COLLEGE MATHSTIC
DOCTORAL BRETAGNE
BRETAGNE OCEANE

Université Bretagne Sud

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ BRETAGNE SUD

ÉCOLE DOCTORALE N° 644
Mathématiques et Sciences et Technologies de l'Information et de la Communication en Bretagne Océane
Spécialité : *Informatique*

Par

Paul BERG

Contributions to Representation Learning in Computer Vision and Remote Sensing

Thèse présentée et soutenue à Vannes, le 13/12/2024
Unité de recherche : Institut de Recherche en Informatique et Systèmes Aléatoires - IRISA
Thèse N° : 703

Rapporteurs avant soutenance :

Dino IENCO — Directeur de Recherche, INRAE
Loïc LANDRIEU — Directeur de Recherche, Université Gustave Eiffel

Composition du Jury :

Président·e :	David PICARD	Senior Research Scientist, Université Gustave Eiffel
Examinateur·trice·s :	Ewa KIJAK	Maître de conférence, Université Rennes 1
Directeur de thèse :	Nicolas COURTY	Professeur des universités, Université de Bretagne Sud
Co-encadrant de thèse :	Minh-Tan PHAM	Maître de conférence, Université de Bretagne Sud

Acknowledgements

Premièrement, je tiens à remercier mes encadrants : Minh-Tan et Nicolas pour leur confiance durant cette thèse. Ce fut un plaisir de travailler à vos côtés pendant ces trois années. Merci Minh-Tan pour tes conseils avisés et ta disponibilité. Merci Nicolas pour m'avoir introduit à des sujets comme le transport optimal.

Je remercie ensuite grandement Dino Ienco et Loïc Landrieu pour avoir acceptés d'être rapporteur de ma thèse ainsi que David Picard et Ewa Kijac pour leur participation au jury. Merci également d'avoir fait le déplacement jusqu'à Vannes pour la soutenance.

Merci à Guillaume et Renan, sans qui ces trois années auraient été bien différentes. J'ai apprécié la découverte du transport optimal à vos côtés.

Je remercie également les membres du laboratoire que j'ai côtoyés durant ces années, Baddie, Corentin, Iris, Manal, Marion, Hugo, les djeun's de la team baby, Aimi, Étienne, Pierre, Léo. Merci également aux permanents de l'équipe Obélix.

Thank you An, I appreciated the scientific collaboration with you and our discussions on research. Thank you Björn for being such a big help with point clouds. I would be eager to collabore again in the future.

Je veux aussi remercier les chercheurs.e.s avec qui j'ai eu l'occasion de collaborer pendant ma thèse : Clément Bonet, Baki Uzun, Björn Michele, Hoang-Ân Lê, François Septier et Lucas Drumetz. Ces collaborations auront été très formatrices dans mon parcours de chercheur.

Merci également à Deise, Minh-Tan et à Sébastien pour l'encadrement lors de mon stage de fin d'étude au sein de l'équipe Obélix. Vous m'avez introduit au monde de la recherche.

Je tiens aussi à remercier mon groupe d'amis de l'UTC avec qui nous avons réussi à maintenir un lien. Nos vacances auront été des bulles d'air pendant ces années.

Merci à mes parents, mes sœurs, ma famille pour votre support et votre enthousiasme qui m'ont largement aidé durant ces trois années.

Table of Contents

1 Introduction **9**

1.1 Motivations 9

1.2 Overview of Contributions 15

2 Background **18**

2.1 Self-Supervised Representation Learning 19

2.2 Hyperbolic Representation Learning 29

2.3 Optimal Transport 35

3 Optimal Transport for Self-Supervised Learning **46**

3.1 Transporting between Samples and Features 47

3.2 Hyperspherical Uniformity using Spherical Sliced Wasserstein 56

3.3 Robust Self-Supervised Object Detection 62

3.4 Conclusion and Discussion 68

4 Multi-Modal Contrastive Learning **70**

4.1 Uni-Modal Self-Supervised Learning in Remote Sensing 71

4.2 Multi-Modal Self-Supervised Learning 76

4.3 Multi-Modal Supervised Contrastive Learning 81

4.4 Conclusion and Discussion 87

5 Horospherical Learning with Hierarchical Prototypes **89**

5.1 Hyperbolic Classification using Horospheres 90

5.2 Hierarchical Initialization 92

5.3 Experiments 97

5.4 Conclusion and Discussion 103

6 Conclusion **104**

6.1 Overview of Contributions 104

6.2 Perspectives 106

A Appendix **108**
A.1 List of Publications 108
A.2 Appendix of Chapter 4 109
A.3 Appendix of Chapter 5 109

B Introduction (Français) **116**
B.1 Motivations 116
B.2 Aperçu des contributions 122

