Mathematical Background for Geometric Deep Learning

Haitz Sáez de Ocáriz Borde and Michael Bronstein

University of Oxford

We provide a succinct review of the key mathematical concepts required for studying Geometric Deep Learning. For a deeper understanding of specific topics, we encourage supplementing studies with additional resources.

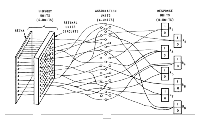
These notes were developed by Haitz Sáez de Ocáriz Borde for the ANAIS 2024 Geometric Deep Learning course in Kathmandu, Nepal, based on Michael Bronstein's notes for the Computer Vision and Pattern Recognition course offered at USI Lugano, Switzerland in 2019, and lecture slides for the Geometric Deep Learning course at the University of Oxford, United Kingdom in 2024.

Introduction

Since the dawn of civilization, humans have tried to understand the nature of intelligence. With the advent of computers, there have been attempts to emulate human intelligence using computer algorithms – a field that was dubbed 'Artificial Intelligence' or 'AI' by the computer scientist John McCarthy in 1956 and has recently enjoyed an explosion of popularity. Many efforts in AI research have focused on the study and replication of what is considered the hallmark of human cognition, such as playing intelligent games, the faculty of language, visual perception, and creativity. While at the time of writing we have multiple successful takes at the above – computers nowadays play chess and Go better than any human, can translate English into Chinese without a dictionary, automatically drive a car in a crowded city, and generate poetry and art that wins artistic competitions – it is fair to say that we still do not have a full understanding of what human-like or 'general' intelligence entails and how to replicate it.

Most of the aforementioned examples of AI are powered by Deep Learning, a class of algorithms whose history can be traced back to attempts in the early 20th century to replicate the connectivity and functioning of biological neurons in the brain in computers in a very abstract manner. Such systems are called neural networks, by analogy to their biological counterparts, and consist of computational units called neurons, which are typically organized into multiple layers (the term 'deep' in Deep Learning refers to neural networks with many such layers). Neurons have parameters that can be tuned for a specific task in an optimization procedure referred to as 'learning'. A subfield of AI studying mathematical methods for the design and optimization of such systems is called Machine Learning (ML).

The Perception, introduced by Frank Rosenblatt in 1957, is perhaps the simplest form of an artificial neural network, consisting of only a single artificial neuron. Modern neural networks can contain millions of neurons with billions of weights.



> *Deep Learning* is an umbrella term for Machine Learning algorithms that rely on artificial neural networks typically consisting of a large number of layers.

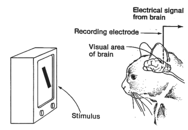
What is Geometric Deep Learning?

In recent years, there has been a rapid proliferation of various artificial neural network architectures, each suggesting different connectivity patterns and internal computations to be performed by the learning systems.

In the early 2020s there has been a clear convergence towards Transformer-based architectures across data modalities.

Geometric Deep Learning is a subfield of Deep Learning that focuses on developing artificial neural networks for data with non-Euclidean structures, such as graphs and manifolds. Traditional deep learning models, operate on grid-like data (e.g., images, time series, text), but many real-world problems involve more complex, irregular geometries. In particular, the field focuses on analyzing neural networks based on the geometric priors they leverage. Different models combat the curse of dimensionality by modeling signals on domains endowed with symmetry groups, which serve as inductive biases for the network.

Imposing inductive biases in learning systems becomes particularly important in data-scarce regimes. While modern Deep Learning is only loosely rooted in biological neural networks, some architectural choices, such as the inductive biases of Convolutional Neural Networks (CNNs), are directly inspired by the workings of the visual cortex.



This inspiration can be traced back to the experiments of Hubel and Wiesel.

> *Geometric Deep Learning* provides a structured approach to incorporating prior knowledge of physical symmetries into the design of new neural network architectures, while also unifying and understanding successful existing models under a common framework.

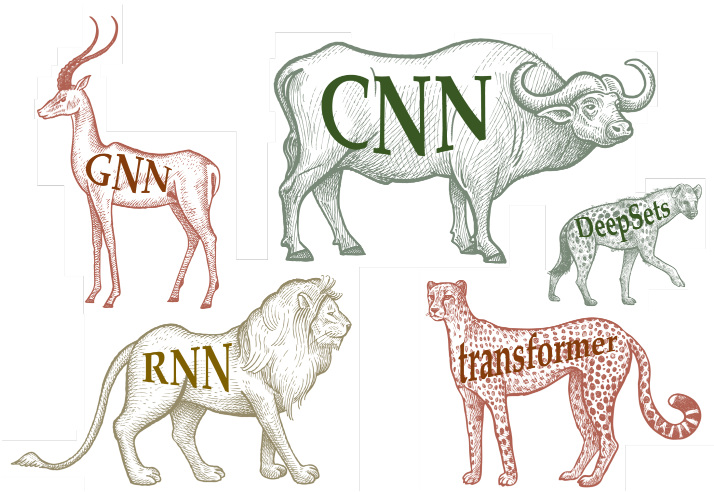
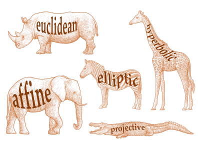


Figure 1: In the spirit of the Erlangen Program, Geometric Deep Learning provides a geometric unification of the zoo of Deep Learning architectures.

In this text, we will not focus directly on (Geometric) Deep Learning and artificial neural networks. Instead, our objective is to provide the necessary preliminary mathematical background often overlooked in standard computer science curricula.

In his research paper, which became a landmark in the history of mathematics as the Erlangen Program, the German mathematician Felix Klein proposed that geometry should be approached as the study of invariants or symmetries. This was in response to a historical context in which the study of different (non-Euclidean) geometries had become fragmented into separate fields.



Contents

1 Algebraic Structures and Mathematics before Numbers

In this section, we study pre-numerical structures that are fundamental to understanding Geometric Deep Learning. Structures such as sets and maps allow us to mathematically describe collections of objects, the connections between them, and the operations that can be performed on them. A key focus is on groups, which are used in Geometric Deep Learning to model the transformations of the data.

1.1 Sets

At first glance, numbers may appear as the most elementary objects in mathematics. However, it is possible to identify even simpler and more basic structures. Indeed, numbers can be added, subtracted, multiplied, and so on, which requires a set of rules defining how these operations are done. But what if we consider just a collection of objects, stripped off any additional assumptions about them?

> A *set* is a collection of distinct objects, called *elements* or *members* of the set.

The elements of a set are not restricted to being numbers; they could also be English words, for instance: $\{\text{cat}, \text{dog}\}$.

These elements can be anything: numbers, symbols, or even other sets. What characterizes a set is that it does not allow for a repetition of elements (i.e., every element appears only once in a set), and the order in which elements appear does not matter (i.e., sets are *unordered*). Set are the basis for defining more complex mathematical structures.

A *multiset* is a set in which elements are allowed to appear more than once. Multisets are common in Geometric Deep Learning in the context of graph neural networks, where they are used to model the neighborhood of a node in the graph.

They are typically denoted by capital letters, such as A, B, X, etc. The members of a set are listed inside curly braces $\{\}$, and if an element x belongs to a set A, we write $x \in A$, which reads as 'x is an element of A'. If x does not belong to A, we write $x \notin A$. For instance, if $A = \{1, 2, 3\}$, then $2 \in A$, but $4 \notin A$.

A non-example would be the collection of all sets: there is no set containing all sets.

Examples of Sets

- $\emptyset$: The empty set, a set with no elements. It is denoted by $\emptyset$ or sometimes by $\{\}$.
- *Singleton Set*: A set with exactly one element, for example, $\{1\}$.
- $\mathbb{N} = \{1, 2, 3, \ldots\}$: The set of natural numbers. The ellipsis $\ldots$ indicates that the set continues indefinitely with positive integers.
- $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$: The set of integers, which includes positive numbers, negative numbers, and zero.
- $\mathbb{Q} = \left\{\frac{p}{q} \mid p \in \mathbb{Z}, q \in \mathbb{N}\right\}$: The set of rational numbers, which are numbers that can be expressed as a ratio of two integers.
- $\mathbb{R}$: The set of all real numbers, including both rational numbers (e.g., $1, 0.75, -3$) and irrational numbers (e.g., π, $\sqrt{2}$).
- $\mathbb{C}$: The set of all complex numbers, which can be written as $a + bi$, where a and b are real numbers and i is the imaginary unit with $i^2 = -1$.

In some textbooks $\mathbb{N}$ may include 0.

The notation $\mathbb{Z}$ for integers comes from the German *Zahlen*, which means 'numbers'.

Set Notation and Operations

- *Set Builder Notation*: Set builder notation is used to describe a set by specifying an expression or the general form of an element, followed by a vertical bar separator $|$, and, to its right, a rule that the expression on the left must satisfy.

  $$\{x|\mathfrak{f}(x)\} = \{\text{expression}|\text{rule satisfied by the expression}\}.$$

  In words, it can be read as 'x such that (for which) $\mathfrak{f}(x)$'.

Sometimes a colon is used instead of a vertical line:

$$\{x : \mathfrak{f}(x)\}.$$

- *Subset*: A set A is a subset of a set B, written $A \subseteq B$, if every element of A is also an element of B. If $A \subseteq B$ but $A \neq B$, we say that A is a proper subset, written $A \subset B$.
- *Union*: The union of two sets A and B, written $A \cup B$, is the set of all elements that are in A, in B, or in both.
- *Intersection*: The intersection of two sets A and B, written $A \cap B$, is the set of all elements that are in both A and B.
- *Difference*: The difference of two sets A and B, written $A \setminus B$, is the set of all elements that are in A but not in B.
- *Complement*: The complement of a set A, written A^c, is the set of all elements not in A, assuming a universal set U that contains all elements under consideration.
- *Power Set*: The power set of a set A, denoted $\mathcal{P}(A)$, is the set of all subsets of A, including the empty set and A itself.
- *Cardinality*: The cardinality of a set is the size or number of elements it contains. If a set is finite, its cardinality is a non-negative integer. For infinite sets, cardinality is definited more abstractly: two infinite sets are said to have the same cardinality if there exists a bijection between their elements. The cardinality of a set A is denoted by $|A|$ or sometimes $\#(A)$.

The cardinality of $\mathbb{N}$ is denoted by the Hebrew letter $\aleph_0$, which reads as *aleph-nought* or *aleph-zero*. This is the 'smallest' type of infinity and represents the size of any countable infinite set, which is a set that can be placed in a one-to-one correspondence (bijection) with $\mathbb{N}$. For example, even though they might appear 'larger' at first glance, the sets $\mathbb{Z}$ and $\mathbb{Q}$ also have cardinality $\aleph_0$ since they are countably infinite.

Examples of Set Builder Notation We provide some examples to build an intuitive understanding. We start with the set builder notation. Below, we show that there are multiple ways to specify a set containing natural even numbers:

$$\{2x|x \in \mathbb{N}\} = \{x \in \mathbb{N}|x \text{ is even}\} = \{2, 4, 6, 8, ...\}.$$

Alternatively, sometimes the rule that must be satisfied by the elements of the set could be an equation:

$$\{x \in \mathbb{Z}|x > 0\} = \mathbb{N},$$

$$\{x \in \mathbb{Q}|x^2 = 0\} = \emptyset.$$

In the last example, the solutions to the equation $x^2 = 0$ are the roots $x = \pm\sqrt{2}$, which are irrational numbers and, therefore, not elements of $\mathbb{Q}$. Thus, the rule has no satisfying elements, meaning we have found a convoluted way of describing the empty set.

Examples of Finite Sets and Simple Operations Next, let us consider the finite sets $B = \{1, 2, 3, 4, 5\}$, $A = \{1, 2, 3\}$, $C = \{1, 2, 3, 4, 5\}$, then $C \subseteq B$ and $A \subset B$. This is because $A \neq B$, whereas $C = B$. Their cardinalities would be $|A| = 3$, $|B| = 5$, and $|C| = 5$. The unions and intersections in this example are $C \cup B = C \cap B = C = B$, $A \cup B = B$, and $A \cap B = A$. Another interesting example is the cardinality of the empty set $|\emptyset| = 0$ and the cardinality of the singleton set containing the empty set $|\{\emptyset\}| = 1$.

Examples of Infinite Sets and Simple Operations Consider the infinite sets $\mathbb{N} = \{1, 2, 3, 4, 5, \dots\}$ and $\mathbb{E} = \{2, 4, 6, 8, \dots\}$, the set of natural numbers and even natural numbers, respectively. Unsurprisingly, $\mathbb{E} \subset \mathbb{N}$ since every element of $\mathbb{E}$ is an element of $\mathbb{N}$. However, unlike finite sets, the cardinalities of $\mathbb{N}$ and $\mathbb{E}$ are *equal*, denoted as $|\mathbb{N}| = |\mathbb{E}| = \aleph_0$. This is due to the fact that there exists a *bijection* between $\mathbb{N}$ and $\mathbb{E}$ (we will explain bijections in more detail soon). One such bijection $f : \mathbb{N} \to \mathbb{E}$ can be defined as $f(n) = 2n$. For every natural number $n \in \mathbb{N}$, $f(n)$ produces a unique element of $\mathbb{E}$, and every element of $\mathbb{E}$ is hit exactly once. For example:

The Hilbert Hotel with infinitely many rooms that are fully occupied can host an infinite number of new guests by moving the old ones into even-numbered rooms and placing the new ones into odd-numbered rooms.

$$f(1) = 2,\ f(2) = 4,\ f(3) = 6,\ \dots$$

Note that despite $\mathbb{E}$ being a proper subset of $\mathbb{N}$, their infinite cardinality remains the same.

In terms of other operations,

$$\mathbb{E} \cup \mathbb{N} = \mathbb{N} \text{ and } \mathbb{E} \cap \mathbb{N} = \mathbb{E}.$$

Notably, the cardinality of the set containing, for instance, the infinite sets $\mathbb{R}$ and $\mathbb{N}$ is actually $|\{\mathbb{R}, \mathbb{N}\}| = 2$, since the set only contains two elements, despite the elements themselves being infinite.

Cartesian Products After introducing sets and some basic operations, let us define the Cartesian product. Although the concept may initially seem abstract, it plays an important role in discussing manifolds and constructing more complex spaces by combining elements from simpler subspaces. The Cartesian product is used to model composite systems and relations between elements of two or more sets.

In topology, the Cartesian product is used to define product spaces composed of two or more subspaces. For example, in the context of product manifolds, a torus can be represented as the Cartesian product of two circles.

The *Cartesian product* of two sets A and B, denoted by $A \times B$, is the set of all ordered pairs (a, b) where $a \in A$ and $b \in B$:

$$A \times B = \{(a, b) \mid a \in A, b \in B\}.$$

For instance, let $A = \{1, 2\}$ and $B = \{b_1, b_2\}$. Their product $A \times B$ is:

$$A \times B = \{(1, b_1), (1, b_2), (2, b_1), (2, b_2)\}.$$

We can also represent it as a table:

$A \times B$	b_1	b_2
1	$(1, b_1)$	$(1, b_2)$
2	$(2, b_1)$	$(2, b_2)$

Maps In many curricula, students are directly introduced to functions. However, before discussing functions, we can explore the more general concept of rules that define *mappings* between elements of different sets.

A *map* is a rule F which assigns to each element of a set A another element of a set B: $F(a) \equiv b \in B \,\forall a \in A$.

In the above expression, we read $\equiv$ as 'is defined as' or 'is equivalent to', indicating that $F(a)$ is explicitly assigned the value b in the set B. The symbol $\forall$ is read as 'for all', emphasizing that this rule applies to every element a in the set A.

It is common to use the following notation

$$F : A \to B.$$

We call A the *domain* and B the *codomain*, the element $a \in A$ fed into the map the *argument* (or *preimage*), and $F(a)$ its *image*. Note that we use different notations to distinguish between the function as a mapping between sets and its behavior on individual elements. For example:

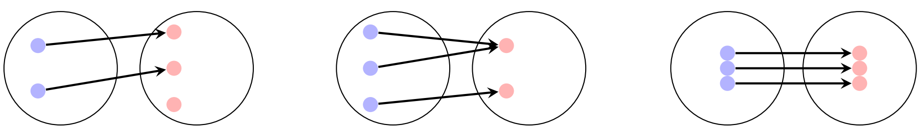
$$F : \mathbb{N} \to \mathbb{Z}, \quad x \mapsto F(x) = x^2,$$

where the expression on the left-hand side focuses on specifying the domain and codomain of F, whereas the right-hand side highlights the action of F on individual elements of the domain, that is, on particular inputs.

A *function* is a special type of mapping, which maps a set into the set of numbers.

Types of Maps Maps can be *surjective*, *injective*, or *bijective*, depending on how they map elements from one set to another. We say that a map between two sets is *bijective* when it is both *injective* and *surjective*.

The terms injection, surjection, and bijection were introduced by a group of French mathematicians publishing under the collective pseudonym Nicholas Bourbaki in 1954, and the adjective forms first used by Claude Chevalley in 1956.



(a) Injective (b) Surjective (c) Bijective

Figure 2: Depiction of injective, surjective, and bijective maps between two sets whose elements are highlighted in blue and red respectively.

Injective (One-to-One): A map $F : A \to B$ is called *injective* (or one-to-one) if different elements in the domain A map to different elements in the codomain B. That is, for all $a_1, a_2 \in A$,

$$F(a_1) = F(a_2) \implies a_1 = a_2.$$

Surjective (Onto): A map $F : A \to B$ is called *surjective* (or onto) if every element in the codomain B has at least one preimage in the domain A. That is, for every $b \in B$, there exists an $a \in A$ such that

$$F(a) = b.$$

Bijective: A map $F : A \to B$ is *bijective* if it is both injective and surjective. In other words, each element of A maps to a unique element of B, and every element of B has a unique preimage in A. A bijective map has an inverse, denoted $F^{-1} : B \to A$, such that

$$F^{-1}(F(a)) = a \quad \forall \quad a \in A, \quad F(F^{-1}(b)) = b \quad \forall \quad b \in B.$$

Composition Maps between different sets can be combined.

Given two maps, $F_1 : A \to B$, and $F_2 : B \to C$, the *composition* of F_1 and F_2, denoted as $F_2 \circ F_1$, is a new map:

$$F_2 \circ F_1 : A \to C$$

.

Note that when we compose injective maps, the result is also injective. Similarly, when we compose surjective maps or two bijective maps, the resulting maps are also surjective and bijective, respectively.

Like maps, functions can also be composed to create new functions. If $f : X \to Y$ and $g : Y \to Z$, their composition, denoted as $g \circ f$, is a function $g \circ f : X \to Z$ defined by:

Note that composition of functions is associative but not commutative.

$$(g \circ f)(x) = g(f(x)).$$

For example, let $f(x) = x^2$ and $g(x) = \sin(x)$. Then the composition $g \circ f$ is:

$$(g \circ f)(x) = g(f(x)) = \sin(f(x)) = \sin(x^2).$$

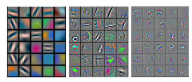
Similarly, the reverse composition $f \circ g$ is:

$$(f \circ g)(x) = f(g(x)) = f(\sin(x)) = (\sin(x))^2.$$

Function Composition and Deep Learning. Arguably, the foundation of Deep Learning lies in function composition, where the input undergoes iterative transformations through successive layers. Each layer processes the output (or activations) of the previous one, passing it as input to the next layer in the neural network. Also note that artificial neural networks are generally not bijective, as they are neither guaranteed to be injective nor surjective.

For instance, Figure 3 displays a schematic of a LeNet-5 neural network. We can observe how the input image is processed from left to right. The feature maps (yet another term for layer outputs or activations) are processed by different layers in the architecture and passed as input to the next layer to produce the subsequent set of feature maps. This is an example of function composition.

