of the algorithm in practice. But its theoretic analysis still need to be carried out.

The non-convexity of our formulation makes the algorithm trapped in local optimum quite often in high dimensions. To find an equiangular tight frame from Heisenberg group, the algorithm needs to be restarted for many times. Thus, it is critical that the algorithm is started near the global optimum. We tried starting our algorithm at the cyclic vector of Alltop sequence, since it generates a frame with coherence that is very close to Welch bound. However, it turns out to be a local optimum and the algorithm is not able to proceed further from there.

Compared with the alternating projection algorithm proposed by Tropp et al. (2005), our framework work with matrices of size $\mathbb{C}^{d\times d}$ instead of a spectral matrices of size $\mathbb{C}^{n\times n}$. But this comes with the constraint that the frame is imposed with a group structure which may prevent the frame to achieve the optimal coherence. From experiments, we can see that the choice of group sets affects the optimal coherence a group frame can achieve. The relation between the group structure and its influence on the coherence of group frames could be a meaningful future work.

4.5 Proof of local convergence of algorithm 1

To prove the local convergence of Algorithm 1, we resort to the work by Noll and Rondepierre (2015) and the idea from Luke (2013). The techniques in Noll and Rondepierre (2015) apply to real Euclidean spaces. Since we work in the complex vector spaces, we will use the isometry between $\mathbb{C}^{d\times d}$ and $\mathbb{R}^{2d^2}$ to adapt the tools from Noll and Rondepierre (2015). More specifically, let $\mathbb{C}^{d\times d}$ and $\mathbb{R}^{2d^2}$ be equipped with the standard inner product. Let $H = A + Bi \in \mathbb{C}^{d\times d}$ where $A, B \in \mathbb{R}^{d\times d}$ are its real and imaginary part. Then, H is equalized with $\alpha(H) := \left[\mathrm{vec}(A)^\top \mathrm{vec}(B)^\top\right]^\top \in \mathbb{R}^{2d^2}$. Let $\mathcal{E} \subset \mathbb{C}^{d\times d}$ be the set of all d-by-d

Hermitian matrices, i.e., $\mathcal{E} = \left\{A + Bi : A = A^\top \in \mathbb{R}^{d\times d}, B = -B^\top \in \mathbb{R}^{d\times d}\right\}$. Then, we can check that for any $H_1, H_2 \in \mathcal{E}$, $\langle H_1,\, H_2\rangle_{\mathbb{C}^{d\times d}} = \langle \alpha(H_1),\, \alpha(H_2)\rangle_{\mathbb{R}^{2d^2}} \in \mathbb{R}$. Note that the outcome of inner product between two matrices from $\mathcal{E}$ is always a real number. In the rest of this section, we will drop the subscript of such inner products. The following definitions are necessary to our proof.

Definition 4.5.1. *Let X be an Euclidean space. Given a non-empty closed subset A of X, the projection onto A is the set-valued mapping P_A associating with $x \in X$ the non-empty set*

$$P_A(x) = \{a \in A : \|x - a\| = d_A(x)\},$$

where $\|\cdot\|$ is the Euclidean norm, induced by the scalar product $\langle\cdot,\cdot\rangle$ and $d_A(x)$ is the distance from x to A,

$$d_A(x) = \min\{\|x - a\| : a \in A\}.$$

Meanwhile, for an $a \in A$, define $P_A^{-1}(a) := \{x \in X : P_A(x) = a\}$.

We will denote the sequence of alternating projections between non-empty closed sets A and B by this standard notation: $\ldots, a, b, a^+, b^+, a^{++}, b^{++}, \ldots$ with $b \in P_B(a), a^+ \in P_A(b), b^+ \in P_B(a^+)$ and so on. And $a \to b \to a^+$, respectively, $b \to a^+ \to b^+$ are referred as building blocks of the sequence. The analysis on alternating projection between Γ_C and $\mathcal{R}(\Phi_\pi)$ is based on such sequences. And, it is valid to assume that every member from these sequences are from $P_{\Gamma_C}(\mathcal{R}(\Phi_\pi))$ and $P_{\mathcal{R}}(\Phi_\pi)$. Each of such member is from $\mathcal{E}$. As we argued previously, inner products between two members from $\mathcal{E}$ is a real value. This allows us to adapt the following definition in Noll and Rondepierre (2015) which was originally proposed for real Euclidean spaces.