Bibliography **125**

Glossary

Acronyms

AHC Average Hierarchical Cost.
AP Average Precision.
BYOL Bootstrap Your Own Latent.
CAFOs Concentrated Animal Feeding Operations.
CNN Convolutional Neural Network.
COOT Co-Optimal Transport.
DenseCL Dense Contrastive Learning.
EMA Exponentially Moving Average.
GPU Graphics Processing Unit.
GW Gromov-Wasserstein.
HIS Hyperbolic Image Segmentation.
KL Kullback-Leibler.
LF Landfill.
ML Machine Learning.
MMCR Maximum Manifold Capacity Representation.
MoCo Momentum Contrast.
OT Optimal Transport.
R&Ts oil Refineries and petroleum Terminals.
RGB Red Green Blue.
S1 Sentinel-1.
S2 Sentinel-2.
SGD Stochastic Gradient Descent.
SSL Self-Supervised Learning.
SSW Spherical Sliced Wasserstein.
SupCon Supervised Contrastive.
SW Sliced Wasserstein.

SwAV	Swapping Assignments between Views.
UOT	Unbalanced Optimal Transport.
VAE	Variational Auto-Encoder.
ViT	Vision Transformer.
WCE	Weighted Cross-Entropy.
WWT	Waste Water Treatment.

Notations

$T \sim \mathcal{T}$	T is a sample drawn from distribution $\mathcal{T}$.
$T_{\mathrm{p}}\mathcal{M}$	Tangent space at point p of manifold $\mathcal{M}$.
$U(a, b)$	Set of couplings with marginals a and b, respectively.
δ_x	Dirac measure at x.
$\exp_{\mathrm{p}}(\cdot)$	Exponential map at point p.
$\|\cdot\|$	Euclidean norm.
$\|\cdot\|_\star$	Nuclear norm.
$\langle \cdot, \cdot \rangle$	Euclidean inner product.
$\langle \cdot, \cdot \rangle_F$	Frobenius inner product.
$\mathbb{S}^d$	Unit-hypersphere of d dimensions.
$\mathcal{H}_d^c$	Hyperbolic space of d dimensions and of curvature c.
$1_{\{\cdot\}}$	Indicator function.
$\mathbf{cat}(x_1, \ldots, x_n)$	Concatenation operator.
$\{1, \ldots, N\}$	Set of natural numbers included in the interval $[1, N]$.

Chapter 1

Introduction

Contents

1.1 Motivations 9

1.2 Overview of Contributions 15

1.1 Motivations

While vision appears as a natural process for humans, computers have no inherent notion of image understanding. In computers, images are generally represented as matrices of pixels with an extra dimension if multiple channels are present. This representation may be convenient for some image analyses but is mainly driven by the nature of image capture sensors which are positioned in a grid pattern. As such, computer programs have no straightforward way to reason about the visual nature of objects and items present in an image. Therefore, the computer vision research community has developed techniques to help computers perform better image analyses and classification by teaching computers to learn better representations than the baseline pixel grids.

The importance of learning representations in computer vision cannot be overstated. Traditional hand-crafted features, which were once the cornerstone of computer vision, have been largely surpassed by learned representations, which offer greater flexibility, adaptability, and performance. The rise in popularity of deep learning techniques has further solidified the significance of learned representations, enabling models to learn complex patterns and hierarchies of features from large labelled datasets.

Nowadays, deep learning methods are ubiquitous in computer vision. Deep networks are fed many annotated examples in order to solve image understanding tasks, sometimes reaching or surpassing human level performance. However, annotating large amounts of images remains a complex and time-consuming task. Depending on the type of annotations, experts may be required to annotate data for the precise visual task they want to

solve. As an example, the ImageNet (Russakovsky et al., 2015) dataset requires a many human-years long effort to fully annotate its more than 14 millions images. Training deep learning models on such datasets is a costly and compute intensive process but the resulting models are not only able to perform well in the dataset of choice but they also serve as a good initialization point for training models on datasets which have similar visual features or across different tasks (Huh et al., 2016). Indeed, as part of its classification task, the model is able to extract features which are effective at describing the visual content of the image, especially in early layers of the model. As a result, its weights can be used as is with great results for initially unrelated tasks such as anomaly detection in images (Defard et al., 2021) or to measure a perceptual distance metric between images (Zhang et al., 2018).

Unfortunately, such datasets and pre-trained weights are not always available when the datasets at hand are not composed of RGB natural images or the model architecture chosen is uncommon. Also, these datasets can contain biases that are not suitable for real world usage depending on how they have been put together. For example, out of the 1000 classes contained in the ImageNet dataset, more than 100 are dog breeds which begs the question about whether or not the model weights can be used for other vision tasks without adaptation. Therefore, representation learning methods not based on regular supervised learning have started to emerge as appealing alternatives to the so-called ImageNet initialization. An increasingly popular set of methods known as Self-Supervised Learning (SSL) have started gaining momentum in the computer vision research community. They are built on the idea that instead of using classification as a task to generate useful representations, pretext tasks can be created only from non-annotated data using cleverly crafted objectives. Therefore, a model can be trained to solve such pretext tasks without relying on human annotations for the dataset as annotations for the pretext tasks are computed by the training algorithm itself. Hence the naming given to these methods; self-supervision. Since these methods do not rely on outside supervision to learn meaningful representations which can then be used in a series of downstream tasks with more success than supervised weights initialization. The rapid growth in popularity of these methods can be visualized in Figure 1.1. This self-supervision trend has started accelerating in parallel to the rise of deep learning in computer vision with seminal works such as those of Gidaris et al. (2018), Noroozi and Favaro (2016), Wu et al. (2018), and Zhang et al. (2016), among others. More recently, a popular branch of Self-Supervised Learning called joint-embedding methods has been introduced. In these methods, a network is