It is possible to visualize the internal filters learned by deep CNNs. The filters in the initial layers typically capture primitive patterns such as edges, corners, and textures, while the filters in deeper layers learn to compose these primitives into more complex features.



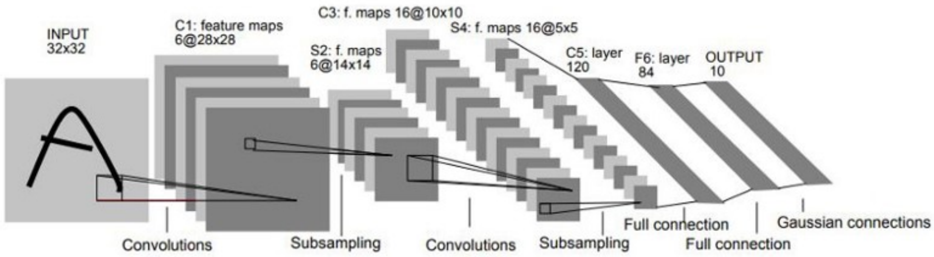


Figure 3: LeNet-5 classical CNN architecture.

1.2 Groups

A group is a way of organizing and understanding how a set of elements interact with one another through a well-defined operation. Groups are used to describe symmetry, structure, and transformations in various mathematical and physical contexts.

Many classes of physical operations can be associated with a group structure. Since Geometric Deep Learning architectures often aim to model such phenomena, groups become essential for designing artificial neural networks whose internal representations align with physical principles.

Let us consider a physical example before diving into the formal definition. Think of a square and the *group of rotations of the square*. The set of elements in this group consists of the different rotations $C_4 = \{0°, 90°, 180°, 270°\}$ that can be applied to the square. The operation here is combining rotations. For instance, applying two $90°$ rotations is equivalent to a single $180°$ rotation. Applying a $0°$ rotation followed by a $90°$ rotation results in just a $90°$ rotation. This shows that combining elements of the set results in elements within the same set.

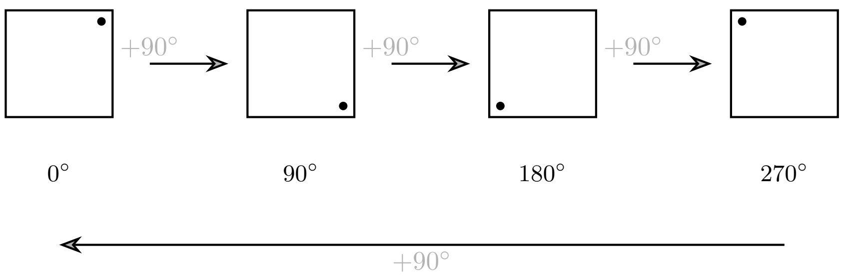


Figure 4: Rotational Symmetries of a Square (C_4).

This situation exemplifies symmetry: the square remains unchanged (*invariant*) under these rotations. In mathematics, symmetry refers to a property of an object or system that remains unchanged under specific transformations or operations.

The term *symmetry* has Greek origins 'symmetria' literally translates to 'same measure'.

Similar schematics can be created, for instance, to represent the symmetry of a triangle under both rotations and reflections. More generally, we refer to these as Cayley graphs

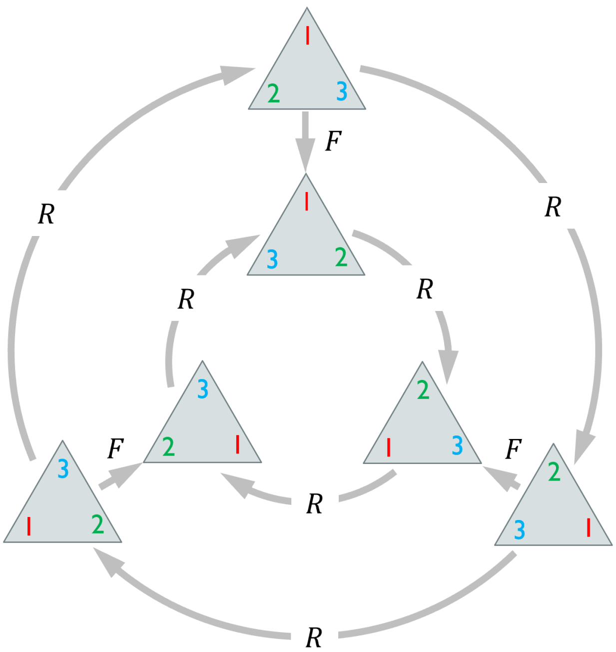


Figure 5: Cayley graph representing the symmetry of a triangle.

A *group* is a set equipped with a binary operation that combines any two elements of the set to form a third element. In a group, the set and the operation can be denoted as $(G, \circ)$, where G is the set and $\circ$ is the binary operation. The operation must satisfy the following fundamental properties, known as the group axioms:

- *Associativity*: For all $a, b, c \in G$, we have $(a \circ b) \circ c = a \circ (b \circ c)$.
- *Identity Element*: There exists an element $e \in G$ such that for all $a \in G$, $e \circ a = a \circ e = a$. This element is called the identity element.
- *Inverse Element*: For each element $a \in G$, there exists an element $b \in G$ such that $a \circ b = b \circ a = e$, where e is the identity element. The element b is called the inverse of a and is denoted a^{-1}.

$a \circ b$ can be denoted by juxtaposition for brevity: $a \circ b = ab$. Also, alternatively, one can use the symbol $*$.

Closure follows from the definition: for all $a, b \in G$, the result of the operation $c = a \circ b$ is also in G, $c \in G$, and commutativity does not necessarily apply in general. Groups can be finite, infinite, discrete, or continuous.

Examples of Groups

- *Integers under Addition*: The set of integers $\mathbb{Z}$ with the operation of addition $(+)$ forms a group. The identity element is 0, and each integer a has an additive inverse $-a$.

- *Non-zero Rational Numbers under Multiplication*: The set of non-zero rational numbers $\mathbb{Q}^* = \mathbb{Q} \setminus \{0\}$ with multiplication $(\cdot)$ forms a group. The identity element is 1, and each element a has a multiplicative inverse $\frac{1}{a}$.
- *Symmetric Group*: The symmetric group S_N consists of all permutations of N elements. The group operation is the composition of permutations, and it is an example of a finite group.

More on Groups

- *Abelian Group*: A group $(G, \circ)$ is called abelian (or commutative) if the operation is commutative, meaning $a \circ b = b \circ a$ for all $a, b \in G$.
- *Subgroup*: A subgroup H of a group G is a subset of G that is itself a group under the operation of G. If H is a subgroup of G, we write $H \leq G$.
- *Order of a Group*: The order of a group is the number of elements in the group.

A non-abelian group contains at least some elements for which $a \circ b \neq b \circ a$.

For instance, in our previous example, the group of rotations of a square, C_4, is abelian and has an order of 4. The group of rotations $C_2 = \{0°, 180°\}$ is a subgroup

$$C_2 \leq C_4.$$

Another important abelian group is that formed by all rotations of three-dimensional space.

Groups and Understanding Data Distributions through the Lens of Geometric Deep Learning. In Geometric Deep Learning, groups formalize the concept of *symmetry* in data. For instance, in computer vision, the group of translations ensures that object categories remain invariant when their positions shift, a property essential for tasks like visual object classification. In computational chemistry, predicting molecular properties requires outputs invariant to both rotations and translations, achieved through the Euclidean group $E(3)$. Similarly, for systems with discrete symmetries, such as permutations in graphs, the symmetric group S_n plays a central role. This group underpins transformations where elements (e.g., particles or nodes) can be arbitrarily reordered, a key aspect in Graph Neural Networks and the message-passing framework.

Graph Neural Networks are a type of artificial neural networks designed to process signals over graph structures.

Group Homomorphisms It is often that we may find groups which are equivalent, or that can be realized in different ways. The essence of a *group homomorphism* lies in preserving structure, rather than focusing solely on particular examples.

A *group homomorphism* is a map between two groups that preserves the group structure. Let $(G, \circ)$ and $(H, *)$ be two groups. A map $\phi : G \to H$ is called a *group homomorphism* if, $\forall a, b \in G$, the following condition holds:

$$\phi(a \circ b) = \phi(a) * \phi(b).$$

A *group isomorphism* is a bijective homomorphism between two groups G and H, establishing a perfect identification between them.

> Two groups $(G, \circ)$ and $(H, *)$ are said to be *isomorphic*, $(G, \circ) \cong (H, *)$, if there exists a bijective map (a one-to-one and onto mapping) $\phi : G \to H$ such that ϕ is a group homomorphism.

The group $C_4 = \{0°, 90°, 180°, 270°\}$ is the group of rotations of a square, where the group operation is addition modulo 360°. Let the group $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ be the group of integers under addition modulo 4. These groups are isomorphic, and the isomorphism can be described by a homomorphism.

The modulo operation (denoted as $a \mod n$) finds the remainder when a is divided by n. Specifically, $a \mod n$ is the integer remainder r such that $0 \leq r < n$ and $a = n \cdot q + r$ for some integer q.

Define the homomorphism $\phi : C_4 \to \mathbb{Z}_4$ as

$$\phi(0°) = 0, \quad \phi(90°) = 1, \quad \phi(180°) = 2, \quad \phi(270°) = 3.$$

This mapping respects the group operation. Let us verify the homomorphism property. The group operation in C_4 is addition modulo 360°, and the group operation in $\mathbb{Z}_4$ is addition modulo 4. To verify ϕ is a homomorphism, check that:

$$\phi(a + b \mod 360°) = \phi(a) + \phi(b) \mod 4, \quad \forall a, b \in C_4.$$

Some examples include

$$\phi(90° + 180° \mod 360°) = \phi(270°) = 3,$$

$$\phi(90°) + \phi(180°) \mod 4 = 1 + 2 \mod 4 = 3.$$

Next, let us illustrate a non-isomorphic mapping between C_4 and $C_2 = \{0°, 180°\}$. While both C_4 and C_2 are cyclic groups, their structures are fundamentally different, and no isomorphism exists between them. However, there are still homomorphisms that preserve the group structure.

Let $C_2 = \{0°, 180°\}$ where the group operation is addition modulo 360°. Define a homomorphism $\psi : C_4 \to C_2$ as:

$$\psi(0°) = 0°, \quad \psi(90°) = 180°, \quad \psi(180°) = 0°, \quad \psi(270°) = 180°.$$

This map is not injective (and therefore not bijective), which means that C_4 and C_2 are not isomorphic. Let us verify the homomorphism property. The group operation in both C_4 and C_2 is addition modulo 360°. To check that ψ is a homomorphism, we must verify:

$$\psi(a + b \mod 360°) = \psi(a) + \psi(b) \mod 360°, \quad \forall a, b \in C_4.$$

Some examples include: let $a = 90°$ and $b = 180°$

$$\psi(90° + 180° \mod 360°) = \psi(270°) = 180°,$$

$$\psi(90°) + \psi(180°) \mod 360° = 180° + 0° \mod 360° = 180°.$$

Group Actions A *group action* is a formal way of describing how a group interacts with a set while preserving its structure. It connects abstract group theory to concrete situations where groups *act* on mathematical or physical objects, such as transforming geometric shapes, permuting elements, or applying symmetry operations.

Let us revisit $C_4 = \{0°, 90°, 180°, 270°\}$ once more. These rotations act on the set of vertices of the square,

$$V = \{\hat{A}, \hat{B}, \hat{C}, \hat{D}\},$$

by permuting their positions. For example:

- A 90° rotation maps $\hat{A} \to \hat{B}$, $\hat{B} \to \hat{C}$, $\hat{C} \to \hat{D}$, $\hat{D} \to \hat{A}$.
- A 180° rotation maps $\hat{A} \to \hat{C}$, $\hat{B} \to \hat{D}$, $\hat{C} \to \hat{A}$, $\hat{D} \to \hat{B}$.

This interaction satisfies the structure-preserving properties of a group action.

A *(left) group action* of a group G on a set X is a mapping:

$$\alpha : G \times X \to X, \quad (g, x) \mapsto \alpha(g, x) = g \cdot x,$$

satisfying the following axioms:

- *Identity*: The identity element $e \in G$ acts as the identity transformation on X:
$$\alpha(e, x) = e \cdot x = x, \quad \forall x \in X.$$
- *Compatibility*: $\forall g, a \in G$ and $x \in X$, the action satisfies:
$$(g \circ a) \cdot x = g \cdot (a \cdot x),$$
where $\circ$ is the group operation in G.

The group operation vanishes on the right-hand side of the compatibility axiom because it is implicitly handled by the action itself. The key idea is that group actions are associative with respect to the group operation. This means that applying the action of $a \circ b$ to x is the same as first applying b to x and then applying a to the result.

Groups Actions on Data. In Geometric Deep Learning, rather than considering groups as abstract entities, we focus on how different mathematical operations, which we can prescribe for our artificial neural network, transform the input data. This enables us to design our model to perform transformations on the data that respect the structure of its domain.

In Geometric Deep Learning, we assume there is a domain underlying our data, which we denote by Ω, and study how groups act on Ω and how we obtain actions on the same group on the space of signals $\mathcal{X}(\Omega)$.

Figure 6: Group action on an image (function). The type of an object can be defined by the way it is transformed by a group.

Fields Before moving on to discussing vector spaces, let us briefly mention fields. Fields, groups, and vector spaces are interconnected in the hierarchy of algebraic structures. A group has a single binary operation with minimal axioms, while a field has two operations with stringent compatibility conditions. Hence, fields impose more structure than groups and belong to a different class of algebraic objects.

A *field* is a set $\mathbb{F}$ equipped with two binary operations, addition $(+)$ and multiplication $(\cdot)$, satisfying the following properties:

- $(\mathbb{F}, +)$ forms an abelian group (with identity element 0).
- $(\mathbb{F} \setminus \{0\}, \cdot)$ forms an abelian group (with identity element 1).
- Multiplication is distributive over addition: $\forall a, b, c \in \mathbb{F},\ a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

1.3 Vector Spaces

Now that we have a basic understanding of groups and fields, we can introduce the concept of a *vector space*. Vectors are ubiquitous in applications of mathematics, especially in physical sciences. Introductory courses often talk about vectors in geometric terms ('arrows that have direction and length') or computer science terms ('arrays of numbers'). Each of these definitions are a crime against humanity: in order to think of vectors as 'arrows', one has to define direction and length by introducing additional structures called inner products and norms; in order to think of vectors as 'arrays', one has to define a basis, with respect to which vectors can be represented as ordered sets of coordinates. The correct mathematical way of thinking of vectors is as abstract objects that can be scaled and added.

V is a *vector space* over a field $\mathbb{F}$ (typically $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$) with binary operations $+ : V \times V \to V$ (*vector addition*) and $\cdot : V \times \mathbb{F} \to V$ (*scalar multiplication*) if for any $u, v, w \in V$ and $\alpha, \beta \in \mathbb{F}$ we have the following properties:

- *Associativity* of $+$: $u + (v + w) = (u + v) + w$
- *Commutativity* of $+ :$ $u + v = v + u$
- *Identity element* of $+$: There exists a unique $0 \in V$ such that $u + 0 = u$
- *Inverse element* of $+$: There exists a unique $-v \in V$ such that $v + (-v) = 0$
- *Distributivity* of $\cdot$ w.r.t. vector addition : $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$
- *Distributivity* of $\cdot$ w.r.t. scalar addition: $(\alpha + \beta) \cdot v = \alpha \cdot v + \beta \cdot v$
- *Compatibility* of $\cdot$ with scalar multiplication: $\alpha \cdot (\beta \cdot v) = (\alpha \cdot \beta) \cdot v$
- *Identity element* of $\cdot$: $\exists! 1 \in \mathbb{R} \quad \text{s.t.} \quad 1 \cdot u = u$

A scalar is a single numerical value, such as a real number, with no direction or dimension. A vector is an ordered array of numbers, representing a point or direction in space, and can be one-dimensional or multi-dimensional. A tensor is a generalization of scalars and vectors to higher dimensions, represented as multi-dimensional arrays. For instance, scalars are 0th-order tensors, vectors are 1st-order tensors, and matrices are 2nd-order tensors. Higher-order tensors extend this concept, representing data with more than two dimensions, such as a sequence of matrices. In Deep Learning we tend to work with high-dimensional tensors.

Note that notation sometimes can be confusing and therefore should be used with care. The same notation is used for scalar addition $\alpha + \beta$ and vector addition $u + v$. It should be understood from context which addition is meant. The same notation is also used for scalar-by-scalar multiplication $\alpha \cdot \beta$ and vector-by-scalar multiplication $\alpha \cdot u$. When no confusion arises, the vector-by-scalar multiplication is often denoted as αu for brevity. The zero vector $0 \in V$ (identity element of vector addition) should not be confused with the zero scalar $0 \in \mathbb{R}$ (identity element of scalar addition), even though they are often denoted in the same way. Lastly, $\exists! u \in V$ means 'there exists a unique u in V', and it implies that there is exactly one element $u \in V$ such that a particular condition is satisfied.

Examples of Vector Spaces

- *Vectors*: $\mathbb{R}^n = \{(v_1, \ldots, v_n) : v_i \in \mathbb{R}, \forall i = 1, \ldots, n\}$ with $u + v = (u_1 + v_1, \ldots, u_n + v_n)$
- *Functions*: $\mathcal{F}(\Omega) = \{f : \Omega \to \mathbb{R}\}$ with $(f + g)(x) = f(x) + g(x)$

$\mathcal{F}$ is used to denote a set of functions on the domain Ω. That is, the set $\mathcal{F}(\Omega)$ consists of functions whose domain is Ω. Here, we talk about functions instead of maps, since we are considering special types of maps that map a set Ω to $\mathbb{R}$, rather than to an arbitrary set.

From Vector Spaces to Tensor Spaces Although it is less commonly discussed in basic linear algebra than vector spaces, in practice in Deep Learning we work with *tensor spaces*. A *tensor* is a multi-dimensional generalization of vectors and matrices. Tensors are particularly relevant in Deep Learning for parallel data processing.

Next, we discuss some basic examples. A *scalar* is a tensor of order (or rank) 0, represented by a single value, say

$$a = 5$$

A *vector* is a tensor of order 1, represented as a one-dimensional array

$$v = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

A *matrix* is a tensor of order 2, represented as a two-dimensional array

$$M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}.$$

A higher-order tensor, of order 3 in this example, is a n-dimensional array, represented as

$$T_{ijk} = \left[\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}, \begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix} \right].$$

T_{ijk} in this particular instance represents a $3 \times 2 \times 2$ tensor. Alternatively, we can express the slices more clearly:

$$T_{ijk} = \begin{cases} T_{1,:,:} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \\ T_{2,:,:} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}, \\ T_{3,:,:} = \begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix}. \end{cases}$$

The *Einstein summation convention* is a shorthand for tensor expressions, where repeated indices imply summation over all their possible values. This convention makes it easier to work with high-dimensional tensors. Let us look at some examples of Einstein summation.

The dot product of two vectors u and v in Einstein notation is written as

$$u_i v^i = \sum_i u_i v^i = a,$$

where we can omit the summation symbol, and the product results in a scalar, a.

For a matrix M and vector v, the matrix-vector multiplication in Einstein notation is

$$M_{ij} v^j = \sum_j M_{ij} v^j = u_i,$$

which results in another vector, u.

Likewise, a tensor contraction, which is a generalization of matrix multiplication, can be written as:

$$T_{ijk} v^j = \sum_j T_{ijk} v^j = M_{ik}.$$

This involves summing over the index j, since it is the repeated index. Multiplying an order 3 tensor and a vector, results in an order 2 tensor, that is, a matrix.