Definition 4.5.2. *(Separable intersection, Noll and Rondepierre (2015) Definition 1) Let A and B be two closed subsets of $\mathbb{R}^n$. We say that B intersects A separably at $\overline{x} \in A \cap B$ with*

exponent $\omega \in [0,2)$ *and constant* $\gamma > 0$ *if there exists a neighborhood* U *of* $\overline{x}$ *such that for every building block* $b \to a^+ \to b^+$ *in* U*, the condition*

$$\left\langle b - a^+,\, b^+ - a^+ \right\rangle \leq \left(1 - \gamma \|b^+ - a^+\|^\omega\right) \|b - a^+\| \|b^+ - a^+\| \tag{4.16}$$

is satisfied. We say that B *intersects* A *separably at* $\overline{x}$ *if* (4.16) *holds for some* $\omega \in [0,2)$*,* $\gamma > 0$*. If it is also true that* A *intersects* B *separably, that is, if the analogue of* (4.16) *holds for building blocks* $a \to b \to a^+$*, then we obtain a symmetric condition, and in that case, we say that* A*,* B *intersect separably at* $\overline{x}$*.*

Definition 4.5.3. *(Normal cones Bauschke et al. (2013) definition 2.1) Let* X *be an Euclidean space and* A*,*B *nonempty subsets of* X*. Let* $a \in A$ *and* $u \in X$*.*

- *The* B*-restricted proximal normal cone of* A *at* a *is*

$$\widehat{N}_A^B(a) := \operatorname{cone}\left((B \cap P_A^{-1}a) - a\right) = \operatorname{cone}\left((B - a) \cap (P_A^{-1}a - a)\right).$$

- *The* B*-restricted normal cone* $N_A^B(a)$ *is implicitly defined by* $u \in N_A^B(a)$ *if and only if there exist sequences* $(a_n)_{n\in\mathbb{N}}$ *in* A *and* $(u_n)_{n\in\mathbb{N}}$ *in* $\widehat{N}_A^B(a_n)$ *such that* $a_n \to a$ *and* $u_n \to u$*.*

Lemma 4.5.1. *(Restricted normal cone of* Γ_C*). Suppose* $X \in \Gamma_C$ *takes the form*

$$X = \frac{1}{n}\left(I + \sum_{g \in G\setminus\{e\}} c_g L(g)\right) + M,$$

for some $c_g \in \mathbb{C}$ *with* $\left|c_g\right| \leq C$ *and* $M \in \mathfrak{N}(\mathbf{H}(\mathbb{Z}_d))^\perp$*. Let* $\Delta := \left\{g \in G\setminus\{e\} : \left|c_g\right| = C\right\}$ *be the set of indices of* c_g *that has modulus strictly equal to* C*. And let* $\Delta^c = G\setminus\left\{\{e\} \cup \Delta\right\}$*.*

The $\mathcal{R}(\Phi_\pi)$-restricted normal cone of Γ_C is

$$N_{\Gamma_C}^{\mathcal{R}(\Phi_\pi)}(X) = \left\{ \frac{s}{n} \left((t-1)I + \sum_{g\in\Delta} \frac{c_g}{|c_g|}(r_g - C)L(g) \right) : s \geq 0, r_g \in \mathbb{R}, r_g \geq C, \forall g \in \Delta,\, t \in \mathbb{C} \right\}.$$

Proof. It is clear that for any $X \in \Gamma_C$,

$$P_{\Gamma_C}^{-1}(X) = \left\{ \frac{1}{n} \left(tI + \sum_{g\in\Delta^c} c_g L(g) + \sum_{g\in\Delta} \frac{c_g}{|c_g|} r_g L(g) \right) + M : r_g \in \mathbb{R}, r_g \geq C, \forall g \in \Delta,\, t \in \mathbb{C} \right\}.$$