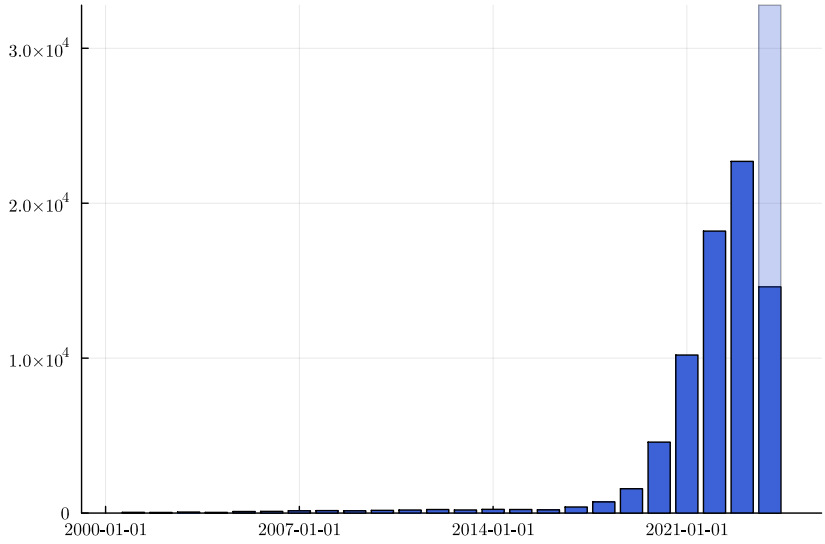


Figure 1.1: Number of publications mentioning "Self-Supervised Learning" published each year since 2000 (source: Google Scholar, the 11/06/2024). Recent years have seen an increase in the number of self-supervision related publications. Note that the year 2024 is not fully completed and has therefore been extrapolated.

trained to produce similar representations for artificially augmented views of the same image. These methods also include a technique to prevent a representational collapse in which the trained model only produces a single constant representation for all input images which is input-invariant but has maximum alignment between artificial views of the same sample. Among these joint-embedding methods, we can distinguish the so-called contrastive methods (Chen et al., 2020a; He et al., 2020) where for each sample, a set of negative samples is pushed further away in the latent space. Non-contrastive methods, on the other hand, resort to other techniques to prevent this collapse such as using a stop gradient operator on on branch (Chen & He, 2021; Grill et al., 2020) or information maximisation (Bardes et al., 2022; Ermolov et al., 2021; Zbontar et al., 2021). Finally, with the rising popularity of Vision Transformer (ViT) (Dosovitskiy et al., 2020), masked auto-encoders (He et al., 2021) offer a promising avenue in Self-Supervised Learning due to the simplicity of the reconstruction objective.

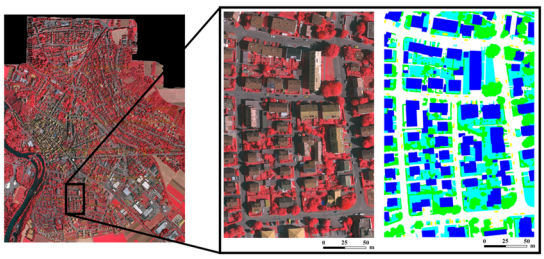
However, in specific domains such as remote sensing, datasets often have specificities compared to natural images datasets. Representation learning methods should be able to leverage these peculiarities in order to yield a more data efficient training process. Moreover, methods from the computer vision literature sometimes cannot be applied to

the remote sensing domain without an adaptation. Indeed, images from remote sensing have different characteristics than natural images. For example, in dense tasks such as semantic segmentation, objects and regions often are small in scale compared to the overall image. There is also a wide variety of spatial resolutions depending on the chosen sensors. This can lead to remote sensing images being very large and therefore computationally expensive to process for off-the-shelf methods without first being pre-processed. The regularity at which satellites observe the earth is another factor which must be taken into account when starting an earth observation project. There is often a trade-off between having high resolution images but at a low regularity and having low resolution images with a higher capture rate. This trade-off is emphasized by the fact that satellite sensors do not all capture the same spectral bands. Some missions only capture RGB and near infrared bands while others capture large ranges resulting in hyperspectral images with hundreds of channels. As such, remote sensing practitioners are often motivated to generate multimodal datasets in order to accumulate the benefits of multiple image providers. Examples of multiple remote sensing datasets can be seen in Figure 1.2.