Tensor Spaces in Deep Learning. A *tensor space* can be thought of as a generalization of vector spaces to higher-dimensional objects, where tensors (multi-dimensional arrays) act as elements in these spaces. More formally, a tensor space can be described as a set of tensors where tensor addition and scalar multiplication follow the usual rules that hold for vector spaces, but are generalized to multi-dimensional arrays. For instance in Computer Vision, we typically process tensors of shape `[B,C,H,W]`, where `B` stands for batch size, `C` for channel dimension (RGB channels), and `H` and `W` are the height and width of the image. In the context of video we can further include a frames (or time) dimension and the tensor gains an additional dimension, `[B,C,F,H,W]`. However, oftentimes in research articles, transformations are represented in terms of matrices, and additional entries such as those for the batch dimension are omitted for clarity.

2 Geometric and Analytical Structures

Geometric structures bring life to abstract mathematical objects by introducing familiar concepts like distance, size, and angles. While groups and vector spaces give us powerful ways to study relationships and transformations, they lack the geometric intuition we often need in real-world applications.

2.1 Norms and Normed Vector Spaces

A norm is a mathematical function that quantifies the size or magnitude of a mathematical object, generalizing our intuitive understanding of length or distance in physical space. Like physical length, a norm assigns a non-negative real number to an object while satisfying specific properties.

Given a vector space V over a field $\mathbb{F} = \mathbb{R}$, a *norm* is a function $\| \ \| : V \to \mathbb{R}$ satisfying for any $u, v \in V$ and $\alpha \in \mathbb{R}$:

- *Positive homogeneity:* $\|\alpha u\| = |\alpha| \|u\|$
- *Triangle inequality:* $\|u + v\| \leq \|u\| + \|v\|$
- *Positive definiteness:* $\|u\| = 0 \quad \Rightarrow \quad u = 0$

$(V, \| \ \|)$ is called a *normed (vector) space*. Intuitively, the norm measures the length of a vector.

The field can also be $\mathbb{F} = \mathbb{C}$, but in the main text we stick to $\mathbb{R}$ for simplicity.

The following properties (often listed as part of axiomatic definition of the norm) are in fact consequences of the above definition:

- $\|0\| = \|0 \cdot u\| \overset{(1)}{=} |0| \|u\| = 0$, i.e. property (3) is iff: $\|u\| = 0 \Leftrightarrow u = 0$.
- $\|u\| \geq 0$,

where (1) refers to positive homogeneity and (3) to positive definiteness.

The notation $\|u\|$ refers to the norm of an element in a vector space, where u is a vector. In contrast, $|\alpha|$ denotes the absolute value of a scalar, which is a special case of a norm when the underlying field is the real or complex numbers. While the norm generalizes the concept of absolute value to vector spaces, the absolute value is specifically used for scalars.

Examples of Norms

- L_p*-norm on* $\mathbb{R}^n$*:* $\|u\|_p = \left(\sum_{i=1}^n |u_i|^p\right)^{1/p}$, in particular
  - L_1*-norm:* $\|u\|_1 = \sum_{i=1}^n |u_i|$
  - L_2*-norm (Euclidean norm):* $\|u\|_2 = \sqrt{\sum_{i=1}^n |u_i|^2}$
  - L_∞*-norm:* $\|u\|_\infty = \max\{|u_1|, \ldots, |u_n|\}$
- L_p*-norm on* $\mathcal{F}(\Omega)$*:* $\|f\|_p = \left(\int_\Omega |f(x)|^p dx\right)^{1/p}$

The L_2-norm, also known as the Euclidean norm, is the most commonly used norm, and it provides the notion of the length of a vector.

The summation in the vector case, is replaced by an integral in the function case. This is because functions can be thought of as vectors with infinitely many components, where the integral serves as a continuous analog of the sum.

2.2 Metrics Induced by Norms and Metric Spaces

A metric represents a mathematical way to measure distances between elements in a set, with norms being a special case that can generate metrics.

Given a normed vector space $(V, \|\cdot\|)$, a *metric* $d : V \times V \to \mathbb{R}$ is naturally defined by:

$$d(u, v) = \|u - v\|, \quad \forall u, v \in V.$$

This metric satisfies the following properties, making (V, d) a *metric space*:

- *Non-negativity:* $d(u, v) \geq 0$
- *Identity of indiscernibles:* $d(u, v) = 0 \Leftrightarrow u = v$
- *Symmetry:* $d(u, v) = d(v, u)$
- *Triangle inequality:* $d(u, w) \leq d(u, v) + d(v, w)$

Note that every normed vector space is also a metric space with a metric induced by its norm. However, not all metric spaces are normed vector spaces.

A metric measures the distance between two elements in a space, generalizing our intuitive notion of distance in physical space. Unlike norms which measure the size of a single vector, metrics quantify the separation between pairs of elements.

While normed vector spaces are inherently metric spaces, not all metric spaces have the additional algebraic structure of a vector space. A vector space requires operations like vector addition and scalar multiplication that satisfy specific axioms. Many metric spaces lack these operations or do not satisfy the vector space axioms. For instance, in $\mathbb{R}^n$, the metric $d(u, v) = \sqrt{|u_1 - v_1|} + \cdots + \sqrt{|u_n - v_n|}$ is a valid metric but cannot be derived from a norm.

Examples of Metrics Induced by Norms

- L_p *distance in* $\mathbb{R}^n$*:* $d_p(u, v) = \|u - v\|_p = \left(\sum_{i=1}^n |u_i - v_i|^p\right)^{1/p}$, in particular
  - L_1 *distance:* $d_1(u, v) = \|u - v\|_1 = \sum_{i=1}^n |u_i - v_i|$
  - L_2 *distance (Euclidean distance):* $d_2(u, v) = \|u - v\|_2 = \sqrt{\sum_{i=1}^n |u_i - v_i|^2}$
  - L_∞ *distance:* $d_\infty(u, v) = \|u - v\|_\infty = \max\{|u_1 - v_1|, \ldots, |u_n - v_n|\}$
- L_p *distance for functions:* $d_p(f, g) = \|f - g\|_p = \left(\int_\Omega |f(x) - g(x)|^p dx\right)^{1/p}$

The Euclidean distance is the most intuitive metric, corresponding to the physical distance between points in space.

Generalizations of Metrics The following are important generalizations of metrics:

- A *pseudo-metric* is a function $d : V \times V \to \mathbb{R}$ satisfying all properties of a metric except the identity of indiscernibles. That is, $d(u, v) = 0$ does not necessarily imply $u = v$.
- A *quasi-metric* also satisfies all properties of a metric space, but it relaxes the triangle inequality to:

  $$d(u, w) \leq \mathcal{C}(d(u, v) + d(v, w)),$$

  known as the $\mathcal{C}$*-relaxed triangle inequality*. When $\mathcal{C} = 1$, this reduces to a standard metric space.

For instance, in the context of general relativity, the term pseudo-metric often refers to the metric tensor of spacetime, which is actually a pseudo-Riemannian metric.

2.3 The Inner Product and Inner Product Spaces

In terms of hierarchy, metric spaces form the foundational mathematical structure defining distance, with normed vector spaces and inner product spaces representing progressively more specialized and structured mathematical environments. Normed vector spaces extend metric spaces by integrating a norm that naturally induces a metric, while inner product spaces further enhance this structure by introducing an inner product that generates a norm.

Inner products provide additional structure beyond what a norm alone can offer. Inner products enable definitions of angles, orthogonality, and support advanced computational techniques like Gram-Schmidt orthogonalization, eigenvalue decomposition, and principal component analysis. These operations leverage the geometric insights intrinsic to inner product structures. Also, norms derived from inner products often have smoother behavior compared to arbitrary norms. This characteristic makes inner product spaces especially valuable in optimization contexts, where they facilitate natural gradient calculations and provide well-defined curvature representations. Finally note that inner products induce norms, but not vice versa

Gram-Schmidt orthogonalization is a method used in linear algebra to transform a set of linearly independent vectors into an orthogonal (or orthonormal) set of vectors; eigenvalue decomposition is a mathematical technique that factors a square matrix into a product involving its eigenvalues and eigenvectors; and principal component analysis is a statistical technique used to reduce the dimensionality of a dataset while retaining as much variance as possible.

Given a vector space V over a field $\mathbb{F} = \mathbb{R}$, an *inner product* is a function $\langle\, , \,\rangle : V \times V \to \mathbb{R}$ satisfying for any $u, v, w \in V$ and $\alpha \in \mathbb{R}$:

- *Conjugate (Hermitian) Symmetry:* $\langle u, v\rangle = \overline{\langle v, u\rangle}$
- *Linearity:* $\langle \alpha u, v\rangle = \alpha\langle u, v\rangle$, $\langle u + w, v\rangle = \langle u, v\rangle + \langle w, v\rangle$
- *Positive Semi-Definiteness:* $\langle u, u\rangle \geq 0$, $\langle u, u\rangle = 0 \Leftrightarrow u = 0$

$(V, \langle\, , \,\rangle)$ is called an *inner product space*.

The field can also be $\mathbb{F} = \mathbb{C}$.

The overline $\overline{(\cdot)}$ is used to denote the complex conjugate. For $z = a + bi$, then its complex conjugate is: $\overline{z} = a - bi$. Note that the complex conjugate of a real number is itself.

The following additional property, called *conjugate linearity* in the second argument, is a consequence of the above definition (considering the field to be $\mathbb{F} = \mathbb{C}$ for more generality):

$$\langle u, \alpha v\rangle = \overline{\langle \alpha v, u\rangle} = \overline{\alpha\langle v, u\rangle} = \overline{\alpha} \cdot \overline{\langle v, u\rangle} = \overline{\alpha}\langle u, v\rangle.$$

Here, we have applied in order: conjugate symmetry, linearity in the second argument of the inner product, the distributive property of complex conjugation, and substitution from the conjugate symmetry.

Also, note that as previously discussed, in Einstein summation convention, repeated indices are implicitly summed over. For example, in the case of real vectors, we can write the inner product as:

$$\langle u, v\rangle = u_i v_i,$$

where the repeated index i is implicitly summed over from 1 to n.

Examples of Inner Products

- *Real vectors $\mathbb{R}^n$:* $\langle u, v\rangle = \sum_{i=1}^{n} u_i v_i = u_i v_i = v^\top u$
- *Complex vectors $\mathbb{C}^n$:* $\langle u, v\rangle = \sum_{i=1}^{n} u_i \overline{v}_i = u_i \overline{v}_i = v^* u$
- *Real matrices:* $\langle A, B\rangle = \mathrm{trace}(\mathrm{AB}^\top)$

A square-integrable function is a function f defined on a domain Ω such that the square of its absolute value is integrable over Ω. Specifically, a function $f(x)$ belongs to the space $L^2(\Omega)$ if:

$$\int_\Omega |f(x)|^2 \, dx < \infty.$$

- *Square-integrable functions* $L^2(\Omega)$: $\langle f, g\rangle = \int_\Omega f(x)\overline{g(x)}dx$
- *Square-summable real sequences* ℓ^2: $\langle x, y\rangle = \sum_{i\geq 1} x_i y_i$

A square-summable real sequence is a sequence of real numbers $\{a_n\}_{n=1}^{\infty}$ such that the sum of the squares of its elements is finite:

$$\sum_{n=1}^{\infty} a_n^2 < \infty.$$

Relation to Norms The inner product naturally defines a norm, given by

$$\|u\| = (\langle u, u\rangle)^{1/2}.$$

This norm satisfies the *Cauchy-Schwarz (Bunyakovsky) inequality:*

$$|\langle u, v\rangle| \leq \|u\| \cdot \|v\|.$$

This inequality is crucial because it provides an upper bound on the inner product in terms of the magnitudes (norms) of the vectors, ensuring that the inner product cannot exceed the product of the norms of the vectors.

The cosine of the angle between two vectors is given by

$$\cos\angle(u, v) = \frac{\langle u, v\rangle}{\|u\| \cdot \|v\|},$$

which expresses the relationship between the vectors in terms of their geometric angle. When $\langle u, v\rangle = 0$, the vectors are said to be orthogonal, meaning the angle between them is 90° (i.e., $u \perp v$). This condition is essential for understanding orthogonality in inner product spaces.

Not every norm defines an inner product! A norm that satisfies the *parallelogram law*:

$$2\|u\|^2 + 2\|v\|^2 = \|u + v\|^2 + \|u - v\|^2,$$

can be used to define an inner product via the *polarization identity*:

$$\langle u, v\rangle = \frac{1}{4}\left(\|u + v\|^2 - \|u - v\|^2\right).$$

The parallelogram law provides a critical condition for determining whether a norm arises from an inner product. It describes how the lengths of vectors behave geometrically when combined through addition or subtraction. Specifically, it expresses a relationship between the squares of the lengths of the vectors and their sums and differences, mirroring the geometry of inner product spaces.

If the parallelogram law is not satisfied, then the norm cannot be derived from an inner product. Without this structure, we lose important geometric concepts like orthogonality, angles, and projections, which are fundamental to understanding the behavior of vectors in the space. For example, spaces with norms that do not satisfy the parallelogram law, such as the L_1 norm, do not allow for meaningful definitions of orthogonality or angles.

Theorem 1 (Generalized Pythagorean Theorem)**.** *For a set of pairwise orthogonal vectors $v_1, v_2, \ldots, v_n \in V$ (i.e., $\langle v_i, v_j \rangle = 0$ for $i \neq j$), we have the following property:*

$$\left\| \sum_{i=1}^{n} v_i \right\|^2 = \sum_{i=1}^{n} \|v_i\|^2.$$

This result directly generalizes the *Pythagorean theorem* from Euclidean geometry: when vectors are orthogonal, the square of the norm of their sum is equal to the sum of the squares of their individual norms. For non-orthogonal vectors, the sum will be less than or equal to the square of the norm of the sum, by virtue of the triangle inequality.

The original Pythagorean theorem states that in a right triangle with legs of length a and b, and hypotenuse of length c, the relation $a^2 + b^2 = c^2$ holds. This theorem can be interpreted geometrically in Euclidean space as the sum of the squares of the orthogonal components of a vector.

Metrics, Norms, and Inner Products in Geometric Deep Learning. Geometric Deep Learning leverages metric spaces, norms, and inner products to capture structural relationships in data:

- Metrics define distance measures for comparing data points across graph, manifold, and point cloud representations, as well as in neural latent (embedding) spaces.
- Norms quantify vector magnitudes, enabling scale-invariant feature learning, adaptive feature scaling, and regularization in artificial neural networks.
- Inner products facilitate angle calculations, orthogonal projections, and feature embeddings in high-dimensional spaces.

These geometric structures provide computational foundations for neural architectures like graph convolutions, manifold-based networks, and equivariant and invariant feature transformations in Geometric Deep Learning models.

3 Topological Foundations

As we have seen so far, normed spaces add the ability to measure the length or magnitude of vectors. Metric spaces then enter the picture, with their additional structure allowing us to measure how far apart elements are, just as we measure distances in everyday space. And finally, inner products complete this geometric toolkit by defining angles between elements, enabling us to determine when vectors are perpendicular or parallel, for instance.

Students are often first introduced to this geometric foundations rather than topology, because the former deal with tangible aspects of space, which are familiar in our everyday lives. However, this focus on geometry can sometimes overshadow topology, a more abstract field that underpins many concepts in geometry and other areas of mathematics. In essence, topology is concerned with connectivity and studies properties of space that remain unchanged under continuous deformations, such as stretching and bending. Topological spaces can later be augmented with additional structures to measure geometric quantities, such as a metric.

A solid understanding of topology provides deeper insights into the nature of space and is fundamental for grasping more advanced mathematical and scientific concepts. Therefore, in this section, we aim to take a step back and introduce the reader to basic concepts in topology, particularly focusing on manifolds, which are central to many Geometric Deep Learning generalizations of traditional neural network models.

3.1 Topological Spaces

As previously discussed, a set is a collection of distinct elements. For example, the set of points in the plane $\mathbb{R}^2$ can be written as using the set builder notation:

$$\mathbb{R}^2 = \{(x, y) \mid x, y \in \mathbb{R}\}$$

This is simply a collection of points, but without any additional structure. There is no notion of 'closeness' or 'nearness' between points, and the points are not connected in any way. For instance, the point $(0, 0)$ is not closer to $(0, 1)$ than to $(10, 10)$: both are just unordered elements of the set.

However, when we introduce a *structure* to this set, such as *connectivity* or *topology*, we begin to impose rules on how the points are related. For example, we can define which sets of points are considered 'close' to each other or which subsets of $\mathbb{R}^2$ are 'open'.

Open Intervals and Open Sets In the context of the real line $\mathbb{R}$, an *open interval* is a set of points that does not include its boundary points. For example, the open interval (a, b) is the set of points x such that:

$$a < x < b.$$

This interval contains all points between a and b, but does not include a and b themselves.

In a more general setting, a set U is called *open* if it contains a 'neighborhood' around each of its points. This means that for every point $x \in U$, there is a small region around x that is entirely contained within U. In the context of metric spaces, this is formalized as follows:

> Let (X, d) be a metric space, where d is the distance function. A subset $U \subseteq X$ is *open* if, for every point $x \in U$, there exists a radius $r > 0$ such that the *open ball* $B(x, r) = \{y \in X \mid d(x, y) < r\}$ is entirely contained within U.

Although the above definition is perhaps intuitive, it relies on a distance function. Actually, it is possible to define open sets without relying on a metric space, as we will see next.

Topological Spaces The concept of open sets can be generalized in the context of topological spaces. A topological space is defined as a *set* X together with a collection of subsets $\mathcal{T}$ (called *open sets*) that satisfy certain properties. These properties ensure that the notion of 'openness' is well-behaved.

The symbols $A \cup B$ and $A \cap B$ refer specifically to the union and intersection of two sets, A and B. In contrast, $\bigcup_{\alpha \in A} U_\alpha$ and $\bigcap_{\alpha \in A} U_\alpha$ are more general notations used to describe the union or intersection of a collection of sets $\{U_\alpha\}_{\alpha \in A}$, where the index α ranges over some set A.

> Let X be a set, and $\mathcal{T} \subseteq \mathcal{P}(X)$ the power set of X. Then $\mathcal{T}$ is a topology on X if:
>
> - $\emptyset, X \in \mathcal{T}$,
> - $\bigcup_{\alpha \in A} U_\alpha \in \mathcal{T},$ for any collection $\{U_\alpha\}_{\alpha \in A} \subseteq \mathcal{T}$,
> - $\bigcap_{i=1}^{n} U_i \in \mathcal{T},$ for any finite collection $\{U_i\}_{i=1}^{n} \subseteq \mathcal{T}$.

These conditions specify the following: the empty set and the entire set X must be included in $\mathcal{T}$, arbitrary unions of open sets must be open, and finite intersections of open sets must be open. Note that $\mathcal{T}$ is a set of subsets.

> The pair $(X, \mathcal{T})$ is called a *topological space*. Elements of X are referred to as *points*, and elements of $\mathcal{T}$ are called *open sets*.

A subset $U \subseteq X$ is called *open* if $U \in \mathcal{T}$.

> A subset $U \subseteq X$ is called *open* if $U \in \mathcal{T}$.

Open sets are a generalization of intervals in $\mathbb{R}$, which are open in the sense that they do not include their boundary points. Metric spaces are specific examples of topological spaces, and, similarly, open balls in a metric space are examples of open sets.