□

Lemma 4.5.2. *(Restricted normal cone of $\mathcal{R}(\Phi_\pi)$). Suppose $Y \in \mathcal{R}(\Phi_\pi)$ takes the form*

$$Y = P_\pi^* \begin{bmatrix} \sigma_Y \psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi,$$

for some $\|\psi\|_2 = 1$ and $\sigma_Y > 0$. Any $X \in N_{\mathcal{R}(\Phi_\pi)}^{\Gamma_C}(Y)$ can be written as

$$X = sP_\pi^* \begin{bmatrix} \sum_{j=2}^d \sigma_j u_j u_j^* & \mathbf{Z} \\ \mathbf{Z}^* & \mathbf{D} \end{bmatrix} P_\pi$$

where $s \geq 0$, $\mathbf{D} \in \mathbb{C}^{(n-d)\times(n-d)}$, $\mathbf{Z} \in \mathbb{C}^{d\times(n-d)}$, σ_j's and u_j's are variables satisfying the following condition,

1. *$\sigma_Y \geq \sigma_2 \geq \cdots \geq \sigma_d \geq 0$, $j = 1, 2, \ldots, d$;*
2. *$u_j \in \mathbb{C}^d$, $u_j \perp \psi$, $j = 1, 2, \ldots, d$;*
3. *$\mathbf{D} = \mathbf{D}^*$;*
4. *$\mathrm{trace}(\mathbf{D}) + \sum_{j=2}^d \sigma_j = 1 - \sigma_Y$.*

Proof. It is clear that $P^{-1}_{\mathcal{R}(\Phi_\pi)}(Y)$ is

$$P^{-1}_{\mathcal{R}(\Phi_\pi)}(Y) = \left\{ P^*_\pi \begin{bmatrix} \sigma_Y \psi\psi^* + \sum_{j=2}^d \sigma_j u_j u_j^* & \mathbf{Z} \\ \mathbf{Z}^* & \mathbf{D} \end{bmatrix} P_\pi : \begin{array}{c} \mathbf{D}=\mathbf{D}^* \in \mathbb{C}^{(n-1)\times(n-d)}, \mathbf{Z} \in \mathbb{C}^{d\times(n-d)}, \\ \sigma_Y \geq \cdots \geq \sigma_d \geq 0, u_j \in \mathbb{C}^d, u_j \perp \psi, j=1,2,\ldots,d \end{array} \right\}$$

and therefore

$$P^{-1}_{\mathcal{R}(\Phi_\pi)}(Y) - Y = \left\{ P^*_\pi \begin{bmatrix} \sum_{j=2}^d \sigma_j u_j u_j^* & \mathbf{Z} \\ \mathbf{Z}^* & \mathbf{D} \end{bmatrix} P_\pi : \begin{array}{c} \mathbf{D}=\mathbf{D}^* \in \mathbb{C}^{(n-1)\times(n-d)}, \mathbf{Z} \in \mathbb{C}^{d\times(n-d)}, \\ \sigma_Y \geq \cdots \geq \sigma_d \geq 0, u_j \in \mathbb{C}^d, u_j \perp \psi, j=1,2,\ldots,d \end{array} \right\}.$$

Note that every member in this set has trace equal to $\text{trace}(\mathbf{D}) + \sum_{j=2}^d \sigma_j$. On the other hand, every member in Γ_C has trace equal to one. If $X \in \Gamma_C - Y$, then $\text{trace}(X) = 1 - \sigma_Y$. By definition, $\widehat{N}^{\Gamma_C}_{\mathcal{R}(\Phi_\pi)}(Y) = \text{cone}\left((\Gamma_C - Y) \cap (P^{-1}_{\mathcal{R}(\Phi_\pi)}(Y) - Y)\right)$. If $X \in \widehat{N}^{\Gamma_C}_{\mathcal{R}(\Phi_\pi)}(Y)$, it must be true that $\text{trace}(\mathbf{D}) + \sum_{j=2}^d \sigma_j = 1 - \sigma_Y$, which is condition (5).