In a multi-modal dataset, a sample is composed of multiple readings from different sensors. These sensors can vary widely and offer truly heterogeneous datasets which require specific methods because more general computer vision methods are not applicable. In multimodal datasets, samples can be co-registered. That is, the capture from the different sensors have been spatially aligned using the geographical coordinates available with each sample. As such, a sample of such a multimodal dataset is composed of multiple images from the same location.

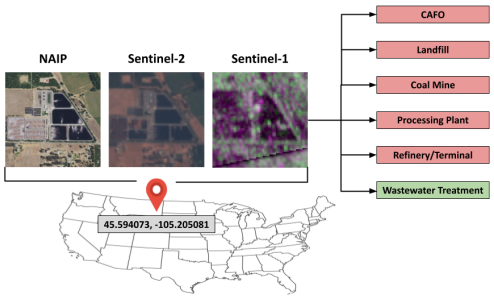
Another trait that image datasets, being from specific computer vision domains or remote sensing, is having an associated hierarchy. Hierarchical datasets are composed of ground truth labels which can be organised in a hierarchical fashion by grouping labels in semantically meaningful super-classes. Datasets taxonomies can be exploited to produce more data-informed representations and help during the training process. In tasks such as semantic segmentation, the hierarchical organisation of labels comes naturally. In remote sensing, labels can oftentimes be grouped in different semantic super-classes which can lead to the creation of a useful hierarchy. Similarity in such hierarchical spaces for two samples should ideally yield a higher similarity in visual features in order for the hierarchy to help improve performance in computer vision tasks. Classification on hierarchical datasets is still an under explored area of the computer vision literature but has a high potential to improve vision systems. One can imagine that while classification errors cannot be

(a) The VEDAI car detection dataset (Razakarivony & Jurie, 2016).



(b) The Vaihingen semantic segmentation dataset (Rottensteiner et al., 2012).

(c) The EuroCrops satellite image time series segmentation dataset (Schneider et al., 2021) with hierarchical labels.



(d) The Meter-ML multi-modal satellite scene classification dataset (Zhu et al., 2022).

Figure 1.2: Examples of remote sensing datasets created for diverse tasks. Images are taken from the respective papers/dataset.

avoided, it is less serious for a classification model to predict a class which is close in the hierarchical sense than to predict a class which is very far in the hierarchy and therefore has no semantic similarity whatsoever. In order to evaluate such systems, hierarchical metrics should be used instead of the baseline performance metrics such as accuracy. A hierarchical metric should be able to evaluate if predictions made by a model are close in the hierarchy to the ground-truth instead of just measuring if right ground-truth class has been predicted. Research in hierarchical classification is also linked to the task of few shot learning since the introduction of new classes can be prepared using the hierarchy if the hierarchical relationship is known between the new classes and the previously present classes. The hierarchical nature of a dataset is also a good motivation to explore different embedding spaces for representations.

These peculiarities encourage the computer vision and remote sensing communities to develop proper methods to leverage or handle datasets specificities. These contributions

can vary in scope and depth. They sometimes require architectural changes compared to the more general methods or to design bespoke training objectives which truly embrace the nature of data in downstream tasks.

One possible area of improvement for specific datasets is to put into question the default embedding space of latent representations which is the Euclidean space. Indeed, traditional deep learning methods only operate on Euclidean embeddings in part because of its well defined distance metric and natural numerical representation. Recently, this status quo has been challenged with the introduction of alternatives to Euclidean spaces for deep representations such as hyperspherical or hyperbolic deep learning models. These alternative embedding spaces are a promising research direction to improve the performance of models in specific tasks where they are naturally suited due to their inherent structure. For example, since hyperbolic spaces are known for their capacity to embed tree-like structure with minimal distortion (Sarkar, 2011), they become a seducing alternative to Euclidean spaces when it comes to classification over hierarchical data.

Finally, in representation learning, several successful works have proposed which consider samples as part of a distribution (Caron et al., 2020; Robinson et al., 2020; Zbontar et al., 2021). This distribution-based point of view considers that the encoder transforms samples from the dataset's distribution in another distribution which should exhibits discriminative properties with respect to the downstream task's objective. For example, in contrastive learning, we can consider the set of positives as being drawn from a local neighborhood among the unknown ground-truth distribution from the larger dataset whereas negatives should faithfully represent the entire universe of possible images (*i.e.* being drawn from the ground-truth distribution). Therefore, we can consider the use of tools to measure discrepancies between distributions when designing representation learning objectives. Optimal Transport is a popular tool to measure distances between distribution. Its formulation and its many variants make of Optimal Transport a versatile solution that machine learning practitioners increasingly reach for.

In this context, this thesis starts with the goal of creating more efficient representation learning procedures in both computer vision and remote sensing. We consider the following main objectives:

- Investigate how Self-Supervised Learning can be improved by looking at it from a distributional perspective.