Examples of Topological Spaces

- *Euclidean Topology:* For $X = \mathbb{R}^n$, the standard topology is generated by open balls. An open ball in $\mathbb{R}^n$ centered at $x \in \mathbb{R}^n$ with radius $r > 0$ is defined as

  $$B(x, r) = \{y \in \mathbb{R}^n : \|x - y\| < r\}.$$

  The topology $\mathcal{T}$ in this case is the collection of all open sets that can be expressed as arbitrary unions of open balls. That is,

  $$\mathcal{T} = \left\{ U \subseteq \mathbb{R}^n : U = \bigcup_{\alpha \in A} B(x_\alpha, r_\alpha) \text{ for some index set } A \right\},$$

where each $B(x_\alpha, r_\alpha)$ is an open ball.

- *Discrete Topology:* In the discrete topology, every subset of X is open. Therefore, for any set X, the topology $\mathcal{T}$ is the power set of X, i.e.,

$$\mathcal{T} = \mathcal{P}(X) = \{U \subseteq X : U \text{ is a subset of } X\}.$$

- *Trivial Topology:* In the trivial topology, only the empty set $\emptyset$ and the entire set X are open. Therefore, the topology $\mathcal{T}$ is

$$\mathcal{T} = \{\emptyset, X\}.$$

The discrete topology is the finest topology because every subset of the space is an open set, making it the topology with the most open sets. In contrast, the trivial topology is the coarsest possible topology, as it contains the fewest open sets.

3.2 Topological Equivalences

Continuity Continuous maps between topological spaces do not 'break' the space, meaning that small changes in the input correspond to small changes in the output, without any sudden jumps or gaps. In other words, the map allows the space to be deformed without tearing it and it preserves the structure of the space, enabling smooth transitions from one point to another.

This definition of continuity does not require the notion of limits, as in the classical sense, but instead relies purely on the topological structure of the spaces involved.

> A map $F : X \to Y$ between topological spaces is continuous if for every open set $U \in \mathcal{T}_Y$, the preimage $F^{-1}(U)$ is an open set in X, i.e., $F^{-1}(U) \in \mathcal{T}_X$.

Homeomorphisms and Homotopy A *homeomorphism* is a special type of continuous map that has a continuous inverse.

It is quite common to confuse homeomorphisms with homomorphisms. A homomorphism is a structure-preserving map between two algebraic structures of the same type, as we saw earlier for groups. In contrast, a homeomorphism is a bijective map between two topological spaces that is continuous and has a continuous inverse. In short, homomorphisms pertain to algebra, while homeomorphisms arise in the context of topology.

> A map $F : X \to Y$ is a *homeomorphism* if it is bijective, continuous, and its inverse $F^{-1} : Y \to X$ is also continuous.

When such a map between two topological spaces exists, we say that X and Y are *homeomorphic*, meaning they are topologically equivalent. For example, the surface of a sphere and that of a cube are homeomorphic, as one can be continuously deformed into the other without tearing or gluing. Note that a homeomorphism is a strong equivalence and denotes that there is a one-to-one correspondence between points in the spaces.

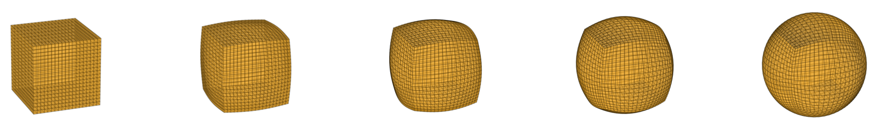


Figure 7: The cube and the sphere are homeomorphic: they are both simply connected (no holes) and can be continuously deformed into each other.

On the other hand, two spaces are *homotopic* (or *homotopy equivalent*) if one can be continuously deformed into the other through a process called *homotopy*. This is a weaker equivalence than homeomorphism since it allows for more general deformations such as collapsing or stretching parts of the space. For example, a circle and a point are homotopy equivalent because the circle can be continuously shrunk to a single point.

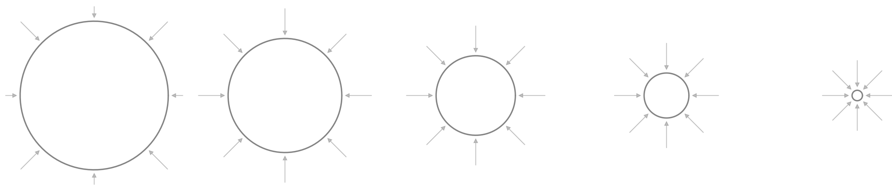


Figure 8: Visualization of circle shrinking into a point. The map between them is surjective, but not injective since all points on the circle are mapped (or collapsed) into the same single point.

3.3 Manifolds

Manifolds are mathematical objects used to describe spaces that may not have a simple, flat, Euclidean structure. However, many natural phenomena occur in spaces (or *domains*) that are curved or have complex structures. Non-Euclidean geometry describes such curved spaces, where the traditional rules of Euclidean geometry no longer hold. Manifolds provide a powerful framework for understanding these spaces.

Euclid's monopoly came to an end in the 19th century, with a remarkable burst of creativity that made geometry into the most exciting field of mathematics.

To understand manifolds, we begin with the simplest notion of a topological manifold, which captures the idea of spaces that locally resemble Euclidean space. From there, we can progressively add more structure to these spaces, eventually obtaining smooth manifolds, which allow for calculus and differential geometry, and Riemannian manifolds, which introduce a way to measure distances and angles.

Topological manifolds A *manifold* is a topological space that locally resembles Euclidean space.

> A topological space $\mathcal{M}$ is an n-dimensional *manifold* if for every point $p \in \mathcal{M}$, there exists an open neighborhood $U \subseteq \mathcal{M}$ and a homeomorphism $\varphi : U \to \mathbb{R}^n$.

Manifolds can be classified based on their dimensionality, such as curves (1-dimensional manifolds), surfaces (2-dimensional manifolds), and higher-dimensional manifolds. Manifolds are the central objects in differential geometry and are fundamental in the study of geometry and physics, particularly in general relativity.

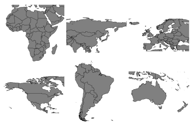
In general relativity, the manifold used to model the universe is a 4-dimensional Lorentzian manifold, which is commonly referred to as spacetime.

The local homeomorphisms between a manifold and Euclidean space are called *charts*. A collection of charts that cover the entire manifold is called an *atlas*.

> An *atlas* for a manifold $\mathcal{M}$ is a collection of *charts* $\{(U_\alpha, \varphi_\alpha)\}$, where U_α is an open subset of $\mathcal{M}$ and $\varphi_\alpha : U_\alpha \to \mathbb{R}^n$ is a homeomorphism. The charts must be compatible, meaning that the transition maps $\varphi_\beta \circ \varphi_\alpha^{-1}$ are homeomorphisms on their domains of overlap.

Smooth manifolds A smooth manifold is a topological manifold equipped with a smooth structure. This means that, in addition to the local homeomorphisms to Euclidean space, the transition maps between overlapping neighborhoods are differentiable. More formally:

The term 'atlas' in mathematics draws an analogy to a collection of maps used in geography. Just as a geographic atlas contains individual maps that collectively describe different regions of the Earth's surface, an atlas on a manifold consists of charts that collectively describe the manifold's structure.



> A topological space $\mathcal{M}$ is an n-dimensional *smooth manifold* if for every pair of points $p, q \in \mathcal{M}$, there exist open neighborhoods $U \subseteq M$ around p and $V \subseteq \mathcal{M}$ around q such that the transition map between the homeomorphisms $\varphi : U \to \mathbb{R}^n$ and $\psi : V \to \mathbb{R}^n$ is a smooth (infinitely differentiable) map.

The tangent space is a key concept for understanding the local geometry of the manifold.

> Given a smooth manifold $\mathcal{M}$ and a point $p \in \mathcal{M}$, the *tangent space* at p, denoted $T_p\mathcal{M}$, is a vector space that represents the possible directions in which one can move away from p. Formally, it is the space of equivalence classes of smooth curves passing through p.

To project points from the tangent space to the manifold and back we use exponential and logarithmic maps.

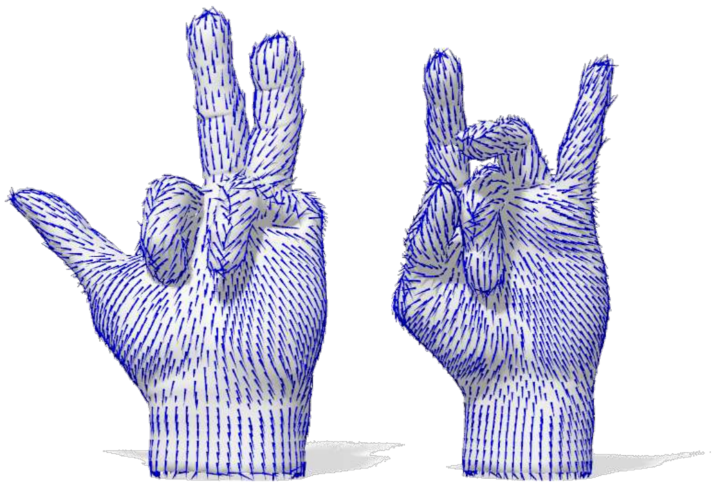


Figure 9: In many problems in Geometric Deep Learning and Geometric Data Processing we work with tangent vector fields. These refer to a smooth assignment of a tangent vector to each point on a manifold.

Diffeomorphisms allow for the transfer of geometric and differential properties between manifolds that share similar local structures.

> A map between two manifolds $\varphi : \mathcal{M} \to \mathcal{N}$ is a *diffeomorphism* if: φ is smooth (infinitely differentiable), φ is bijective, and $\varphi^{-1} : \mathcal{N} \to \mathcal{M}$ is also smooth.

While both homeomorphisms and diffeomorphisms are bijections that preserve certain structures, homeomorphisms preserve topological properties (such as continuity and connectedness), whereas diffeomorphisms preserve smooth (differentiable) structures.

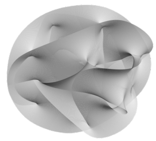
The smooth structure of smooth manifolds allows for the definition of smooth functions, smooth curves, and other objects in differential geometry, making them central to the study of calculus on manifolds.

Riemannian Manifolds A Riemannian manifold is a smooth manifold equipped with a Riemannian metric, which is a smoothly varying inner product on the tangent spaces of the manifold. Formally:

> A smooth manifold $\mathcal{M}$ is a *Riemannian manifold* if it is equipped with a Riemannian metric, which is a smooth assignment of an inner product on the tangent space at each point $p \in \mathcal{M}$, i.e., a map $g_p : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$ that is smooth in p, where $T_p\mathcal{M}$ is the tangent space at p. We typically denote the Riemannian manifold as a tuple $(\mathcal{M}, g)$.

The Riemannian metric enables the measurement of distances between points and the definition of geodesics, which are the shortest paths between points. In general, these geodesics may not be straight lines. For example, when connecting two points on the surface of a sphere, the shortest path is an arc of a great circle, rather than a straight line.

Note that, more generally, manifolds can have variable curvature and very intricate structures, and that homogeneous manifolds with constant curvature, as well as products thereof, are simply easier-to-study special cases. See below an example of variable-curvature Riemannian geometry on manifolds.



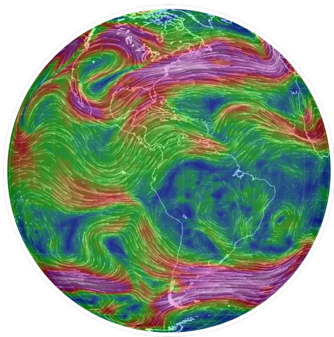


Figure 10: The sphere is an example of a Riemannian manifold, locally resembling Euclidean space. Indeed, when walking on the surface of the Earth, it appears flat. We can define functions on this manifold to characterize various phenomena, such as the distribution of atmospheric pressure or the velocity of the wind.

The sphere in particular is both a *homogeneous manifold* and has *constant curvature*. Without getting into formal definitions, a homogeneous manifold is a manifold with a high degree of symmetry, where the manifold looks the same at every point. A manifold is a *constant curvature manifold* if its curvature (a measure of how the manifold bends in space) is the same at every point.

Product Manifolds Moreover, similarly to Cartesian products between sets, it is also possible to define product manifolds based on the Cartesian product of two subspaces where the product of two manifolds $\mathcal{M}$ and $\mathcal{N}$ is a new manifold $\mathcal{M} \times \mathcal{N}$ with a natural product structure. The tangent space at a point $(p, q) \in \mathcal{M} \times \mathcal{N}$ is the direct sum of the tangent spaces at $p \in \mathcal{M}$ and $q \in \mathcal{N}$, i.e.,

The direct sum $\oplus$ refers to the combination of two vector spaces (or tangent spaces) in such a way that each element of the resulting space is uniquely a pair consisting of one element from each of the original spaces.

$$T_{(p,q)}(\mathcal{M} \times \mathcal{N}) = T_p\mathcal{M} \oplus T_q\mathcal{N}.$$

A Riemannian metric on the product manifold is then defined as the sum of the individual metrics on $\mathcal{M}$ and $\mathcal{N}$. For example, taking the Cartesian product of two spheres we obtain a torus. In ML we can use product manifolds to obtain more complex but computationally tractable manifolds based on simpler manifolds with closed-form solutions.

3.4 The Manifold Hypothesis

Many ML and AI algorithms rely on the *manifold hypothesis* (sometimes also called the manifold assumption), which suggests that although most datasets seem to be high-dimensional in the original data space, data points can actually be described by a low-dimensional manifold which resides within the observed high-dimensional space. This could explain why high-dimensional datasets that appear to require a great number of parameters to be described, can in practice be summarized using lower dimensional latent variables. As a disclaimer, note that the term 'manifold' is used loosely in this context and not in a mathematically rigorous sense, as there are no formal guarantees that the low-dimensional representation possesses the mathematical properties discussed earlier.

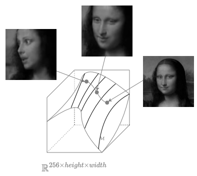
Often the term manifold is abused in ML and AI.

This concept can be more easily illustrated with an example. Consider a dataset comprising grayscale images of a given height and width. The dataset

$$\mathcal{D} = \{\text{image}_1, \text{image}_2, \ldots, \text{image}_n\} \subset \mathbb{R}^{256\times\text{height}\times\text{width}},$$

where the data space lies within (or is a subset of) a $256 \times$ height $\times$ width hypercube. However, most of the coordinates in this hypercube correspond to noisy, random images, while only a few data points are valid images. Specifically, as shown in Figure 11, a data point on the manifold corresponds to a face, whereas a random point in the hypercube is simply noise. The underlying task in many machine learning algorithms is to learn this low-dimensional manifold that best describes the data.

As shown below, moving along the manifold allows us to retrieve different faces, or variations of the same face with different angles and expressions. It is important to note that simply linearly interpolating between points would result in noisy images. Empirically we find it possible to obtain generally smooth transitions in the latent space of autoencoders, for instance. Nevertheless, as of now there are no theoretical guarantees that such smooth interpolations should exist between any two arbitrary points in general.



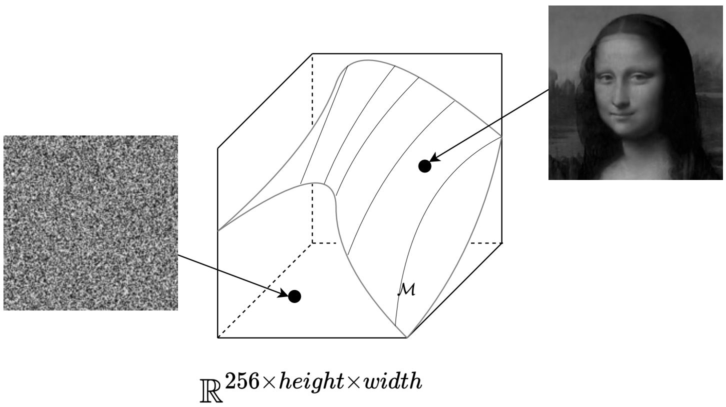


Figure 11: The manifold which encapsulates all images of faces, is expected to be substantially more low-dimensional than the space $\mathbb{R}^{256\times\text{height}\times\text{width}}$. Points on the manifold correspond to valid face images, whereas the remaining points in the hypercube are likely to produce meaningless, noisy images.

Manifolds in Geometric Deep Learning. Geometric Deep Learning aims to extend neural network architectures to effectively handle data defined on general non-Euclidean domains, including manifolds such as surfaces in 3D space or more abstract, higher-dimensional spaces. This involves leveraging tools from differential geometry, like geodesics, curvature, and local charts, to design models that respect the manifold's intrinsic geometry. For example, convolution-like operations on manifolds may be defined in terms of local neighborhoods, where the neighborhood structure is governed by the manifold's geometry rather than a regular grid.

Traditional neural network architectures can be adapted to work on manifolds, meshes, and geometric graphs by focusing on local neighborhoods.

4 Vector calculus

Scalar and vector fields represent quantities that vary across space. These concepts differ from the abstract notion of a vector space, which is purely an algebraic structure. In this section, we examine scalar fields, vector fields, and calculus, which provides essential tools for quantifying variations across space. The latter enables the description of scalar and vector field behavior through operations like differentiation and integration. Differentiation is used to quantify local field behavior, while integral operators establish relationships between infinitesimal variations and macroscopic field properties.

In practice, modeling scalar and vector fields is common in Geometric Deep Learning, particularly in applications such as data-driven physics simulations and 3D graphics. These fields are often represented as, or assumed to be, continuous functions that can be approximated using artificial neural networks. Moreover, beyond particular downstream applications, gradient based optimization is central to training any Deep Learning model.

4.1 Scalar Fields, Vector Fields, and Derivatives

A *scalar field* is a function $f : \mathbb{R}^n \to \mathbb{R}$ that assigns a single scalar value to every point in n-dimensional space, $f(x) = f(x_1, \ldots, x_n)$.

In $\mathbb{R}^3$, $f(x, y, z)$ could represent the temperature at a specific point (x, y, z) in a room. The value of $f(x)$ at each point is a scalar, meaning it has magnitude but no direction.

A derivative captures how a function changes with respect to a change in its input. More concretely, it quantifies the rate of change or the slope of the function at a given point.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth scalar field. A *directional derivative* of f at x in direction $d \in \mathbb{R}^n$ is given by

$$\partial_d f(x) = f_{x_i}(x) = \lim_{\epsilon \to 0} \frac{f(x + \epsilon d) - f(x)}{\epsilon}.$$

In this context, by smoothness we imply being at least twice continuously differentiable (often denoted as $\mathcal{C}^2$), i.e., having continuous second-order derivatives.

The directional derivative quantifies how the function f changes as one moves from the point x in the direction specified by the vector d.

A *partial derivative* of f at x w.r.t. coordinate x_i is given by

$$\frac{\partial}{\partial x_i} f(x) = f_{x_i}(x) = \lim_{\epsilon \to 0} \frac{f(x_1, \ldots, x_i + \epsilon, \ldots, x_n) - f(x_1, \ldots, x_n)}{\epsilon},$$

and is thus a directional derivative in the direction x_i.

Hence, partial derivatives are special cases of directional derivatives, where the direction aligns with the unit vector along the i-th coordinate axis.

In its simplest form, when the scalar field has a single input dimension $f : \mathbb{R} \to \mathbb{R}$, the derivative $f'(x)$ measures the rate of change of f with respect to the single variable x, and we can simply right $f'(x) = \frac{d}{dx} f(x)$, instead of using the ∂ notation.

A *vector field* is a function $F : \mathbb{R}^n \to \mathbb{R}^m$ that assigns a vector to each point in space.

For instance, in $\mathbb{R}^3$, $F(x, y, z) = (F_1(x, y, z), F_2(x, y, z), F_3(x, y, z))$ might represent the velocity of a fluid or the direction and magnitude of a force at each point in space. In this physical example, the value of $F(x)$ at each point has both magnitude and direction, distinguishing it from a scalar field. Note, however, that in the mathematical sense, a vector field is simply a function that assigns a vector to each point in some domain, hence, strictly speaking each of the vector field components can be an independent scalar function.