□

Definition 4.5.4. *(CQ-condition Bauschke et al. (2013) definition 6.6) Let $c \in X$. Let $A, \tilde{A}, B$ and $\tilde{B}$ be nonempty subsets of X. Then the $(A, \tilde{A}, B, \tilde{B})$-CQ condition holds at c if*

$$N^{\tilde{B}}_A(c) \cap \left(-N^{\tilde{A}}_B(c)\right) \subseteq \{0\}.$$

Proposition 4.5.3. *(CQ implies 0-separability, Noll and Rondepierre (2015) proposition 1). Let $P_A(\partial B \backslash A) \subset \tilde{A}$, $P_B(\partial A \backslash B) \subset \tilde{B}$, and suppose $(A, \tilde{A}, B, \tilde{B})$ satisfies the CQ-condition at $\overline{x} \in A \cap B$. Then A, B intersect 0-separably at $\overline{x}$.*

Now we are ready to proceed in our proof. First, we show that Γ_C and the relaxed set $\mathcal{R}(\Phi_\pi)$ intersect separably, if the intersection exists.

Lemma 4.5.4. *Suppose that Γ_C and $\mathcal{R}(\Phi_\pi)$ intersect at $\overline{X} \in \mathbb{C}^{d\times d}$. $(\Gamma_C, \mathcal{R}(\Phi_\pi), \mathcal{R}(\Phi_\pi), \Gamma_C)$ satisfies the CQ-condition at $\overline{X}$. Further, Γ_C and $\mathcal{R}(\Phi_\pi)$ intersect 0-separably at $\overline{X}$.*

Proof. Suppose $\overline{X} \in \Gamma_C \cap \mathcal{R}(\Phi_\pi)$ takes the form

$$\overline{X} = \frac{1}{n}\left(I + \sum_{g \in G\setminus\{e\}} \overline{c_g} L(g)\right) + \overline{M} = P_\pi^* \begin{bmatrix} \overline{\psi}\,\overline{\psi}^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi.$$

for some $\overline{\psi} \in \mathbb{C}^d$ with $\|\overline{\psi}\|_2 = 1$, $\overline{c_g} \in \mathbb{C}$ with $\left|\overline{c_g}\right| \leq C$, $\overline{M} \in \mathfrak{N}(\mathbf{H}(\mathbb{Z}_d))^\perp$. Let $\overline{\Delta}$ and $\overline{\Delta}^c$ be defined in the same way as previous. Suppose $b \in N_{\mathcal{R}(\Phi_\pi)}^{\Gamma_C}$. By Lemma 4.5.2, it takes the form

$$b = s_1 P_\pi^* \begin{bmatrix} \sum_{j=2}^d \sigma_j u_j u_j^* & \mathbf{Z} \\ \mathbf{Z}^* & \mathbf{D} \end{bmatrix} P_\pi$$

with $u_j \perp \overline{\psi}$ and $\mathrm{trace}(b) = \mathrm{trace}(\mathbf{D}) + \sum_{j=2}^d \sigma_j = 0$. Then, it is easy to see that

$$\left\langle b, P_\pi^* \begin{bmatrix} \psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi \right\rangle = 0.$$

On the other hand, suppose $a \in N_{\Gamma_C}^{\mathcal{R}(\Phi_\pi)}$ takes the form

$$a = \frac{s_2}{n}\left((t-1)I + \sum_{g \in \overline{\Delta}} \frac{\overline{c_g}}{|\overline{c_g}|}(r_g - C)L(g)\right)$$

for some $r_g \geq C$. Now, assume $a = -b$. It is thus necessary to have

1. $\mathrm{trace}(a) = \mathrm{trace}(-b) = 0$.
2. $\left\langle a,\, P_\pi^* \begin{bmatrix} \psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi \right\rangle = -\left\langle b,\, P_\pi^* \begin{bmatrix} \psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi \right\rangle = 0.$

