1 Disambiguations and definitions

1.1 Naming conventions

1.1.1 Basic notions

Let's recall the naming conventions of basic notions.
A *function* $f : E \to F$ maps objects $x \in E$ to objects $y \in F$, as $y = f(x)$.
Its *definition domain* $\mathcal{D}_f = E$ is the set of objects onto which it is defined. We will often just use the term *domain*.
We also say that f is *taking values* in its *codomain* F.
The *image per* f of the subset $U \subset E$, denoted $f(U)$, is $\{y \in F, \exists x \in E, y = f(x)\}$.
The *image of* f is the image of its domain. We denote $\mathcal{I}_f$.
A vector space E, which we will always assume to be finite-dimensional in our context, is defined as $\mathbb{R}^n$, and is equipped with pointwise addition and scalar multiplication.

1.1.2 Signals

A *signal* s is a function taking values in a vector space. In other words, a signal can also be seen as a *vector* with an *underlying structure*, where the vector is composed from its image, and the underlying structure is defined by its *domain*.

For example, images are signals define on a set of pixels. Typically, an image s in RGB representation is a mapping from pixels p to a 3 dimensional vector space, as $s_p = (r, g, b)$. The underlying structure of images are a grid as the pixels are arranged as such.

1.1.3 An example : graph Signals

A *graph* $G = (V, E)$ is defined as a set of nodes V, and a set of edges $E \subseteq \binom{V}{2}$.
The words *node* and *vertex* will be used equivalently, but we will rather use the first.
A *graph signal*, or *graph-structured signal* is a signal defined on the nodes of a graph, for which the underlying structure is the graph itself. A *node signal* is a signal defined on a node, in which case it is a *node embedding* in a vector space.
Although this is rarely seen, a signal can also be defined on the edges of a graph, or on an edge. We then coin it respectively *dual graph signal*, or *edge signal / edge embedding*.
Graph-structured data can refer to any of these type of signals.
A dataset of signals is said to be *static* if all its signals share the same underlying structure, it is said to be *non-static* otherwise.
For image datasets, being non-static would mean that the dataset contains

images of different sizes or different scales. For graph signal datasets, it would mean thats the underlying graph structures of the signals are different.
The point in specifying that objects of a dataset of a machine learning task are signals is that we can hope to leverage their underlying structure.

1.2 Regularity of a domain

In this subsection, we explain the notion of *regularity* relatively to the context of deep learning and convolutional representations. Deep learning on regular domains refer to deep learning of signals with a regular underlying structure; although being on irregular domains means that signals are defined over an irregular underlying structure.

Definition 1 () *Regular domain*

Let consider a signal $s : \mathcal{D} \to s(\mathcal{D})$. We assume $\mathcal{D}$ to be finite.

1.3 Disambiguation of the title

This thesis manuscript is entitled *Deep Learning on Irregular or Unstructured Data*.
Deep learning on unstructured data means that the input data of the deep learning models are not signals, for example they are just objects or feature vectors, without any underlying structure *a priori*. On the other hand, the input data of deep learning models on *irregular data* are signals defined on irregular domains.

1.4 Theoretical results on regularity and convolutions

2 Datasets

2.1 Tasks

3 Goals

3.1 Invariance

Definition 2

In order to be observed, invariances must be defined relatively to an observation. Let's give a formal definition to support our discussion.
...

3.2 Methods