While the definitions above assume the domain is Euclidean $\mathbb{R}^n$, they extend naturally to more general domains Ω, such as graphs or manifolds. In such cases, derivatives are interpreted using the domain's intrinsic structure (e.g., graph gradients or Laplacians for graphs, and covariant derivatives on manifolds). We will discuss this in more depth in Section 7.

Numerical Methods and Approximations of the Derivative To compute derivatives in practical settings, especially when analytical expressions are unavailable, numerical methods are used. These approximations leverage finite differences to estimate derivatives:

For a scalar field $f : \mathbb{R} \to \mathbb{R}$, the derivative $f'(x)$ at a point x can be approximated using finite differences:

- *Forward Difference:*

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

  where $h > 0$ is a small step size.

- *Backward Difference:*

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}.$$

- *Central Difference:*

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

Central differences are generally more accurate, as they reduce the truncation error to $\mathcal{O}(h^2)$.

The notation, $\mathcal{O}(h^2)$, is called 'Big-O' notation, and it indicates that the leading term of the truncation error is proportional to h^2. This effectively means that the error increases quadratically as a function of the step size.

Finite difference methods introduce truncation errors due to the approximation of the limit. The magnitude of the error depends on the choice of h.

4.2 Gradient

The *gradient* of f is a vector-valued function (*vector field*) $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ satisfying $\langle \nabla f(x), d \rangle = \partial_d f(x)$ for all $x, d \in \mathbb{R}^n$.

The gradient is a linear functional assigning to each direction how much the function f

changes in that direction. We stress that vectors should be correctly treated as *abstract objects* rather than their coordinates in some basis. However, if one wishes to express the gradient w.r.t. to the standard basis of unit vectors $\{e_1, \ldots, e_n\}$ on $\mathbb{R}^n$, this is possible by applying $\langle \nabla f(x), e_i \rangle = \frac{\partial}{\partial x_i} f(x)$. This leads to the usual (somewhat primitive) way of thinking of the gradient as a vector of partial derivatives,

$$\nabla f(x) = \left(\frac{\partial}{\partial x_1} f(x), \ldots, \frac{\partial}{\partial x_n} f(x) \right).$$

Using the gradient, one can provide a linear approximation (first-order *Taylor expansion*) of f around x,

$$f(x + dx) = f(x) + \langle \nabla f(x), dx \rangle + \mathcal{O}(\|dx\|^2),$$

where dx is some infinitesimal displacement. Note the direct relation to numerical methods and the forward difference.

The Taylor series expansion provides a polynomial approximation of the smooth function f.

The Jacobian Matrix The *Jacobian matrix* generalizes the gradient to vector fields. For a vector-valued function $F : \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian matrix $J_F(x)$ at a point $x \in \mathbb{R}^n$ is defined as the matrix of all first-order partial derivatives of the components of F. That is,

$$J_F(x) = \left[\frac{\partial F_i}{\partial x_j} \right]_{i=1,\ldots,m, j=1,\ldots,n} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}.$$

Each element of the Jacobian represents how a single component of the vector field F changes in response to a change in one of the coordinates of the domain. The Jacobian provides valuable information about the local behavior of the function, such as how the function stretches or compresses space.

4.3 Integrals

The *integral* of a function f over a domain Ω is a value that represents the total accumulation of f across Ω. For functions $f : \mathbb{R}^n \to \mathbb{R}$, the integral is formally defined as

$$\int_\Omega f(x)\, dV,$$

where dV denotes the infinitesimal volume element.

Integration generalizes the notion of summation to continuous domains. For scalar functions f, the integral provides a measure of how f 'adds up' across the domain Ω. For instance, in the case of $n = 1$, integration corresponds to calculating the signed area under the curve $f(x)$ over an interval. In higher dimensions, the infinitesimal volume element dV depends on the coordinate system used. For Cartesian coordinates in $\mathbb{R}^n$, $dV = dx_1 dx_2 \cdots dx_n$. In polar, cylindrical, or spherical coordinates, dV includes factors to account for the geometry of the domain.

When the domain Ω is defined by bounds on individual coordinates, the multi-dimensional integral can be split into a series of one-dimensional integrals. This is known as Fubini's theorem.

Riemann Integral The Riemann integral is one of the foundational approaches to defining integration.

For a bounded function $f : [a, b] \to \mathbb{R}$, its *Riemann integral* is defined as the limit of Riemann sums:

$$\int_a^b f(x)\, dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i^*)\Delta x_i,$$

where $[a, b]$ is divided into n subintervals of width Δx_i, and x_i^* is a chosen point in each subinterval.

If the function is not bounded or if it presents severe discontinuities, the Riemann integral fails. We say that such functions are, not Riemann integrable. Alternatives like the Lebesgue integral can handle such cases.

This approach intuitively captures the idea of summing up small contributions $f(x_i^*)\Delta x_i$. Similar to the forward difference method for approximating derivatives, when the closed-form solution to an integral is unknown, the Riemann sum is often used as a numerical approximation in computational methods.

Line and Surface Integrals Integration extends beyond volumes to lower-dimensional objects, such as curves and surfaces.

A *line integral* accumulates a function f along a curve C:

$$\int_C f(x)\, ds,$$

where ds is the infinitesimal arc length.

A *surface integral* accumulates a function f on a S, with the infinitesimal area element dA:

$$\int_S f(x)\, dA.$$

Fundamental Theorem of Calculus (FTC) The Fundamental Theorem of Calculus bridges the concepts of integration and differentiation.

Theorem 2 (Fundamental Theorem of Calculus)**.** *In one dimension, for a function f with antiderivative F:*

$$\int_a^b f(x)\, dx = F(b) - F(a),$$

where $b > a$.

An anti-derivative of a function f is a function F such that $F' = f$. Note that a given function can have infinite many anti-derivatives. For instance, if $F'(x) = f(x)$ then $F(x) + C$ for any constant C is also an anti-derivative of $f(x)$.

4.4 Divergence

Let $F : \mathbb{R}^n \to \mathbb{R}^m$ be a smooth *vector field*, $F(x) = (F_1(x), \ldots, F_n(x))$.

The *divergence* of F is a scalar field $\mathrm{div}F : \mathbb{R}^n \to \mathbb{R}$, satisfying

$$\mathrm{div}F(x) = \sum_{i=1}^{n} \frac{\partial}{\partial x_i} F_i(x) \quad \equiv \quad \nabla \cdot F.$$

Thinking of $F(x)$ as a flow around x, the divergence can be given the interpretation of the density of an outward flux from an infinitesimal volume around x.

Theorem 3 (Gauss-(Ostrogradsky-Stokes) or simply Divergence theorem)**.** *Let $\Omega \subseteq \mathbb{R}^n$ be a region in space with boundary $\partial\Omega$. Then,*

$$\int_{\Omega} \mathrm{div}F dV = \int_{\partial\Omega} \langle F, \hat{n} \rangle dS,$$

where $\hat{n}(x)$ denotes the unit normal vector to the boundary surface $\partial\Omega$ at point x on thereon.

The unit normal vector $\hat{n}(x)$ is a vector of length 1 that is perpendicular to the tangent plane of the boundary $\partial\Omega$ at point x. Its direction is chosen conventionally to point outward from Ω unless stated otherwise. The boundary integral $\int_{\partial\Omega}$ represents integration over the boundary surface $\partial\Omega$. The scalar product $\langle F, \hat{n} \rangle$ measures how the vector field F aligns with the normal direction, while dS indicates the infinitesimal surface area element on $\partial\Omega$.

Note that in the above theorem one assumes that Ω is a smooth region and likewise F is a sufficiently smooth vector field (at least continuously differentiable).

The divergence theorem is a mathematical statement of the physical conservation law that, in the absence of the creation or destruction of matter, the density within a region of space can change only by having it flow into or away from the region through its boundary.

In a sense, the divergence does an operation 'opposite' to that of the gradient; in fact, the two operators are adjoint w.r.t. the appropriate inner products defined on the spaces of scalar and vector fields:

$$\langle \nabla f, F \rangle = -\langle f, \mathrm{div}F \rangle.$$

The negative sign in the adjoint relationship does not prevent them from being adjoint operators; however, we sometimes refer to such operators as skew-adjoint operators to distinguish them from the perhaps more standard positive case.

More concretely, let $\Omega \subset \mathbb{R}^n$ be a bounded domain with smooth boundary $\partial\Omega$. Define inner products, for scalar fields $f, g \in C^\infty(\Omega)$:

$$\langle f, g \rangle_{L^2(\Omega)} = \int_{\Omega} fg \, dx,$$

and or vector fields $F, G \in [C^\infty(\Omega)]^n$:

$$\langle F, G \rangle_{L^2(\Omega)} = \int_{\Omega} F \cdot G \, dx.$$

The left side of the original expression expands as:

$$\langle \nabla f, F \rangle_{L^2(\Omega)} = \int_{\Omega} \nabla f \cdot F \, dx = \int_{\Omega} \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} F_i \, dx$$

Let us apply integration by parts to each term in the summation above:

$$\int_{\Omega} \frac{\partial f}{\partial x_i} F_i \, dx = \int_{\Omega} f F_i dx - \int_{\Omega} f \frac{\partial F_i}{\partial x_i} \, dx = \int_{\partial\Omega} f F_i \hat{n}_i \, dS - \int_{\Omega} f \frac{\partial F_i}{\partial x_i} \, dx,$$

where the boundary terms comes from the divergence theorem (Theorem 3) and we transition from the volume element dx to the surface element dS. Summing over i from 1 to n:

$$\langle \nabla f, F \rangle_{L^2(\Omega)} = \int_{\partial\Omega} f(F \cdot \hat{n})\, dS - \int_{\Omega} f \sum_{i=1}^{n} \frac{\partial F_i}{\partial x_i}\, dx.$$

Since $\mathrm{div} F = \nabla \cdot F = \sum_{i=1}^{n} \frac{\partial F_i}{\partial x_i}$:

$$\langle \nabla f, F \rangle_{L^2(\Omega)} = \int_{\partial\Omega} f(F \cdot \hat{n})\, dS - \int_{\Omega} f(\mathrm{div} F)\, dx.$$

The boundary term vanishes under any of these conditions:

- Dirichlet boundary condition: $f|_{\partial\Omega} = 0$
- $F|_{\partial\Omega} = 0$
- Normal component vanishes: $F \cdot n|_{\partial\Omega} = 0$
- If $\Omega = \mathbb{R}^n$ and F decays faster than $\|x\|^{-n}$ as $\|x\| \to \infty$

Adopting any of the above:

$$\langle \nabla f, F \rangle_{L^2(\Omega)} = \int_{\partial\Omega} f(F \cdot \hat{n})\, dS - \int_{\Omega} f(\mathrm{div} F)\, dx = 0 - \int_{\Omega} f(\mathrm{div} F) = -\langle f, \mathrm{div} F \rangle_{L^2(\Omega)}.$$

4.5 Laplacian

The Laplacian operator is a measure of how a function behaves locally in terms of its rate of change.

The Laplacian of a scalar field f is given by

$$\Delta f(x) = \mathrm{div} \nabla f.$$

It is common to define the Laplacian as $-\mathrm{div}\nabla f$, to make it a positive-semidefinite operator.

The quadratic functional $\langle f, \Delta f \rangle = \langle \nabla f, \nabla f \rangle$, known in physics as the *Dirichlet energy*, is a measure of how variable the function f is.

Theorem 4. *The Laplacian is rotation-invariant.*

Proof. Write the Laplacian as the trace of the Hessian, $\Delta f(x) = \mathrm{tr}(\nabla^2 f(x))$. Note that when representing the Hessian as a matrix w.r.t. the standard basis, its diagonal contains second order derivatives $\frac{\partial^2}{\partial x_i^2} f(x)$:

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

Let Ax be some transformation of coordinates. Then, applying the chain rule, we have

$$\begin{aligned}\nabla_x f(Ax) &= A^\top \nabla_{Ax} f(Ax) \\ \nabla_x^2 f(Ax) &= A^\top \nabla_{Ax}^2 f(Ax) A.\end{aligned}$$

Assuming A is an orthogonal matrix ($AA^\top = A^\top A = I$) and using matrix commutativity under trace we get

The trace of a product of matrices has the property $\mathrm{tr}(XY) = \mathrm{tr}(YX)$.

$$\begin{aligned}\Delta_x f(Ax) &= \mathrm{tr}(A^\top \nabla_{Ax}^2 f(Ax) A) \\ &= \mathrm{tr}(\nabla_{Ax}^2 f(Ax) AA^\top) \\ &= \mathrm{tr}(\nabla_{Ax}^2 f(Ax)) = \Delta_{Ax} f(Ax)\end{aligned}$$

□

This invariance suggests that the behavior of the Laplacian does not depend on the specific orientation of the coordinate system, but rather on the intrinsic geometry of the scalar field itself.

When transforming coordinates, a change of basis can be represented by multiplying by a matrix A. If A is an orthogonal matrix, the transformation does not distort the geometry of the space, that is, distances and angles remain unchanged. This is a necessary condition for the invariance of the Laplacian under rotation.

4.6 Gradient Descent Optimization in Deep Learning

In Deep Learning, gradients play a central role in training models, that is, in optimizing the parameters of artificial neural networks. Although we have not yet introduced artificial neural networks properly, we can think of them as mapping functions (vector fields) $F(x; w) : \mathbb{R}^n \to \mathbb{R}^m$ parametrized by a set of weights (and biases) w.

Loss Functions as Scalar Fields A loss function can be thought of as a scalar field, $\mathcal{L}(w)$, where w represents the model parameters. The loss function assigns a scalar value that indicates how well the model performs. In *supervised learning*, this is generally computed with respect to some reference *ground truth* prediction

$$\mathcal{L}(w) = \mathcal{L}(F(x; w), \hat{y}),$$

where $\hat{y}$ is the ground truth (or the label), $F(x; w)$ represents the artificial neural network output (or prediction), and the loss is, for instance, the mean squared error loss in some regression tasks. Note, however, that the exact setup is task dependent, and more generally we can think of the loss function as returning a scalar based on the artificial neural network parameters w.

> A loss function $\mathcal{L}(w)$ is a scalar field that assigns a scalar value to each set of parameters w, quantifying the model's error.

Just like in vector calculus, we are interested in how $\mathcal{L}(w)$ changes with respect to small changes in the parameters w. This is captured by the gradient of $\mathcal{L}(w)$, denoted as $\nabla\mathcal{L}(w)$. The gradient tells us the direction and rate at which the loss function increases most rapidly. By adjusting the parameters in the opposite direction of the gradient (steepest descent), we can minimize the loss.

Gradient Descent Optimization Gradient descent is the most common optimization method used in Deep Learning.

Typically stochastic gradient descent (SGD) is mentioned as the optimization technique of choice in most textbooks. However, in contemporary Deep Learning more modern variations of SGD are used, such as the AdamW optimizer.

> *Gradient descent* leverages the gradient $\nabla\mathcal{L}(w)$ of the loss function to adjust the parameters of a parametrized model in order to minimize the loss,
>
> $$w_{t+1} = w_t - \eta\nabla\mathcal{L}(w_t),$$
>
> where w_t are the model parameters at iteration (or time step) t , η is the learning rate, a scalar that controls the step size, and $\nabla\mathcal{L}(w_t)$ is the gradient of the loss function with respect to the parameters at w_t.

The gradient guides the model parameters toward a local minimum of the loss. Using this procedure we 'translate the weights in space', from an initial random configuration to a suitable location that is able to model the data with low error. That is, the final weight configuration is a able to mimic the patterns present in the data.

Loss plot credits to the research paper *'Visualizing the Loss Landscape of Neural Nets'*.

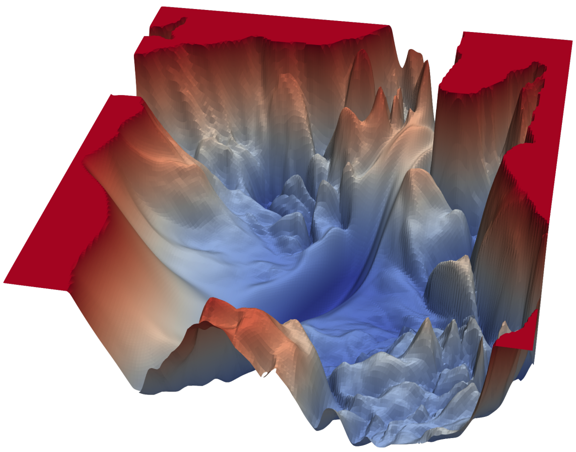


Figure 12: Example loss landscape visualization for a neural network.

Backpropagation and the Chain Rule The gradient of the loss function with respect to the model parameters is typically computed using *backpropagation*. This method relies on the chain rule of calculus to propagate gradients through the network.

Given a point $x \in \mathbb{R}^n$, the composition of two vector fields $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^m \to \mathbb{R}^p$ is written as

$$g \circ f(x) = g(f(x)),$$

which represents the transformation of x through both functions f and g.

The *chain rule* states that given two vector fields $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^m \to \mathbb{R}^p$ and their respective Jacobian matrices J_f and J_g, the derivative of their composition is given by the matrix product:

$$\frac{d}{dx}(g \circ f(x)) = J_g(f(x)) \cdot J_f(x)$$

Artificial neural networks are composed of multiple layers, which can be understood in terms of function composition. The gradient of the loss function $\mathcal{L}$ with respect to each layer's weights is computed iteratively:

$$\nabla_{w^{(l)}}\mathcal{L} = J_L(a^{(L)}) \cdot J_{a^{(L)}}(a^{(L-1)}) \cdot ... \cdot J_{a^{(l+1)}}(w^{(l)})$$

where $a^{(l)}$ is the activation (the output of an intermediate transformation) of the l-th layer.

The Jacobian-based representation can handle cases where activations or transformations are vector-valued, which is typically the case in Deep Learning (technically we work with tensors which becomes even more complex). In the scalar or element-wise gradient context, we can rewrite the expression above as

$$\nabla_{w^{(l)}}\mathcal{L} = \frac{\partial \mathcal{L}}{\partial a^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial a^{(L-1)}} \cdot \dots \cdot \frac{\partial a^{(l+1)}}{\partial w^{(l)}},$$

which may be more accessible to readers less familiar with matrix calculus.

Backpropagation uses the chain rule to compute gradients of the loss function with respect to each layer's weights, which are then used to update the weights of artificial neural networks in an iterative fashion.

Vector Calculus and the Laplacian in Geometric Deep Learning. Beyond other use cases of the gradient such as in gradient descent optimization, in Geometric Deep Learning, the Laplacian is often used to understand the smoothness of functions defined on graphs or manifolds. These structures, such as the vertices and edges of a graph, or the points on a surface, require modifications of traditional calculus tools to account for the inherent irregularities of the data. Hence, vector calculus is not only foundational in classical analysis but are also key components in the development of algorithms for learning over non-Euclidean data.

5 Functional Analysis

Functional analysis is a branch of mathematical analysis that studies spaces of functions and the operators that act on them. Functional analysis provides a powerful framework for understanding infinite-dimensional spaces, where classical linear algebraic methods fail, and establishes the foundation for spectral theory. This section explores key concepts such as completeness, convergence, and the structural properties of vector spaces, with a focus on Banach and Hilbert spaces as fundamental mathematical structures.

Banach and Hilbert Spaces in Geometric Deep Learning. Banach and Hilbert spaces serve as a critical foundation for key concepts such as eigenfunctions, eigenvalues, and Fourier analysis, which we will study in Section 6 and which are widely used in many Geometric Deep Learning algorithms. We encourage readers to review the material on Banach and Hilbert spaces, operators, and functionals. While an in-depth study of these concepts may not be necessary, a basic understanding is useful to tackle spectral theory.