The condition (1) implies $t = 1$. To check condition (2), we observe that

$$\begin{aligned}\left\langle a,\, P_\pi^* \begin{bmatrix} \psi\psi^* & 0 \\ 0 & 0 \end{bmatrix} P_\pi \right\rangle &= \left\langle a,\, \frac{1}{n}\Big(I + \sum_{g\in G\setminus\{e\}} \overline{c_g} L(g) \Big) + \overline{M} \right\rangle \\ &= \frac{1}{n} \sum_{g\in\overline{\Delta}} \left|\overline{c_g}\right| \left(r_g - C\right) \\ &\geq 0.\end{aligned}$$

Equality holds if and only if $r_g = C$ for every $g \in \overline{\Delta}$ which is the case for $a = 0$. Therefore, $a = -b$ if and only if $a = b = 0$. This shows that $N^{\Gamma_C}_{\mathcal{R}(\Phi_\pi)}(\overline{X}) \cap \left(-N^{\mathcal{R}(\Phi_\pi)}_{\Gamma_C}(\overline{X})\right) \subseteq \{0\}$. Hence, the CQ-condition holds.

Further, given the CQ-condition and the fact that $P_{\Gamma_C}(\mathcal{R}(\Phi_\pi)) \subset \Gamma_C$ and $P_{\mathcal{R}(\Phi_\pi)}(\Gamma_C) \subset \mathcal{R}(\Phi_\pi)$, we conclude that Γ_C and $\mathcal{R}(\Phi_\pi)$ intersect 0-separably from Proposition 4.5.3. □

Finally, Theorem 2 from Noll and Rondepierre (2015) shows the local linear convergence.

Theorem 4.5.5. *(Noll and Rondepierre (2015), Theorem 2). Let A, B intersect 0-separably at x^* with constant $\gamma \in (0, 2)$. Suppose B is 0-Hölder regular at x^* with respect to A with constant $c < \gamma/2$. Then there exists a neighborhood V of x^* such that every sequence of alternating projections that enters V converges R-linearly to a point $b^* \in A \cap B$.*

We refer to Noll and Rondepierre (2015) for details on Hölder regularity and its properties (especially, Definition 2 and Corollary 3). In our case, we have shown the 0-seperability of Γ_C and $\mathcal{R}(\Phi_\pi)$. Meanwhile, since the set Γ_C is convex, it is prox-regular and is $\sigma-$Hölder regular with respect to $\mathcal{R}(\Phi_\pi)$ for every $\sigma \in [0, 1)$ with a constant $c > 0$ that may be chosen arbitrarily small. Thus, this theorem applies and concludes our proof on Theorem 4.2.7 in the main text.

References

Alltop, W. O. (1980). Complex sequences with low periodic correlations. *IEEE Transactions on Information Theory*, 26:350–354.

An, L. T. H. and Tao, P. D. (2005). The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Ann. Oper. Res.*, 133:23–46.

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. (2018). Understanding deep neural networks with rectified linear units. *International Conference on Learning Representations*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barbieri, D., Hernández, E., and Parcet, J. (2014). Riesz and frame systems generated by unitary actions of discrete groups. *Applied and Computational Harmonic Analysis*.

Bauschke, H. H., Luke, D. R., Phan, H. M., and Wang, X. (2013). Restricted normal cones and the method of alternating projections: theory. *Set-Valued and Variational Analysis*, 21(3):431–473.

Bengio, Y. and Delalleau, O. (2011). On the expressive power of deep architectures. In *International Conference on Algorithmic Learning Theory*, pages 18–36. Springer.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Bordes, A., Usunier, N., Chopra, S., and Weston, J. (2015). Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Bordes, A., Weston, J., Collobert, R., Bengio, Y., et al. (2011). Learning structured embeddings of knowledge bases. In *AAAI*, volume 6, page 6.

Bordes, A., Weston, J., and Usunier, N. (2014). Open question answering with weakly supervised embedding models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 165–180. Springer.

Boyd, S. and Vandenberghe, L. (1993). *Convex optimization*. Cambridge.