- Contribute to reducing the need for ground-truth annotations in the process of

image representation learning.

- Propose representation learning methods tailored to computer vision and remote sensing datasets specificities such as multi-modality or hierarchical structure in order to train encoders more efficiently.

We contribute to these aspects in several ways. Our contributions range from proposing improvements to Self-Supervised Learning by leveraging tools such as Optimal Transport to proposing a novel framework for multi-modal representation learning with applications to remote sensing. In Section 1.2, we outline the content of this manuscript chapter by chapter.

1.2 Overview of Contributions

In this thesis, we contribute representation learning in the context of computer vision and remote sensing downstream tasks, ranging from scene classification to object detection and semantic segmentation. To that end, we consider tools from the machine learning such as Optimal Transport. This manuscript is organised in the following manner:

- In Chapter 2, we present the technical background required to understand the rest of the document. Therefore, state-of-the-art Self-Supervised Learning methods are presented as an introduction to the field of SSL for image representation learning. This chapter also contains background information on hyperbolic spaces and non-Euclidean deep learning which will be needed for later chapters. We also introduce Optimal Transport theory and its computational aspects. Along with general optimal transport, variants of OT are also presented such as entropic OT, Unbalanced Optimal Transport (UOT) and methods for solving OT in incomparable spaces.

- Chapter 3 introduces our work to leverage Optimal Transport for self-supervised representation learning. Our contributions range from proposing a variant of contrastive learning based on the transport plan between samples in a Self-Supervised Learning pipeline to a non-contrastive hyperspherical uniformity objective used to train a self-supervised model. We also introduce a modification to Dense Contrastive Learning which works by modeling images as empirical distributions from their patch representations specifically for the dense downstream tasks such as semantic segmentation and object detection. While some of the presented research is

currently unpublished, Section 3.2 is based on research that was published in the following conference article:

> "*Spherical Sliced-Wasserstein*", C. Bonet, **P. Berg**, N. Courty, F. Septier, L. Drumetz, and M.-T. Pham, in International Conference on Learning Representations, 2023.

- Chapter 4 is dedicated to the problem of multi-modal representation learning problem which is commonly encountered in the field of remote sensing. As such, we propose a multi-modal contrastive framework for self-supervised pre-training of models over multi-modal datasets. Moreover, we introduce a multi-modal supervised contrastive loss that allows us to improve the task specific finetuning when comparing to the baseline cross-entropy finetuning. This chapter is based on the following publications:

  > "*Self-Supervised Learning for Scene Classification in Remote Sensing: Current State of the Art and Perspectives*", **P. Berg**, M.-T. Pham, and N. Courty, Remote Sensing, vol. 14, 16, p. 3995, 2022.

  > "*Joint Multi-Modal Self-Supervised Pretraining in Remote Sensing: Application to Methane Source Classification*", **P. Berg**, M.-T. Pham, and N. Courty, in IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2023, pp. 6624 - 6627.

  > "*Multimodal Supervised Contrastive Learning in Remote Sensing Downstream Tasks*", **P. Berg**, B. Uzun, M.-T. Pham, and N. Courty, IEEE Geoscience and Remote Sensing Letters, 2024.

- Chapter 5 is devoted to the problem of classification over hierarchical datasets. In these datasets, ground-truth labels can be organised in a hierarchy which provides an interesting prior knowledge to improve the classification performance over such datasets. Indeed, depending on how the hierarchy is built, we find that classes close by in the hierarchy often exhibits similar visual features which can help in the task of image classification. To that end, we leverage hyperbolic spaces which are Riemannian manifolds with interesting properties to embed trees and hierarchies.

Our contribution is two-folds, we first introduce a hyperbolic classification layer based on ideal prototypes. Secondly, we propose a hierarchically-informed scheme to initially position these prototypes. The content of this chapter has been the subject of the following article:

> "*Horospherical Learning with Smart Prototypes*", **P. Berg**, B. Michele, M.-T. Pham, L. Chapel, and N. Courty, British Machine Vision Conference (BMVC) 2024.

- In Chapter 6, we summarize our contributions and conclude the manuscript by discussing potential future research directions following our contributions. This is the final chapter of this manuscript.

Chapter 2

Background

Contents

2.1 Self-Supervised Representation Learning 19

2.2 Hyperbolic Representation Learning 29

2.3 Optimal Transport 35

In computer vision, the starting point to solving a task is often an image which is represented as a set of pixels in an n-dimensional color space. Working directly on pixels in order to solve a high-level task such as image classification is difficult. This is also true when operating with remote sensing images. Therefore, the research community has worked on developing higher level image representations that would be more discriminative for the task at hand. While these image features were first hand-crafted, it then became clear that learning-based methods were more efficient at creating optimal representations from a training dataset. The task of extracting discriminative representations from images can be resumed as projecting an image to a lower dimensional space where solving the task becomes easier. As an example, the optimal representation space for a classification task is actually the one-hot encoded label space. Since deep learning based methods require a differentiable objective, models create representations in continuous spaces and a final classification step is used to create predictions in label space. This continuous representation space is often naturally chosen as the Euclidean space, but recently, alternative spaces have emerged as interesting options. Among them, hyperbolic spaces provide interesting properties, specifically for tasks related to hierarchical representations. This is in part due to their capacity to embed trees with minimal distortion compared to other embedding spaces of choice in representation learning.