5.1 Cauchy Sequences and Banach Spaces

A sequence of vectors $v_1, v_2, \ldots \in V$ in a normed vector space V is a *Cauchy sequence* if for every $\epsilon > 0$, there exists an integer N such that

$$\|v_m - v_n\| < \epsilon \quad \text{for all } m, n > N.$$

As indices m and n become arbitrarily large, the vectors v_m and v_n approach each other in norm, satisfying:

$$\lim_{m,n\to\infty} \|v_m - v_n\| = 0.$$

Critically, a Cauchy sequence does not inherently guarantee a limit within the space V. The existence of such a limit depends on the space's completeness.

A *Banach space* is a normed vector space V that is complete, meaning every Cauchy sequence $(v_n)_{n\geq 1}$ has a limit $v \in V$ such that:

$$\lim_{n\to\infty} \|v_n - v\| = 0,$$

equivalently converging in the topology induced by the norm:

$$\lim_{n\to\infty} v_n = v.$$

Consider two spaces $V_1 = (0, 1]$ and $V_2 = (0, 1)$, and the sequence $d_n = 1 - \frac{1}{n}$, where n is a positive integer. As $n \to \infty$, the sequence tends to 1. In the case of V_1 the sequence converges within the space. On the other hand, in V_2 the boundary is not part of the space, and hence the sequence does not converge within V_2 even though it is Cauchy.

Banach spaces provide a framework for studying convergence in infinite-dimensional spaces, and they generalize the notion of completeness from real numbers to vector spaces.

A prototypical Banach space is ℓ^p (for $1 \leq p < \infty$), defined by sequences $(x_n)_{n\geq 1}$ satisfying:

$$\|x\|_p = \left(\sum_{n=1}^{\infty} |x_n|^p\right)^{1/p} < \infty.$$

The importance of completeness is illustrated by a counterexample in $\mathbb{Q}$ with the absolute value norm. Consider the sequence approximating $\sqrt{2}$:

$$v_n = \text{the } n\text{-th rational approximation of } \sqrt{2}.$$

This sequence is Cauchy in $\mathbb{Q}$, but its limit $\sqrt{2}$ lies outside $\mathbb{Q}$. This demonstrates why completeness is crucial: it prevents Cauchy sequences from 'escaping' the original space.

5.2 Hilbert Spaces

> A *Hilbert space* is a complete inner product space.

Hilbert spaces combine the algebraic structure of inner products with the topological properties of completeness. Completeness ensures that the space is well-suited for analyzing convergence of Fourier series, solving partial differential equations, and modeling quantum systems. Hilbert spaces unify algebra, geometry, and analysis in an infinite-dimensional setting.

Hilbert spaces extend the notion of Banach spaces by introducing an inner product $\langle \cdot, \cdot \rangle$ that induces the norm:

$$\|v\| = \sqrt{\langle v, v \rangle}.$$

The inner product allows Hilbert spaces to generalize the geometry of finite-dimensional Euclidean spaces to infinite dimensions. Key examples include L^2 (square-integrable) spaces, where functions are treated as infinite-dimensional vectors.

Orthogonal Bases Let V be a Hilbert space and let $S \subseteq V$.

$$\text{span}(S) = \left\{ \sum_{i=1}^{n} \alpha_i v_i : n \in \mathbb{N},\, v_i \in S,\, \alpha_i \in \mathbb{C} \right\}$$

is the set of all finite linear combinations from S.

S is *linearly independent* if for any finite subset $\{v_1, \ldots, v_n\} \subseteq S$ and any coefficients $\alpha_1, \ldots, \alpha_n \in \mathbb{C}$,

$$\sum_{i=1}^{n} \alpha_i v_i = 0 \implies \alpha_i = 0\, \forall i.$$

The equation states (contrapositive form) that if the linear combination equals the zero vector, then all the coefficients α_i must be zero. This is a defining property of linear independence.

S is *orthogonal* if $\langle u, v \rangle = 0 \quad \forall u, v \in S$ s.t. $u \neq v$.

As mentioned in Section 2.3, orthogonality is typically denoted via $u \perp v$. To denote that vectors are orthonormal sometimes the following notation is used: $u \perp\!\!\!\perp v$.

S is *orthonormal* if it is orthogonal and all vectors have unit length, i.e. $\|u\| = 1 \quad \forall u \in S$.

When $\{e_i\}_{i \in I}$ forms an orthonormal basis for V, every element $v \in V$ has a unique infinite representation:

$$v = v_1 e_1 + v_2 e_2 + \cdots = \sum_{i \in I} v_i e_i = v_i e_i = \sum_{i \in I} \langle v, e_i \rangle e_i,$$

where $\langle v, e_i \rangle$ are the *Fourier coefficients* (Section 6.2), and the series converges in the norm induced by the inner product.

Sometimes Hilbert spaces are tacitly assumed separable, yielding the property of isometry to ℓ^2.

Functions as Infinite-Dimensional Vectors in L^2 A square-integrable function is a function f defined on a domain Ω such that the square of its absolute value is integrable over Ω. Specifically, a function $f(x)$ belongs to the space $L^2(\Omega)$ if:

$$\int_\Omega |f(x)|^2 \, dx < \infty.$$

Functions in L^2 spaces can be understood as infinite-dimensional vectors by representing them in terms of a set of basis functions. Just as finite-dimensional vectors in $\mathbb{R}^n$ can be expressed using a basis (e.g., $v = v_1e_1 + v_2e_2 + \cdots + v_ne_n = v_ie_i$), a function $f(x)$ in L^2 can be written as a linear combination of basis functions:

$$f(x) = f_1\phi_1(x) + f_2\phi_2(x) + f_3\phi_3(x) + \ldots$$

Here, $\{\phi_k(x)\}_{k=1}^\infty$ is a set of orthonormal basis functions, and the coefficients f_k represent how much of each basis function $\phi_k(x)$ contributes to $f(x)$. The coefficients f_k are computed using the inner product of $f(x)$ with the basis function $\phi_k(x)$:

An orthonormal set of basis functions are orthogonal $\langle \phi_i, \phi_j \rangle = 0$ for $i \neq j$, and normalized $\langle \phi_i, \phi_i \rangle = 1$.

$$f_k = \langle f, \phi_k \rangle = \int f(x)\phi_k(x) \, dx.$$

This step is analogous to finding the components of a vector in $\mathbb{R}^n$ by projecting it onto the coordinate axes. Once the coefficients $f_1, f_2, f_3, \ldots$ are determined, the function $f(x)$ can be viewed as an infinite-dimensional vector:

Analogous to the expression above: $\sum_{i \in I} v_i e_i = \sum_{i \in I} \langle v, e_i \rangle e_i$.

$$f \equiv [f_1, f_2, f_3, \ldots].$$

In this sense, the 'vector' $[f_1, f_2, f_3, \ldots]$ describes $f(x)$ completely, just as the coordinates $[v_1, v_2, \ldots, v_n]$ describe a vector in finite-dimensional space.

For example, consider the interval $X = [0, 1]$ with basis functions $\phi_1(x) = 1$, $\phi_2(x) = \sin(10\pi x)$, and $\phi_3(x) = \cos(\pi x)$. A function $f(x) = 2 + 17\sin(10\pi x) - \cos(\pi x)$ can be written as:

$$f(x) = 2 \cdot \phi_1(x) + 17 \cdot \phi_2(x) - 1 \cdot \phi_3(x).$$

In this case, the coefficients are $f_1 = 2$, $f_2 = 17$, and $f_3 = -1$, and the function $f(x)$ is represented as the vector $[2, 17, -1]$. Extending this idea to infinitely many basis functions gives the full L^2 perspective, where $f(x)$ is reconstructed as a weighted sum of basis functions.

This approach provides an intuitive understanding of functions as vectors in infinite-dimensional spaces, where concepts like orthogonality, projection, and decomposition of functions naturally extend from finite-dimensional vector spaces.

5.3 Operators and Functionals

In the context of Banach and Hilbert spaces, operators and functionals serve as essential tools for understanding the relationships between elements within and across spaces. They form the backbone of functional analysis. For the sake of brevity, here we only provide a very concise and high-level description of the aforementioned concepts.

Operators on Banach and Hilbert Spaces Operators are mappings that transform elements from one space into another while preserving structure. Through operators, we can describe how vectors interact, how they transform, and how these transformations affect the overall structure of the space.

> An *operator* in this context is a map $A : U \to V$ between two spaces U and V (Banach or Hilbert), usually preserving some structure.

Let $(U, \| \|_U)$ and $(V, \| \|_V)$ be Banach spaces with their respective norms, and consider an operator $A : U \to V$.

A is *continuous* if it preserves convergence, i.e., $u_n \xrightarrow{\| \|_U} u \Rightarrow Au_n \xrightarrow{\| \|_V} Au$.

u_n here refers to a sequence in the space $u_n = (u_n)_{n \geq 1}$.

A is *bounded* if $\exists c > 0$ s.t. $\|Au\|_V \leq c\|u\|_U \quad \forall u \in U$.

A is *linear* if $A(\alpha u + \beta w) = \alpha Au + \beta Aw \quad \forall u, w \in U$ and $\alpha, \beta \in \mathbb{C}$.

A is an *isometry* if it is length-preserving, i.e. $\|Au\|_V = \|u\|_U$.

Let $(V, \langle \, , \, \rangle)$ be a Hilbert space and consider an operator $A : V \to V$.

In the context of Hilbert spaces we use the asterisk symbol $(\cdot)^*$ to denote adjoint operators.

A^* is *adjoint to* A if $\langle Au, v\rangle = \langle u, A^* v\rangle \quad \forall u, v \in V$.

A is *self-adjoint* if $A^* = A$, i.e. $\langle Au, v\rangle = \langle u, Av\rangle \quad \forall u, v \in V$.

A is *compact* if it maps weak limits to strong limits, i.e. $v_n \rightharpoonup v \quad \Rightarrow \quad Av_n \to Av$.

Weak limit ($v_n \rightharpoonup v$): v_n converges weakly to v if $\langle v_n, w\rangle \to \langle v, w\rangle$ for all $w \in V$. This means that v_n converges to v in the sense of how they interact with other vectors, but not necessarily in norm.

Strong limit ($v_n \to v$): v_n converges strongly to v if $\|v_n - v\| \to 0$, i.e., the distance between v_n and v in the norm goes to zero.

The *rank* of an operator A, denoted as $\text{rank}(A)$, is the dimension of the image of A, i.e., the number of linearly independent vectors in the set of vectors that A maps to.

Note that in the space of finite-dimensional real vectors, operators can be expressed as matrices: $\langle Au, v\rangle = (Au)^\top v = u^\top (A^\top v) = \langle u, A^\top v\rangle$. More on this next.

Functionals on Hilbert Spaces Functionals are maps that assign scalar values to vectors. They provide a way to probe and measure elements of a space.

> A *functional* is a map of the form $\phi : V \to \mathbb{C}$ on a Hilbert space V.

ϕ is *continuous* if it preserves convergence, i.e., if $v_n \xrightarrow{\| \|_V} v$ in V, then $\phi(v_n) \longrightarrow \phi(v)$, where $\| \cdot \|_V$ is the norm on V.

Note that continuity implies boundedness, that is, there exists a constant C such that $|\phi(v)| \leq C\|v\|_V$.

ϕ is a *linear* functional if $\phi(\alpha v + \beta w) = \alpha\phi(v) + \beta\phi(w) \; \forall v, w \in V$ and $\alpha, \beta \in \mathbb{C}$.

Dual (or *conjugate*) *space* to V is the space of *linear continuous functionals* on V, denoted

$$V^* = \{\phi : V \to \mathbb{C} \;\; \text{linear+continuous}\}$$

The elements of V^* are called *dual vectors*.

6 Spectral Theory

Spectral theory studies the properties of operators and matrices by analyzing their spectra, that is, their eigenfunctions and associated eigenvalues.

6.1 Eigenfunctions and Eigenvalues

Eigenfunctions and eigenvalues arise when we study linear transformations, whether on finite-dimensional vector spaces or infinite-dimensional spaces like function spaces. They allow us to decompose and *diagonalize* operators. This can enable us to work with a simplified version of the original problem, one that might exhibit complex, non-linear dynamics in the original space. These concepts are particularly central to *spectral theory*.

> Let $A : V \to V$ be an operator on Hilbert space V. A vector $v \neq 0$ satisfying for some λ
> $$Av = \lambda v$$
> is called an *eigenfunction* of A, and λ is the corresponding *eigenvalue*.

Note that eigenfunctions are defined up to scale: if v is an eigenfunction of A, so is αv for any $\alpha \neq 0$, since we can multiply both sides of the equation by $A(\alpha v) = \lambda(\alpha v)$ by α. It is common to assume eigenfunctions of unit length, i.e. $\|v\| = 1$.

Eigenvectors and Eigenvalues in Finite-Dimensional Vector Spaces When we are first introduced to eigenvectors and eigenvalues, A typically denotes a matrix, which is a finite, rectangular array of numbers that defines a linear transformation in a finite-dimensional vector space. Eigenvectors are the vectors in the vector space that are scaled by the linear transformation represented by A. In finite-dimensional spaces, eigenfunctions are essentially eigenvectors, but the term *eigenfunction* is more commonly used in the context of infinite-dimensional spaces. For example, if A is an $n \times n$ matrix, eigenvalues and eigenvectors are solutions to the equation:

$$Av = \lambda v, \quad v \neq 0,$$

where v is a vector in $\mathbb{R}^n$ or $\mathbb{C}^n$. This is usually solved finding values of λ that satisfy the *characteristic equation*:

$$\det(A - \lambda I) = 0,$$

where I is the $n \times n$ identity matrix. The solutions $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of A, and for each eigenvalue λ, we find the corresponding eigenvector(s) v by solving the system of linear equations:

$$(A - \lambda I)v = 0.$$

The eigenvalue λ determines how A stretches or compresses the direction v, which remains unchanged under the transformation, except for sign flips.

There can be multiple eigenvectors corresponding to the same eigenvalue. If $\lambda > 0$ the direction of v remains unchanged, but it is stretched if $|\lambda| > 1$ or compressed if $|\lambda| < 1$. Eigenvalues can be negative. If $\lambda < 0$ the direction of v is reversed, since multiplication by a negative scalar reflects the vector across the origin.

Generalization to Hilbert Spaces: Eigenfunctions and Eigenvalues Here, we are interested in the generalization from finite-dimensional vector spaces to infinite-dimensional spaces. In this generalized setting, A is a *linear operator* $A : V \to V$ which acts on vectors in the Hilbert space V, instead of a matrix. In a finite-dimensional space, a matrix A maps vectors in $\mathbb{R}^n$ to $\mathbb{R}^n$, whereas in an infinite-dimensional space, an operator A maps functions in a space such as L^2 to itself. The characteristic equation for eigenvectors $Av = \lambda v$ still applies in this case, but here v might be a function (hence called an *eigenfunction*), and λ is a scalar *eigenvalue* associated with v.

The Spectral Theorem The spectral theorem states that self-adjoint operators, both in finite and infinite-dimensional spaces, can be fully diagonalized in terms of their eigenvalues and eigenfunctions. This theorem plays a crucial role in understanding the structure of such operators in Hilbert spaces. Remember that A is *self-adjoint* if $A^* = A$, i.e. $\langle Au, v\rangle = \langle u, Av\rangle \quad \forall u, v \in V$.

We begin by discussing important properties of self-adjoint operators.

Theorem 5 (Spectral Theorem for Self-Adjoint Operators)**.** *Self-adjoint operators have real eigenvalues.*

Proof. Let $Av = \lambda v$, with $v \neq 0$. Since $A = A^*$, we have:

$$\langle Av, v\rangle = \langle v, Av\rangle.$$

Substituting $Av = \lambda v$, we get:

$$\langle \lambda v, v\rangle = \langle v, \lambda v\rangle.$$

Because λ is a scalar, we can factor it out of both inner products:

$$\lambda\langle v, v\rangle = \overline{\lambda}\langle v, v\rangle.$$

Note that on the right, we have applied conjugate linearity from Section 2.3. Since $v \neq 0$, $\langle v, v\rangle > 0$. Thus, we can divide both sides by $\langle v, v\rangle$ to obtain:

$$\lambda = \overline{\lambda},$$

which implies that $\lambda \in \mathbb{R}$. □

Theorem 6 (Orthogonality of Eigenfunctions)**.** *Eigenfunctions of self-adjoint operators corresponding to different eigenvalues are orthogonal.*

Proof. Let $Av = \lambda v$ and $Aw = \mu w$ with $\lambda \neq \mu$ and $v, w \neq 0$. Since $A = A^*$, we have:

$$\langle Av, w\rangle = \langle v, Aw\rangle.$$

Substituting the eigenvalue equations, we get:

$$\langle \lambda v, w\rangle = \langle v, \mu w\rangle.$$

Since λ and μ are real (from Theorem 5), we can factor out the scalars without conjugation:

$$\lambda\langle v, w\rangle = \mu\langle v, w\rangle.$$

Thus,

$$(\lambda - \mu)\langle v, w\rangle = 0.$$

Since $\lambda \neq \mu$, it follows that:

$$\langle v, w\rangle = 0,$$

i.e., $v \perp w$. □

The set of eigenvalues can be either finite or countably infinite. A set is countable if there is a way to list its elements in a sequence, that is, there is a one-to-one correspondence between the set and the set of natural numbers, $\mathbb{N}$. When we say that the spectrum is discrete we mean that each eigenvalue is separated by some positive distance from others, that is, the eigenvalues are isolated. The only exception is $\lambda = 0$. There is no continuous spectrum where eigenvalues can form a continuous range or interval.

Theorem 7 (Spectral Theorem)**.** *A compact self-adjoint operator $A : V \to V$ has eigenvalues $\{\lambda\}$ with corresponding eigenfunctions $\{v_\lambda\}$ such that:*

$$Av_\lambda = \lambda v_\lambda.$$

These eigenfunctions form an orthonormal basis of V, and the set of eigenvalues is countable. Furthermore, the eigenvalue spectrum is discrete, with the only possible accumulation point being $\lambda = 0$.

This statement implies that the eigenvalues of a compact self-adjoint operator form a countable set, all of which are real. The corresponding eigenfunctions form an orthonormal basis of the Hilbert space V. If $\lambda \neq 0$, then λ is an isolated eigenvalue (discrete spectrum). The only possible accumulation point of the spectrum is $\lambda = 0$.

Thus, the Spectral Theorem builds on the properties established in Theorems 5 and 6 and provides a complete characterization of the structure of a Hilbert space under a compact self-adjoint operator.

Principal Component Analysis (PCA) uses a finite-dimensional version of the Spectral Theorem to identify key directions in data.

Spectral Theorem Example In the following, we provide an illustration of the spectral theorem in the context of a differential operator and verify key properties like self-adjointness and orthogonality of eigenfunctions.

Let us work with

$$L^2([-\pi, +\pi]) = \left\{ f : [-\pi, +\pi] :\to \mathbb{C} \quad \text{s.t.} \quad \int_{-\pi}^{+\pi} |f(x)|^2 dx < \infty \right\},$$

the space of square-integrable periodic functions, meaning their squared magnitude integrates to a finite value, with standard inner product

$$\langle f, g\rangle = \frac{1}{2\pi} \int_{-\pi}^{+\pi} f(x)\overline{g(x)}dx,$$

where $\overline{g(x)}$ denotes the complex conjugate of $g(x)$.