Candès, E. J., Plan, Y., et al. (2009). Near-ideal model selection by 1 minimization. *The Annals of Statistics*, 37(5A):2145–2177.

Casazza, P. G. and Kutyniok, G. (2013). *Finite Frames: Theory and Applications (Chapter 5 by S. Waldron)*. Birkhuser.

Cegielski, A. and Suchocka, A. (2008). Relaxed alternating projection methods. *SIAM Journal on Optimization*, 19(3):1093–1106.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Delalleau, O. and Bengio, Y. (2011). Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems*, pages 666–674.

Donoho, D. L., Elad, M., and Temlyakov, V. N. (2006). Stable recovery of sparse overcomplete representations in the presence of noise. *Information Theory, IEEE Transactions on*, 52(1):6–18.

Eldan, R. and Shamir, O. (2016). The power of depth for feedforward neural networks. In *Conference on Learning Theory*, pages 907–940.

Fickus, M. and Mixon, D. G. (2015). Tables of the existence of equiangular tight frames. *arXiv preprint arXiv:1504.00253*.

Fickus, M., Mixon, D. G., and Tremain, J. C. (2012). Steiner equiangular tight frames. *Linear algebra and its applications*, 436(5):1014–1027.

Gabardo, J.-P., Han, D., et al. (2003). Frame representations for group-like unitary operator systems. *Journal of Operator Theory*, 49(2):223–244.

Gearhart, W. B. and Koshy, M. (1989). Acceleration schemes for the method of alternating projections. *Journal of Computational and Applied Mathematics*, 26(3):235–249.

Graves, A. (2012). Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Greene, C. and Zaslavsky, T. (1983). On the interpretation of whitney numbers through arrangements of hyperplanes, zonotopes, non-radon partitions, and orientations of graphs. *Transactions of the American Mathematical Society*, 280(1):97–126.

Gritzmann, P. and Sturmfels, B. (1993). Minkowski addition of polytopes: computational complexity and applications to gröbner bases. *SIAM Journal on Discrete Mathematics*, 6(2):246–269.

Gröchenig, K. (2013). *Foundations of time-frequency analysis*. Springer Science & Business Media.

Guibas, L. J., Nguyen, A., and Zhang, L. (2003). Zonotopes as bounding volumes. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 803–812. Society for Industrial and Applied Mathematics.

Guu, K., Miller, J., and Liang, P. (2015). Traversing knowledge graphs in vector space. In *EMNLP 2015*.

Han, D. and Larson, D. R. (2000). *Frames, bases and group representations*, volume 697. American Mathematical Soc.

Hartman, P. et al. (1959). On functions representable as a difference of convex functions. *Pacific J. Math*, 9(3):707–713.

Hayashi, K. and Shimbo, M. (2017). On the equivalence of holographic and complex embeddings for link prediction. *arXiv preprint arXiv:1702.05563*.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Holtz, O. and Ron, A. (2011). Zonotopal algebra. *Advances in Mathematics*, 227(2):847–894.

Itenberg, I., Mikhalkin, G., and Shustin, E. I. (2009). *Tropical algebraic geometry*, volume 35. Springer Science & Business Media.

Joulin, A., Grave, E., Bojanowski, P., Nickel, M., and Mikolov, T. (2017). Fast linear model for knowledge graph embeddings. *arXiv preprint arXiv:1710.10881*.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *EMNLP*, volume 3, page 413.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. (2015). Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

Luke, D. R. (2013). Prox-regularity of rank constraint sets and implications for algorithms. *Journal of Mathematical Imaging and Vision*, 47(3):231–238.

Maclagan, D. and Sturmfels, B. (2015). *Introduction to tropical geometry*, volume 161. American Mathematical Soc.

McMullen, P. (1971). On zonotopes. *Transactions of the American Mathematical Society*, 159:91–109.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *hlt-Naacl*, volume 13, pages 746–751.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mixon, D. G., Bajwa, W. U., and Calderbank, R. (2011). Frame coherence and sparse signal processing. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 663–667. IEEE.

Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932.