Data distributions can often be encountered in representation learning. Indeed, one can consider a series of representations in latent space as being an empirical distribution drawn from an unknown ground-truth distribution. Modeling such problems under the distribution prism can be beneficial instead of considering each sample on its own. For

example, mini-batches in a deep learning pipeline can be seen as an empirical distribution drawn from the ground-truth distribution of all images considered within the current task. As such, we reach for tools to measure and compare empirical distributions which have proven successful in deep learning applications. One such tool is the Optimal Transport (OT) problem which can be used to measure distances between distributions. In Section 2.3, we introduce to the required background in OT theory which will later be leveraged in the rest of the document.

2.1 Self-Supervised Representation Learning

The supervised deep learning-based state-of-the-art methods in computer vision often rely on large amounts of annotated images in order to learn relevant image features. Nonetheless, big datasets are very time and labor intensive to annotate. One of the biggest annotated image recognition datasets is ImageNet (Russakovsky et al., 2015) with more than 14 million training images and it has taken several human years to annotate. As a practical approach in many vision-based applied fields, exploiting supervised models pre-trained on ImageNet is a common way to boost the performance of deep neural networks when performing transfer learning or fine-tuning on smaller domain-specific image data. Regarding the concept of transfer learning, using model weights pre-trained on the ImageNet dataset can improve performance over randomly initializing network weights (*i.e.* training from scratch) (Huh et al., 2016). The pre-trained weights exhibit better representation capabilities than random parameters, in particular in the first layers of the network. Still, deeper layers should be trained on the domain-specific data in a process called fine-tuning so that the network is able to extract features relevant to the new task and can actually perform predictions. In spite of that, for certain tasks and datasets, there exists no such pre-trained models. In order to learn discriminative representations without relying so much on data annotations, Self-Supervised Learning (SSL) methods were introduced.

In this section, we briefly review the most significant state-of-the-art self-supervised methods, mostly proposed within the machine learning and computer vision communities. Without loss of generality, we divide these methods into four categories including generative, predictive, contrastive and non-contrastive SSL. We note that in the literature, contrastive and non-contrastive approaches can be regrouped into a single category sometimes referred to as joint-embedding approaches. Our choice is to distinguish these two,

without any loss of generality. For a more thorough background review of self-supervised approaches, readers are invited to dedicated review papers such as Jing and Tian (2020), Liu et al. (2021), and Ohri and Kumar (2021).

Generative Methods. A common pretext task is to reconstruct the input image after compression by using an auto-encoder. To minimize the reconstruction loss, the model has to learn to compress all significant information from the image into a latent space with a lower dimension, using the first network's component called encoder. Then, a second network's component named decoder tries to reconstruct the image from the latent space. Denoising auto-encoders (Vincent et al., 2008) have also been proved to provide robust image representations by learning to remove artificial noise from images. The added noise prevents the network from learning the identity function. Variational Auto-Encoder (VAE) (Kingma & Welling, 2014) improve over the auto-encoder framework by encoding the parameters of the latent space distribution. They are trained to minimize both the reconstruction error and an additional term minimizing the Kullback-Leibler divergence between a known latent distribution often picked as the unit centered Gaussian distribution and the one produced by the encoder. This regularization over the latent space allows to sample from the generated distribution. More recently, the use of vision transformers (Dosovitskiy et al., 2020) has enabled the development of large masked auto-encoders (He et al., 2021) working at a patch level instead of pixel-wise to reconstruct entire patches with only 25% of visible patches. This reconstruction task produces robust image representations by appending a class token to the sequence of patches or simply by using a global average pooling on all the patch tokens.

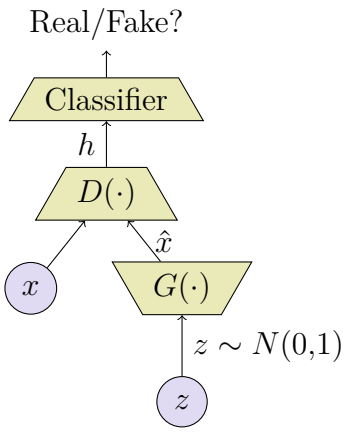


Figure 2.1: The GAN architecture (Goodfellow et al., 2014) for use in generative self-supervised representation learning. The representation used is h, the last discriminator activation before a binary classification in fake/real labels.