Consider the Laplacian operator (second-order derivative, see Section 4): $\Delta = \frac{d^2}{dx^2}$. First, we verify that Δ is self-adjoint. To do so, we must show:

$$\langle \Delta f, g \rangle = \langle f, \Delta g \rangle \quad \forall f, g \in L^2([-\pi, \pi]).$$

From the product differentiation rule,

$$\frac{d}{dx}(f(x)g(x)) = f'(x)g(x) + f(x)g'(x).$$

Also, the fundamental theorem of calculus tells us,

Assuming continuity and differentiability, or piecewise smoothness.

$$\int_{-\pi}^{+\pi} \frac{d}{dx}(f(x)g(x))\, dx = \left. f(x)g(x)\right|_{-\pi}^{+\pi},$$

and given that we are considering periodic functions, we have the boundary conditions $f(\pi) = f(-\pi)$ and $g(\pi) = g(-\pi)$. Hence,

$$\left. f(x)g(x)\right|_{-\pi}^{+\pi} = f(\pi)g(\pi) - f(-\pi)g(-\pi) = f(\pi)g(\pi) - f(\pi)g(\pi) = 0.$$

Therefore,

$$\int_{-\pi}^{+\pi} \frac{d}{dx}(f(x)g(x))dx = 0 \implies \int_{-\pi}^{+\pi} f(x)g'(x)dx = -\int_{-\pi}^{+\pi} f'(x)g(x)dx,$$

where for simplicity, we ignore complex conjugates. Applying this result to $f'g'$ we have

Let f and g swap roles to obtain both sides of the equation and perform a change of variables.

$$-\int_{-\pi}^{+\pi} f'(x)g(x)dx = \int_{-\pi}^{+\pi} f'(x)g'(x)dx = -\int_{-\pi}^{+\pi} f(x)g'(x)dx$$

from which self-adjointness $\langle \Delta f, g \rangle = \langle f, \Delta g \rangle$ follows

$$\langle \Delta f, g \rangle = \int_{-\pi}^{+\pi} f'(x)g(x)dx = \int_{-\pi}^{+\pi} f(x)g'(x)dx = \langle f, \Delta g \rangle.$$

After having verified the self-adjointness of the Laplacian, let us now consider the Laplacian acting on the function e^{inx} where $n \in \mathbb{Z}$. From $\Delta e^{inx} = \frac{d^2}{dx^2}e^{inx} = -n^2 e^{inx}$, it immediately follows that eigenfunctions have the form e^{inx} with corresponding real eigenvalues $-n^2$. Remember that in infinite-dimensional space, eigenfunctions are linear operators: indeed the Laplacian scales linearly the function e^{inx} by a factor of $-n^2$.

$\frac{d}{dx}e^{ax} = ae^{ax}$

In Theorem 5 we stated that self-adjoint operators have real eigenvalues: $-n^2$ is real. Next, to verify orthogonality and Theorem 6, write

Remember we need to consider the complex conjugate of the eigenfunction: $\overline{e^{imx}} = e^{-imx}$

$$\langle e^{inx}, e^{imx} \rangle = \frac{1}{2\pi}\int_{-\pi}^{+\pi} e^{inx}e^{-imx}dx = \frac{1}{2\pi}\int_{-\pi}^{+\pi} e^{i(n-m)x}dx.$$

For $n \neq m$,

This can be shown using the integral of a complex exponential and rewriting the result in terms of the sine function.

$$\int_{-\pi}^{+\pi} e^{i(n-m)x}dx = 0 \implies \langle e^{inx}, e^{imx} \rangle = 0.$$

This is because the function is periodic with zero average over the full period and shows that distinct eigenfunctions are orthogonal. For $n = m$,

$$\int_{-\pi}^{+\pi} e^{i(n-m)x} dx = \int_{-\pi}^{+\pi} 1 dx = 2\pi \implies \langle e^{inx}, e^{inx} \rangle = 1,$$

which reflects normalization, that is, the eigenfunctions are orthonormal.

Hence we have that,

$$\langle e^{inx}, e^{imx} \rangle = \delta_{nm},$$

where δ_{mn} is the Kronecker delta.

The Kronecker delta is defined as

$$\delta_{nm} = \begin{cases} 1, & \text{if } n = m, \\ 0, & \text{if } n \neq m. \end{cases}$$

Singular Values The spectral theorem focuses on self-adjoint operators. For more general operators, we turn to the concept of *singular values* and their corresponding *singular vectors*. Singular values provide a more general way to characterize how an operator transforms vectors in a space, and they are particularly useful when dealing with non-self-adjoint operators, such as general matrices.

Before providing formal definitions, let us clarify the intuitive difference between eigenvalues and singular values. These quantities capture different aspects of how a linear operator transforms elements of the space. Eigenvalues measure how much a transformation stretches or compresses an eigenfunction along its direction, without changing that direction (except for sign flips). On the other hand, singular values measure the overall magnitude of an operator's action, independent of any specific direction, that is, they describe how much the operator stretches or compresses functions in general. These concepts provide fundamental tools for analyzing operators, whether finite-dimensional (as matrices) or infinite-dimensional.

> An operator $A : V \to V$ is *compact* iff it can be written in the form
>
> $$Aw = \sum_{n \geq 1} \sigma_n \langle v_n, w \rangle u_n, \quad \forall w \in V.$$
>
> $\{\sigma_n\}_{n\geq1}$ are the *singular values* and $\{v_n\}_{n\geq1}$, $\{u_n\}_{n\geq1}$ are the corresponding (left- and right-) *singular vectors* of A.

Singular vectors, both left and right, represent directions in the domain and codomain of A.

Note that this is an alternative definition of compactness. Compact operators are often studied because they have certain nice properties, such as having a countable set of singular values. Importantly, these singular values can accumulate only at zero. This means that after some index N, the singular values become zero, indicating that the operator has finite rank. In this case, the rank of A is equal to N, and we have:

$$\text{rank}(A) = N.$$

In the finite-dimensional case the rank corresponds to the number of linearly independent rows or columns in the matrix representing the operator, whereas in the infinite-dimensional case the rank is the number of non-zero singular values. Even though the operator may act on an infinite-dimensional space, its rank remains finite.

After discussing the most general case, let us now examine particular cases. If A is self-adjoint, we can write it in the form:

$$Aw = \sum_{n \geq 1} \lambda_n \langle v_n, w \rangle v_n, \quad \forall w \in V.$$

Here, $\{\lambda_n\}$ are the eigenvalues of A, and $\{v_n\}$ are the corresponding eigenvectors of A. This is a special case of the more general singular value decomposition, where the singular values coincide with the eigenvalues, and the left and right singular vectors are the same.

Next, let us discuss singular value decomposition (SVD) of matrices.

An $m \times n$ matrix A can be written in the *singular value decomposition (SVD)* form:

$$A = U\Sigma V^* = \begin{pmatrix} | & & | \\ u_1 & \dots & u_n \\ | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix} \begin{pmatrix} - & \overline{v}_1 & - \\ & \vdots & \\ - & \overline{v}_n & - \end{pmatrix},$$

where U is an $m \times m$ unitary matrix whose columns are the *left singular vectors* u_i, Σ is an $m \times n$ diagonal matrix containing the singular values σ_i, and V^* is the conjugate transpose of the $n \times n$ unitary matrix V, whose rows are the *right singular vectors* $\overline{v}_i$.

Example Eigenvalues vs Singular Values To further build on our intuition regarding the difference between eigenvalues and singular values, let us consider a rotation matrix. A rotation matrix has no real eigenvalues because it does not stretch or compress space along fixed directions. However, it has singular values all equal to 1, reflecting that it preserves lengths. More concretely, a rotation matrix R in 2D is defined as:

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

where θ is the rotation angle. To find the eigenvalues, we solve the characteristic equation:

$$\lambda^2 - 2\lambda\cos\theta + 1 = 0.$$

Thus, the eigenvalues are:

$$\lambda = e^{i\theta}, \quad \lambda = e^{-i\theta}.$$

These eigenvalues are complex and lie on the unit circle in the complex plane. Hence, there are no real eigenvalues unless $\theta = 0$ or π (identity and reflection).

The singular values of R are obtained from the eigenvalues of $R^T R$:

Multiplying $R^T R$:

$$R^T R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I.$$

The eigenvalues of $R^T R$ are therefore both 1, and the singular values of R (the square roots of these eigenvalues) are: $\sigma_1 = 1$ and $\sigma_2 = 1$.

6.2 Fourier analysis

In eigenfunctions and eigenvalues, singular value decomposition, and Fourier series, the fundamental concept is the decomposition of an object—whether a self-adjoint operator, an operator, or a function—into a sum of components along specific directions or bases. More commonly, Fourier series are associated with a trigonometric basis (sine, cosine, or complex exponential). However, the concept is general and applies to any orthonormal basis.

Let $\{v_\alpha\}$ be an orthonormal basis in a Hilbert space V. Then, $u \in V$ can be expressed as a *Fourier series*

$$u = \sum_\alpha \langle u, v_\alpha \rangle v_\alpha$$

The coefficients $\langle u, v_\alpha \rangle = \hat{u}_\alpha$ in the above series are called *Fourier coefficients* (or *transforms*) of u.

For clarity, remember that the expression above can be expanded as follows:

$$u = \sum_\alpha \langle u, v_\alpha \rangle v_\alpha = \hat{u}_\alpha v_\alpha = \hat{u}_1 v_1 + \hat{u}_2 v_2 + \hat{u}_3 v_3 + ...$$

Fourier Decomposition for Vectors For vectors, the Fourier decomposition can be written in matrix form:

$$u = \underbrace{\begin{pmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{pmatrix}}_{V} \underbrace{\begin{pmatrix} - & \overline{v}_1^\top & - \\ & \vdots & \\ - & \overline{v}_n^\top & - \end{pmatrix}}_{V^\dagger} u,$$

where $V \in \mathbb{C}^{n \times n}$ is the matrix whose columns are the basis vectors v_i, $V^\dagger$ is the Hermitian conjugate (conjugate transpose) of V, and $V^\dagger u = (\langle u, v_1 \rangle, \langle u, v_2 \rangle, \ldots, \langle u, v_n \rangle)^\top$ contains the Fourier coefficients. Thus

$$u = V(V^\dagger u),$$

where $V^\dagger u$ gives the coefficients, and $V(V^\dagger u)$ reconstructs the vector. From this, it is evident that it is a unitary operation (see below).

A unitary operation is a linear operation that preserves the inner product in a complex vector space.

Continuous Fourier Transform Note that in general, α here can be a continuous index, in which case the sum should be replaced with an integral:

$$\langle u, v_\alpha \rangle = \int u(x)\overline{v_\alpha(x)}\, dx.$$

This is the case with the continuous *Fourier transform* using a basis of the form $e^{i\omega x}$, with $\omega \in \mathbb{R}$:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega x}\, d\omega,$$

where $\hat{f}(\omega)$ are the Fourier coefficients of $f(x)$, representing the contribution of each frequency component. The Fourier coefficients are obtained based on the inner product:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x}\, dx.$$

Note that the computations in the continuous case are analogous to obtaining the coefficients and reconstructing the vector using matrix multiplication, as discussed earlier in the context of vectors.

Fourier Series Example Consider $L^2([-\pi, +\pi])$, the space of square-integrable periodic functions, with the standard inner product $\langle f, g \rangle = \frac{1}{2\pi} \int_{-\pi}^{+\pi} f(x)\overline{g(x)} dx$ and the basis $\{e^{inx}\}_{n \geq 1}$. The Fourier series assume the classical form

$$f(x) = \sum_{n \geq 1} \frac{1}{2\pi} \int_{-\pi}^{+\pi} f(y) e^{-iny} dy\, e^{inx}.$$

The Fourier series provides a discrete decomposition because the function being considered is periodic, leading to discrete frequencies.

Parseval's Identity Parseval's identity establishes that the inner product, and hence the geometry, of a Hilbert space V is perfectly captured by the Fourier coefficients. The identity guarantees that this mapping is an isometry, and it allows us to work with Fourier coefficients as a proxy for the original function or vector.

Theorem 8 (Parseval's identity)**.** *Let $u = \sum_\alpha \hat{u}_\alpha v_\alpha$ and $w = \sum_\alpha \hat{w}_\alpha v_\alpha$ be Fourier series of $u, w \in V$ with respect to the orthonormal basis $\{v_\alpha\}$. Then $\langle u, w \rangle = \sum_\alpha \hat{u}_\alpha \overline{\hat{w}}_\alpha$.*

In other words, we can define a map $V \ni u \mapsto \hat{u} = \{\langle u, v_\alpha \rangle\} \in \ell^2$ from vectors to (square summable) sequences. This map is an isometry :

Recall that an isometry is length-preserving.

$$\|u\|_V^2 = \sum_\alpha |\langle u, v_\alpha \rangle|^2 = \sum_\alpha |\hat{u}_\alpha|^2 = \|\hat{u}\|_{\ell^2}^2.$$

This, in turn, is nothing else but the application of the Pythagorean theorem, Theorem 2.3

(possibly in infinite dimensions),

$$\|u\|^2 = \Big\| \sum_\alpha \langle u, v_\alpha \rangle v_\alpha \Big\|^2 = \sum_\alpha \| \langle u, v_\alpha \rangle v_\alpha \|^2 = \sum_\alpha |\langle u, v_\alpha \rangle|^2,$$

where we used the orthonormality of the basis $\{v_\alpha\}$.

The heat equation Consider the following partial differential equation, called the *heat equation*, under Dirichlet boundary conditions:

$$\begin{cases} \Delta f(x,t) = f_t(x,t) \\ f(x,0) = g(x) \qquad \text{(initial conditions)} \end{cases}$$

on a circle, where $f : S^1 \times [0,\infty) \to \mathbb{R}$ (periodic in the first coordinate) represents the temperature , $\Delta = \frac{\partial^2}{\partial x^2}$ is the one-dimensional Laplacian operator, and $g(x)$ is the initial temperature distribution at time $t = 0$. Since S^1 is a circle, there are no boundary conditions on the spatial domain.

$f(x,t)$ is the temperature at point x at time t.

Fourier analysis was originally developed for solving this kind of partial differential equation (PDE), and we will show how it applies here. First, assume the solution has a separable form:

$$f(x,t) = X(x)T(t),$$

where $X(x)$ is the spatial part and $T(t)$ is the temporal part. Assuming X, T never vanish, we substitute this into the heat equation:

$$\Delta f - \frac{\partial}{\partial t} f = X'T - XT' = 0.$$

Since the above holds for any (x,t), it follows that:

$$\frac{X'}{X} = \frac{T'}{T} = -\lambda \quad \text{(some constant)}.$$

In other words, the spatial and temporal parts of the solution are eigenfunctions of the Laplacian and first-order derivative operators, respectively:

$$X' = \Delta X = -\lambda X, \quad T' = \frac{\partial}{\partial t} T = -\lambda T,$$

which we can express in closed form as:

$$\Delta e^{inx} = -n^2 e^{inx}, \quad \frac{\partial}{\partial t} e^{-n^2 t} = -n^2 e^{-n^2 t},$$

where $\lambda = -n^2$ is the corresponding eigenvalue.

Hence, solutions to the equation take the form $f_n(x,t) = e^{inx} e^{-n^2 t}$. Due to the linearity of the equation, any linear combination of such solutions is also a solution, so the general solution can be written as:

$$f(x,t) = \sum_{n=-\infty}^{\infty} a_n e^{inx} e^{-n^2 t}.$$

Note that we sum over all integer values of n (including both positive and negative values) to account for the full Fourier expansion.

To find a unique solution, we must use the initial condition. The set $\{e^{inx}\}_{n\in\mathbb{Z}}$ forms an orthonormal basis for $L^2(S^1)$. Therefore, we can express the initial condition $g(x)$ as a Fourier series:

$$g(x) = \sum_{n=-\infty}^{\infty} \langle g, e^{inx}\rangle e^{inx},$$

where $\langle g, e^{inx}\rangle$ is the Fourier coefficient for $g(x)$.

Since $f(x,0) = g(x)$, we can identify $a_n = \hat{g}_n = \langle g, e^{inx}\rangle$. Using the standard inner product for periodic functions, we obtain the general solution:

$$\begin{aligned} f(x,t) &= \sum_{n=-\infty}^{\infty} \frac{1}{2\pi}\int_{-\pi}^{\pi} g(y)e^{-iny}\,dy\, e^{inx}e^{-n^2t} \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} g(y) \underbrace{\sum_{n=-\infty}^{\infty} e^{-n^2t}e^{-in(x-y)}}_{h_t(x-y)}\,dy = g\star h_t, \end{aligned}$$

where $\star$ denotes convolution.

The function $h_t(x)$ is called the *fundamental solution* of the heat equation, or the *heat kernel*. In particular, for the case where the initial condition is the Dirac delta function, $g(x) = \delta(x)$ (an impulse initial condition), we have:

$$\langle \delta, e^{inx}\rangle = e^{in0} = 1, \quad \text{so} \quad a_n = 1 \quad \forall n,$$

which implies that the solution is:

$$f(x,t) = \sum_{n=-\infty}^{\infty} e^{-n^2t}e^{inx} = h_t(x).$$

In signal processing terms, h_t is referred to as the *impulse response* of the system.

Spectral Theory in Geometric Deep Learning. Spectral theory provides a mathematically rigorous framework for extending traditional Deep Learning approaches for Euclidean data to irregular domains such as graphs and manifolds while maintaining important properties like translation invariance and locality.

Note that $g(x)$ does not depend on time, so we use the eigenfunctions of the Laplacian.

The Dirac delta function is analogous to the Kronecker delta but in the continuous case. It is defined as:

$$\delta(x-y) = \begin{cases} \infty, & \text{if } x = y, \\ 0, & \text{if } x \neq y, \end{cases}$$

with the important property that its integral over the entire real line is equal to 1:

$$\int_{-\infty}^{\infty} \delta(x-y)\,dy = 1.$$

In the context of Fourier analysis, the Dirac delta function can be represented as:

$$\delta(x-y) = \sum_{n=-\infty}^{\infty} e^{in(x-y)}.$$

The Dirac delta function acts as an identity element in the Fourier transform, meaning that for any function $f(x)$:

$$\int_{-\infty}^{\infty} f(y)\delta(x-y)\,dy = f(x).$$

7 Graph Theory

While continuous geometry might examine smooth curves or surfaces, discrete geometry focuses on structures that can be enumerated or broken down into distinct, countable elements. Graph theory is a subset of discrete geometry that is central to Graph Neural Network (GNNs), which are perhaps the quintessential artificial neural network architecture in Geometric Deep Learning.

Note however that Geometric Deep Learning is a broader framework that extends Deep Learning techniques to non-Euclidean domains, with one such instantiation being learning over graphs.

7.1 Preliminaries on Graphs and Notation

We start by discussing basic definitions and notation to describe graphs.

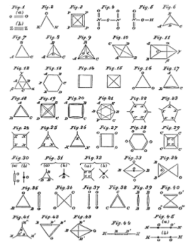
A *graph* is an ordered tuple:

$$G = (V, E),$$

where V is a set of nodes (or vertices), and $E \subseteq (V \times V)$ is a 2-tuple set representing the edges (or links) in the graph.

Edges may be directed or undirected. Directed edges are uni-directional relations from a source node v_i to a target node v_j; thus, $(v_i, v_j) \in E$, and importantly, $(v_i, v_j) \neq (v_j, v_i)$.

J. Sylvester mentions the term ‘graph’ as early as 1878 in a chemical context.



A *directed graph* (or *digraph*) is a graph $G = (V, E)$ where each edge in E is an ordered pair of nodes.

In contrast, undirected edges are bidirectional, so $(v_i, v_j) = (v_j, v_i)$. When an edge connects a node to itself, we call it a self-loop (v_i, v_i).