Neelakantan, A., Roth, B., and McCallum, A. (2015a). Compositional vector space models for knowledge base completion. *arXiv preprint arXiv:1504.06662*.

Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. (2015b). Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.

Nickel, M., Rosasco, L., and Poggio, T. (2015). Holographic embeddings of knowledge graphs. *arXiv preprint arXiv:1510.04935*.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816.

Noll, D. and Rondepierre, A. (2015). On local convergence of the method of alternating projections. *Foundations of Computational Mathematics*, pages 1–31.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Renes, J. M., Blume-Kohout, R., Scott, A. J., and Caves, C. M. (2003). Symmetric informationally complete quantum measurements. *arXiv preprint quant-ph/0310075*.

Scott, A. J. and Grassl, M. (2010). Symmetric informationally complete positive-operator-valued measures: A new computer study. *Journal of Mathematical Physics*, 51(4):042203.

Socher, R., Chen, D., Manning, C. D., and Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934.

Strohmer, T. and Heath, R. W. (2003). Grassmannian frames with applications to coding and communication. *Applied and computational harmonic analysis*, 14(3):257–275.

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Sustik, M. A., Tropp, J. A., Dhillon, I. S., and Heath, R. W. (2007). On the existence of equiangular tight frames. *Linear Algebra and its applications*, 426(2):619–635.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Tarela, J. and Martinez, M. (1999). Region configurations for realizability of lattice piecewise-linear models. *Mathematical and Computer Modelling*, 30(11-12):17–27.

Thill, M. and Hassibi, B. (2015). Group frames with few distinct inner products and low coherence. *Signal Processing, IEEE Transactions on*, 63(19):5222–5237.

Thill, M., Muthukumar, V., and Hassibi, B. (2014). Frames from generalized group fourier transforms and sl 2 (f q). In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4923–4927. IEEE.

Thill, M. D. (2016). *Algebraic Techniques in Coding Theory: Entropy Vectors, Frames, and Constrained Coding*. PhD thesis, California Institute of Technology.

Tropp, J. A., Dhillon, I. S., Heath Jr, R. W., and Strohmer, T. (2005). Designing structured tight frames via an alternating projection method. *Information Theory, IEEE Transactions on*, 51(1):188–209.

Trouillon, T. and Nickel, M. (2017). Complex and holographic embeddings of knowledge graphs: a comparison. *arXiv preprint arXiv:1707.01475*.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080.

Tsiligianni, E. V., Kondi, L. P., and Katsaggelos, A. K. (2014). Construction of incoherent unit norm tight frames with application to compressed sensing. *Information Theory, IEEE Transactions on*, 60(4):2319–2330.

Turing, A. M. (1938). On computable numbers, with an application to the entscheidungsproblem: A correction. *Proceedings of the London Mathematical Society*, 2(1):544.

Vale, R. and Waldron, S. (2004). Tight frames and their symmetries. *Constructive approximation*, 21(1):83–112.

Vilnis, L. and McCallum, A. (2014). Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Waldron, S. (2009). On the construction of equiangular frames from graphs. *Linear Algebra and its Applications*, 431(11):2228–2242.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119. Citeseer.

Weston, J., Chopra, S., and Bordes, A. (2014). Memory networks. *arXiv preprint arXiv:1410.3916*.

Weston, J., Weiss, R. J., and Yee, H. (2013). Nonlinear latent factorization by embedding multiple user interests. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 65–68. ACM.

Williams, J. D., Kamal, E., Mokhtar Ashour, H. A., Miller, J., and Zweig, G. (2015). Fast and easy language understanding for dialog systems with microsoft language understanding intelligent service (LUIS). In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 159.

Xia, P., Zhou, S., and Giannakis, G. B. (2005). Achieving the welch bound with difference sets. *Information Theory, IEEE Transactions on*, 51(5):1900–1907.

Zauner, G. (1999). *Quantendesigns: Grundzüge einer nichtkommutativen Designtheorie*. na.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zorlein, H. and Bossert, M. (2015). Coherence optimization and best complex antipodal spherical codes. *Signal Processing, IEEE Transactions on*, 63(24):6606–6615.