Last but not least, another primordial unsupervised generative learning model that has been significantly explored in the literature is the Generative Adversarial Network (GAN) (Goodfellow et al., 2014). This architecture and many of its extensions attempt to generate new data from a random noise with the aim of mimicking real data. GANs are trained in an adversarial minimax two-player game where one network called the generator

$G(\cdot)$ learns to transform the random noise $z \sim \mathcal{N}(0, 1)$ to synthetic data $\hat{x}$ which tries to follow the original data distribution. In an adversary approach, a second network called the discriminator $D(\cdot)$ learns to classify between images from the generator and real images from the original data set (see Figure 2.1). The output score of the discriminator is 1 when it is confident that the input image is coming from the real data distribution and 0 for images created by the generator. This adversarial objective can be written as:

$$\min_G \max_D \frac{1}{N} \sum_{i=1}^{N} \log(1 - D(G(z_i))) + \frac{1}{M} \sum_{i=1}^{M} \log(D(x_i)), \tag{2.1}$$

where $z \sim \mathcal{N}(0, 1)$ is a set of N random noise vectors and $x \sim p_{data}(x)$ is a set of M real images. With such training, the discriminator learns to identify details in the real images in order to discriminate between real and fake images. To this end, a common way to produce image level representations from GAN-based models is to use a pre-trained discriminator as a feature extractor as proposed in Radford et al. (2016).

Predictive Methods. The second category of SSL methods involves models which are trained to predict the effect of an artificial transformation of the input image. Such an approach is motivated by the intuition that predicting the transformation requires learning relevant characteristics of semantic objects and regions within the image. By pre-training a model to predict the relative position of two image patches, (Doersch et al., 2015) boosted the performance of a model against a random initialization and to get closer to the performance of the initialization with ImageNet pre-trained weights in iconic computer vision datasets.

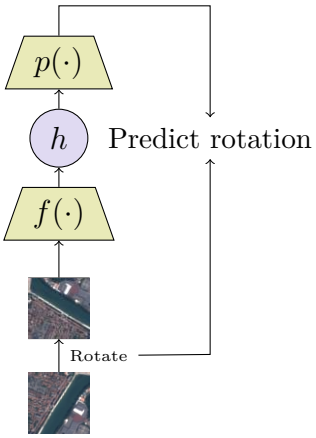


Figure 2.2: In RotNet (Gidaris et al., 2018), a random rotation is applied to input images and the model is then tasked to classify which rotation was applied. The model is composed of an encoder f whose output representations h are used by a predictor p to classify the random rotation.

Other possible predictive pretext tasks have been proposed to learn representations. One of them is the image colorization proposed in (Zhang et al., 2016). In such an approach, the input image is first converted to its grayscale version. Then, an auto-encoder is trained to colorize the grayscale version back to the color one by minimizing the mean squared error between the reconstruction and the original image. The feature repre-

sentations provided by the encoder are considered for later downstream tasks. Another well-known predictive SSL method is the RotNet (Gidaris et al., 2018) which proposes to train a model to predict the rotation that was randomly applied to the input image (see Figure 2.2 for an illustration). Solving this rotation prediction task requires the model to extract meaningful features that help to understand the semantic content of the image. Similarly, another SSL model is developed to solve a jigsaw puzzle in Noroozi and Favaro (2016) to predict relative positions of image partitions that were previously shuffled. Also, by considering several types of augmentations, the Exemplar CNN (Dosovitskiy et al., 2014) is trained to predict the augmentations applied to images. In Exemplar CNN, the authors proposed several augmentation classes including cropping, rotation, color jittering and contrast modification.

By fulfilling one of these aforementioned pretext tasks, a SSL model is able to learn in-depth representations of image content. Nevertheless, depending on the pretext task and on the dataset, the network will not necessarily be able to perform well on all downstream tasks. As an example, predicting random rotations of an image as in RotNet (Gidaris et al., 2018) would not perform particularly well on a remote sensing dataset, since the orientation of objects is not as strictly important as in object-centric datasets.

Contrastive Methods. Another way to yield effective image representations is to force the features of multiple views of an image to be similar. The final representations are then invariant to the augmentations used to create the different image views. Yet, without proper care, the network can converge to a constant representation which is independent of the input image and which satisfies the invariance constraint (*i.e.* the collapsing problem (Jing et al., 2021)).

A common solution to learn diverse representations with the above objective while preventing the collapsing issue is to use a contrastive loss. Such a loss function attempts to force the model to discriminate representations between views from the same image (*i.e.* positives) and those from different images (*i.e.* negatives). In other words, it aims to obtain similar feature representations for positive pairs while pushing away representations for negative pairs. Within this family of methods, the simplest objective is the triplet loss (Dong & Shen, 2018) from which a model is trained to provide a smaller distance between representations of an anchor and its positive than the distance between that anchor and a random negative (see Figure 2.3 for an illustration). The triplet loss function

can be formulated as follows.

$$\mathcal{L}_{\text{triplet}} = \max(\|f(x) - f(x^+)\| - \|f(x) - f(x^-)\| + m, 0), \tag{2.2}$$

where x^+ and x^- are the positive and negative samples of the anchor x, respectively; $f(.)$ is an embedding function and m represents a margin parameter. This idea is then motivated by the authors in (Wu et al., 2018) who propose to train an image classifier with as many labels as training samples, which creates well-performing representations in downstream tasks.