The (one-hop) *neighborhood* of a node v_i is the set of nodes that share an edge with v_i, denoted as

$$\mathcal{N}(v_i) = \mathcal{N}_i = \{v_j | (v_i, v_j) \in E\}.$$

A *subgraph* $H = (V_H, E_H)$ of a graph $G = (V_G, E_G)$ is a graph where $V_H \subseteq V_G$ and $E_H \subseteq E_G$.

If we consider the set $\{v_i\} \cup \mathcal{N}(v_i)$ as nodes and include all edges in E that connect these nodes, this defines a *neighborhood subgraph* of v_i, which is a subgraph of G.

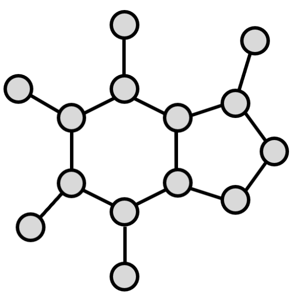


Figure 13: Diagram of a graph with nodes and edges.

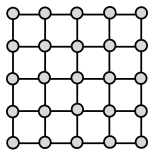
The Adjacency Matrix Graphs can be represented using matrices. For a graph with $N = |V|$ number of nodes, its adjacency matrix $A \in \mathbb{R}^{N \times N}$ represents the connectivity structure between nodes. A can be weighted or unweighted. If it is weighted, its entries $A_{ij} \in \mathbb{R}$ represent the weight or strength of the connection, and if $(v_i, v_j) \notin E$, then $A_{ij} = 0$. $w : E \to \mathbb{R}^+$ is the weight function assigning positive real numbers to edges: if $e = (v_i, v_j)$, then $w(e) = A_{ij}$. In the case of an unweighted adjacency matrix, $A_{ij} = 1$ when there is an edge and $A_{ij} = 0$ when there is no edge. So that,

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{if } (v_i, v_j) \notin E. \end{cases}$$

Hence, if the graphs' edges are unweighted and undirectional, the corresponding adjacency matrix is binary and symmetric. On the other hand, the adjacency matrix of a digraph is generally asymmetric, since $A_{ij} \neq A_{ji}$ in the case of directed edges. Lastly, the diagonal degree matrix $D \in \mathbb{R}^{N \times N}$ is defined as the matrix where each entry on the diagonal is the row-sum of the adjacency matrix: $D_{ii} = \sum_j A_{ij}$, which is also symmetric for undirected graphs.

Graph Connectivity Whether a graph is connected or not determines if information propagation across all vertices of the graph is possible.

Convolutional Neural Networks operate on images and preserve the connectivity equivalent to that of a grid.



> A graph $G = (V, E)$ is said to be *connected* if there is a path between every pair of nodes in the graph. In other words, for any two nodes v_i and v_j, there exists a sequence of edges $e_1, e_2, \ldots, e_k \in E$ such that v_i and v_j are endpoints of this sequence.

Conversely, in a *disconnected* graph, there exist pairs for which no such path exists.

At the node level, degree centrality is a measure of the importance or influence of a node in a graph based on its connectivity.

> The *degree centrality* of a node measures the number of direct connections a node has. In an undirected graph, the degree $d(v_i)$ of a node v_i is simply the number of edges connected to it:
>
> $$deg(v_i) = \sum_j A_{ij} = \sum_j A_{ji}.$$
>
> In directed graphs, the *in-degree* and *out-degree* are defined as the number of incoming and outgoing edges, respectively:
>
> $$deg_{in}(v_i) = \sum_j A_{ji}, \quad deg_{out}(v_i) = \sum_j A_{ij}.$$

Intuitively, a node with high degree centrality is likely to have smaller shortest path distances to other nodes.

The *shortest path (graph geodesic) distance* between two nodes $v_i, v_j \in V$ in a weighted graph $G = (V, E)$, denoted $d_G(v_i, v_j)$, is the minimum total weight of any path connecting these nodes. Formally, for a path $P = (e_1, \ldots, e_k)$ where $e_i \in E$, we define:

$$d_G(v_i, v_j) = \min_{P \in \mathcal{P}_{ij}} \sum_{e_k \in P} w(e_k)$$

where $\mathcal{P}_{ij}$ is the set of all paths from v_i to v_j in G, and $w : E \to \mathbb{R}^+$ is the weight function assigning positive real numbers to edges.

If no path exists between v_i and v_j, we define the shortest path to be $d_G(v_i, v_j) = \infty$. For unweighted graphs, the distance equals the minimum number of edges in any path between the nodes. Note that the shortest path distance induces a metric space (V, d_G) over the vertex set of the graph G.

For optimization purposes alternative definitions of the distance between disconnected nodes may be more appropiate than using ∞.

The diameter of a graph is the longest shortest path between any two nodes in a graph.

The *diameter* diam(G) is defined as the maximum value of the shortest path distances between all pairs of nodes:

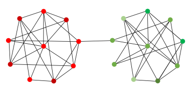
$$diam(G) = \max_{v_i, v_j \in V} d_G(v_i, v_j),$$

where $d(v_i, v_j)$ is the shortest path distance between nodes v_i and v_j.

Homophily and Heterophily We can assign class labels y_i to each node v_i. Most real-world graph datasets adhere to the principle of homophily, where connected nodes tend to belong to the same class. For example, in citation networks, similar research works cite each other. Homophily can be calculated as the fraction of intra-class graph edges:

It is also possible to assign labels at the graph or edge level.

Example highly homophilic graph.



$$h = \frac{1}{|E|} \sum_{(v_i, v_j) \in E} \mathbb{1}(y_i = y_j),$$

where $\mathbb{1}$ is the indicator function evaluating to one when the labels of adjacent nodes are equal. The homophily level h can take values between 0 and 1. We refer to graphs with low h values as being heterophilic or non-homophilic. Most classical GNN architectures rely on the implicit assumption that graph labels are homophilic.

7.2 Types of Graphs

Next, we discuss important types of graphs based on their connectivity structures, or *graph topology*. At one extreme, we can consider graphs that are completely disconnected, known as point clouds. These are actually common in many applications, such as remote sensing technology and surface reconstruction.

The term point cloud is often associated with points (or nodes) having coordinates in $\mathbb{R}^2$ or $\mathbb{R}^3$, while the term null graph is more commonly used in graph theory textbooks to refer to graphs without feature vectors.

However, in the GNN literature, point clouds do not necessarily have spatial coordinates.

A *point cloud* (or *null graph* N_N, where the subscript stands for $N = |V|$) is a graph $G = (V, E)$ whose edge set is the empty set $E = \emptyset$.

At the other end of the spectrum, we have complete graphs, which represent the maximum possible number of edges in a graph with N vertices, where every vertex is directly connected to every other vertex.

A *complete graph* is a graph in which every pair of distinct vertices is connected by a unique edge. A complete graph with N vertices is denoted K_N.

The ubiquitous attention mechanism in Transformers performs computations over a complete graph, where N is the number of tokens in the context window.

This means that in a complete graph, there are no disconnected components and all vertices are reachable from each other, with a graph geodesic distance equal to 1 for unweighted graphs.

A *bipartite graph* $G = (V, E)$ consists of a set of vertices V, which can be partitioned into two disjoint subsets V_1 and V_2, such that $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$, and a set of edges $E \subseteq \{\{u, v\} \mid u \in V_1, v \in V_2\}$, meaning that edges only connect vertices in V_1 to vertices in V_2.

In simpler terms, a bipartite graph is a graph in which the vertices can be divided into two disjoint sets, such that no two vertices within the same set are adjacent, and edges connect only vertices from different sets. Bipartite graphs are commonly used for modeling in recommendation systems and for matching products to users.

Paths and Cycles Next, we discuss paths and cycles as graph substructures.

A *path graph* is a graph where the vertices are arranged in a linear sequence, such that each vertex is connected to at most two others. A path graph with N vertices is denoted P_N.

Path graphs can represent linear sequences or chains in networks.

P_N consists of N vertices and $N - 1$ edges, where the endpoints (also called leaves) have degree 1, and all other vertices have degree 2. For instance, consider the vertex set $V = \{1, 2, \ldots, N\}$, where each vertex corresponds to an element of $\mathbb{N}$ and the edge set is $E = \{(v_i, v_{i+1}) \mid i \in \{1, 2, \ldots, N-1\}\}$, representing the connections between consecutive numbers. This construction discretizes the natural numbers by treating them as evenly spaced points on a line.

A cycle in a graph is a path that starts and ends at the same node, with all intermediate vertices being distinct. An acyclic graph is one that does not contain any cycles (or closed loops).

A *cycle graph* is a graph that consists of a single cycle, where each vertex is connected to exactly two others, forming a closed loop. A cycle graph with N vertices is denoted C_N.

The circular structure of a cycle graph can be used to represent periodic phenomena.

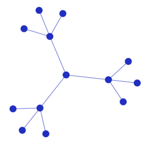
A *directed acyclic graph (DAG)* is a directed graph that contains no cycles. In a DAG, the edges have a direction, and there is no directed path that leads back to the starting node.

DAGs are often used to describe causality.

A *tree* is a connected, acyclic graph where there is exactly one path between any two nodes. It has $|V| - 1$ edges for $|V|$ vertices.

A directed tree is a type of DAG, but trees can also be undirected.

Trees have negative curvature and exhibit exponential volume growth.



Regular Graphs In many applications where the underlying graph connectivity is unknown, such as in latent graph inference and bioinformatics, one assumes the underlying graph to be regular.

A *regular graph* is a graph where every vertex has the same degree. If each vertex has degree k, the graph is called k*-regular*.

- The *null graph* N_N, which is 0-regular (no edges).
- The *cycle graph* C_N, which is 2-regular.
- The *complete graph* K_N, which is $(N-1)$-regular.
- *Cubic graphs*, a special class of 3-regular graphs, such as the Petersen graph.

The Petersen graph is a 10-vertex, 15-edge undirected graph that plays a prominent role in graph theory, often used as a key example or counterexample in various problems.

Geometric Graphs In geometric graphs nodes are represented as points in Euclidean space and their relationships are often defined based on distance or some other notion of geometric proximity according to the space's metric.

Proximity is used to infer the graph connectivity of molecules based on electron cloud images obtained through X-ray crystallography.

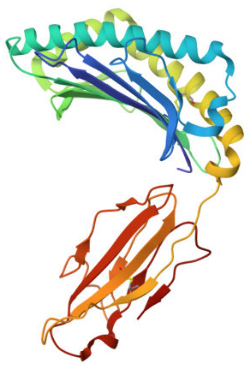


Figure 14: Geometric graphs can be used as mathematical abstractions of biomolecules.

A *geometric graph* $G = (V, E)$ is a graph where each node $v_i \in V$ is associated with a point in a geometric space, typically $\mathbb{R}^2$ or $\mathbb{R}^3$, and edges $(v_i, v_j) \in E$ are determined by the positions of the nodes.

Often, in geometric graphs we use the *unit disk graph* approach where edges $(v_i, v_j) \in E$ are included if the distance $d(v_i, v_j)$ between nodes v_i and v_j is less than or equal to a fixed threshold ϵ, i.e., $d(v_i, v_j) \leq \epsilon$.

An alternative approach is to use *k-nearest neighbor (k-NN) graphs*. In a k-NN graph, each node is connected to its k-closest neighbors in the geometric space, based on the distance metric $d(v_i, v_j)$. This method does not rely on a fixed threshold, but instead ensures that each node is connected to exactly k other nodes, that is, it is a k-regular graph.

k-NN type properties might be desirable if the graph's density is intended to remain consistent, as it can also prevent the occurrence of disconnected components. However, it imposes constraints on the graph's connectivity structure and may result in connections between nodes that are unreasonably distant.

7.3 Group Theory and Graphs

In many graph machine learning applications, it is important to preserve the structure of the data under reordering, since the numbering of the nodes is arbitrary to begin with. This is where symmetric groups and permutation-invariant aggregators come into play.

Let S be a set with $|S| = N$. The *symmetric group* of S, denoted by S_N, is the set of all bijections from S to itself:

$$S_N = \{\sigma : S \to S \mid \sigma \text{ is a bijection}\}.$$

A *permutation-invariant aggregator* is a function $\bigoplus : \mathcal{X}^N \to \mathcal{Y}$ that satisfies the condition

$$\bigoplus(x_1, x_2, \ldots, x_N) = \bigoplus(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(N)}),$$

for any permutation $\sigma \in S_N$, and $\mathcal{X}^N$ denotes the set of all ordered tuples of N elements from the set $\mathcal{X}$.

Common examples of permutation-invariant aggregators include summation $\sum_{i=1}^{N} x_i$, mean $\frac{1}{N}\sum_{i=1}^{N} x_i$, and maximum $\max_{i=1}^{N} x_i$, where x_i are features vectors associated to each node v_i as later discussed in Section 7.4.

Graph Homomorphisms Similar to group homomorphisms which allow us to relate equivalent groups that can be realized differently (Section 1.2), graph homomorphisms provide a mathematical framework for studying mappings between graphs that preserve their structural properties. This can be particularly relevant in the context of network compression, graph colorings, and GNN expressivity analysis.

A *graph homomorphism* is a mapping $F : V_G \to V_H$ between the vertex sets of two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ such that if $(v_i, v_j) \in E_G$, then $(F(v_i), F(v_j)) \in E_H$.

Intuitively, a graph homomorphism maps edges of G to edges of H, preserving the adjacency structure: if v_i and v_j are adjacent in G, their images $F(v_i)$ and $F(v_j)$ are adjacent in H. Note that in general, a homomorphism can map multiple vertices or edges of G onto a single vertex or edge in H. This enables the simplification (or *coarsening*) of graph structures while retaining connectivity properties.

A *graph isomorphism* is a bijective mapping $F : V_G \to V_H$ between the vertex sets of two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ such that $(v_i, v_j) \in E_G$ if and only if $(F(v_i), F(v_j)) \in E_H$.

Graph isomorphisms are a specific class of graph homomorphisms in which the mapping must be bijective, and the edge-preservation condition is bidirectional.

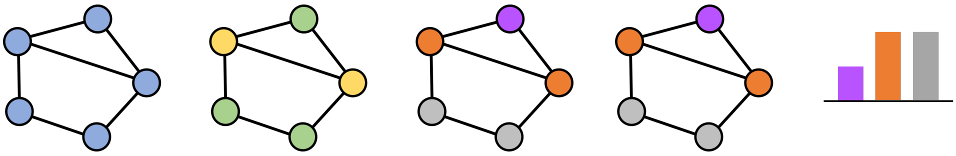


Figure 15: The Weisfeiler-Lehman (WL) test is a method used to determine whether two graphs are isomorphic by iteratively refining node labels based on their neighborhoods.

Examples of Graph Homomorphisms

- Consider a cycle graph C_6 with six vertices and a complete graph K_3. A homomorphism $F : V_{C_6} \to V_{K_3}$ exists, where vertices of C_6 are mapped to vertices of K_3 in a repeating pattern.
- For bipartite graphs, any homomorphism maps vertices in one partition to one set of vertices in the target graph and the other partition to the other set.
- Let P_{11} be a path graph with eleven vertices, and C_{10} be a cycle graph with ten vertices. A homomorphism $F : V_{P_{11}} \to V_{C_{10}}$ exists where each vertex of P_{11} is mapped to a vertex of C_{10}, and edges of P_{11} are mapped to edges of C_{10}. Note that in this case the vertices at the start and end of the path graph would be mapped (or collapsed) to a single vertex.

7.4 Vector Fields on Graphs

Although so far our discussion has centered on graphs in terms of their connectivity structure, in practical scenarios and particularly in the context of Geometric Deep Learning, we primarily deal with graphs that have node attributes. Next, we consider graphs where each node has associated feature vectors and introduce relevant notation.

A *feature vector* x_i at node v_i is a D-dimensional vector that represents the characteristics or attributes of the node in the graph.

These vectors are organized into a matrix $X \in \mathbb{R}^{N \times D}$ for all nodes $N = |V|$ in the graph. In the following expression, each entry x_{ij} represents the j-th feature of node i:

$$X = \begin{bmatrix} -x_1^\top- \\ -x_2^\top- \\ \vdots \\ -x_N^\top- \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}.$$

Equivalently, linking this discussion back to Section 4.1, we can define the feature vector field F as a mapping from the graph domain (nodes in the graph) to $\mathbb{R}^D$, where D is the number of features for each node:

$$F : V \to \mathbb{R}^D, \quad F(v_i) = x_i \in \mathbb{R}^D, \quad \forall v_i \in V.$$

The Graph Laplacian The Laplacian plays a key role in analyzing graph structures, particularly in spectral graph theory.

> The *graph Laplacian* matrix L for a graph $G = (V, E)$ is defined as:
>
> $$L = D - A,$$
>
> where A and D are the adjacency and degree matrices of the graph, respectively.

For undirected graphs, the graph Laplacian is symmetric and positive-semidefinite.

The quadratic form associated with the graph Laplacian can be written as:

$$x^\top L x = x^\top (D - A)x = \sum_{i=1}^{n} d_i x_i^2 - \sum_{(v_i,v_j)\in E} w_{ij} x_i x_j = \frac{1}{2} \sum_{(v_i,v_j)\in E} w_{ij}(x_i - x_j)^2,$$

where $w_{ij} = w(e_{ij}) = w((v_i, v_j))$ is the weight of the edge (v_i, v_j), and x_i and x_j are the feature values at nodes v_i and v_j, respectively. Note that this is effectively computing a gradient-like quantity over the graph, which measures the smoothness of the vector field over the graph and is analogous to the Dirichlet energy in continuous settings. It is often referred to as the *graph Dirichlet energy* or simply the *Dirichlet energy on a graph*.

The Dirichlet energy is the continuous setting is the quadratic functional $\langle f, \Delta f \rangle = \langle \nabla f, \nabla f \rangle$.

> The *normalized graph Laplacian* matrix L_{norm} is defined as:
>
> $$L_{\text{norm}} = I - D^{-1/2} A D^{-1/2},$$
>
> where I is the identity matrix, A is the adjacency matrix, and D is the degree matrix.

The form above has several useful properties: the eigenvalues of L_{norm} lie in the range $[0, 2]$ and the multiplicity of the eigenvalue 0 corresponds to the number of connected components in the graph.

The multiplicity of the eigenvalue refers to the number of times a specific eigenvalue appears in the spectrum of a matrix.

Message-Passing on Graphs For GNNs, we say we are *learning a signal over a graph*, where the graph structure guides the flow of information between nodes. We often couple the graph on which the signal is defined with the computational graph of the artificial neural network.

More concretely, a message passing GNN layer l over a graph G is computed as

$$x_i^{(l+1)} = \phi\Big(x_i^{(l)}, \bigoplus_{j \in \mathcal{N}(v_i)} \psi(x_i^{(l)}, x_j^{(l)})\Big),$$

where ψ and ϕ are non-linear functions, and $\bigoplus$ is an aggregation function, which must be permutation-invariant. The above equation constraints the information flow for each layer to local neighbourhoods.

Graph Theory in Geometric Deep Learning. Graph theory plays a central role in Geometric Deep Learning, particularly in the context of Graph Neural Networks, which are designed to learn signals over graph structures. The underlying graph domain serves as a geometric prior, typically assuming that connected nodes share similar features. Graph Neural Networks have been applied to diverse areas, including social networks, recommendation systems, and bioinformatics, for both supervised learning and generative modeling.

Transformers perform attentional message passing over a fully connected graph. Alternatively, one can interpret the attention scores as 'discovering' the underlying graph.

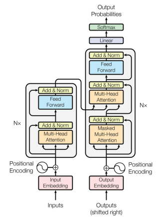


Figure 1: The Transformer - model architecture.