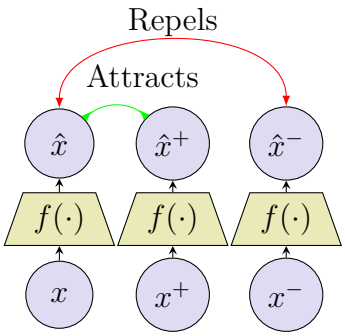


Figure 2.3: The triplet loss (Dong & Shen, 2018) is used to learn discriminative representations by learning an encoder which is able to discriminate between negative and positive samples.

SimCLR (Chen et al., 2020a), one of the most popular SSL approaches, proposes a form of contrastive representation learning. For each image in the training batch, two different views are created by sampling random augmentations. These augmented images are then fed into the representation model followed by a predictor network whose goal is to project the representation onto a D-dimensional hypersphere. The whole model is trained to maximize the cosine similarity between a representation z and its positive counterpart z^+ (coming from the same original image) and to minimize the similarity between z and all the other representations in the batch z^-, resulting in the following term:

$$l(z, z^+, z^-) = -\log \frac{\exp(\langle z, z^+ \rangle / \tau)}{\sum_{z' \in z^- \cup \{z^+\}} \exp(\langle z, z' \rangle / \tau)}, \tag{2.3}$$

where $\langle x, y \rangle$ is the dot product between x and y; and τ is a temperature parameter scaling the sharpness of the similarity distribution. The complete loss called normalized temperature cross entropy (NT-Xent) is computed as follows:

$$\mathcal{L}_{\text{NT-Xent}} = \frac{1}{2N} \sum_{z,z^+,z^-} l(z, z^+, z^-). \tag{2.4}$$

where N is the number of images in the batch.

Since the representations are normalized before the NT-Xent loss is computed, the loss acts only on the direction of the features within the D-dimensional hypersphere as illustrated in Figure 2.4 and not on their norm. This loss acts as a proxy to maximize the mutual information between the two views leading to representations that are both independent to style and informative only about the content of the image.

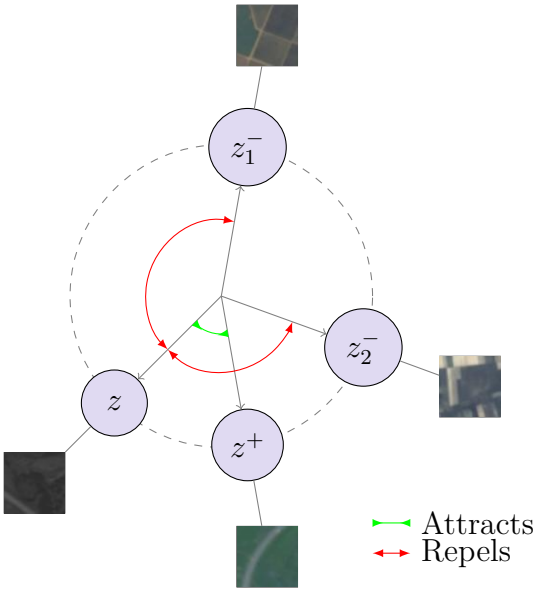


Figure 2.4: Illustration of contrastive loss on the 2-dimensional unit sphere with two negatives (z_1^- and z_2^-) and one positive (z^+) samples from the EuroSAT (Helber et al., 2019) dataset.

In parallel to SimCLR, the Momentum Contrast (MoCo) (He et al., 2020) method is proposed to allow for smaller batches with the same effective number of negative samples when computing the contrastive loss. It uses a sample queue to provide more negative samples per batch (cf. Figure 2.5) as well as a momentum encoder whose weights are updated using an Exponentially Moving Average (EMA) of the main encoder's weights. For each batch, the oldest samples in the queue are discarded and replaced with the new positives. Other methods such as Swapping Assignments between Views (SwAV) (Caron et al., 2020), cluster representations to a common set of prototypes and learn to match views to consistent clusters between positive pairs. To prevent that all representations

converge to the same clusters (*i.e.* collapsing), the entropy-regularized optimal transport plan (Peyré, Cuturi, et al., 2019) between the representations and the clusters is used. Finally, the loss minimizes the cross-entropy between the optimal assignments of one branch with the predicted distribution for the other branch. This method is one example of successfully leveraging optimal transport for self-supervised representation learning. Using an optimal transport plan ensures that the matching is complete between samples from the batch and the clusters. Said differently, every cluster will have the same quantities of samples (*i.e.* the same amount of probabilistic mass) matched with it, leading to a uniform distribution among clusters. In practice, contrastive methods often require a large batch size to provide enough negative samples to the loss and to prevent collapsing representations. In the case of SwAV, this ensures that the mini-batch is faithful to the actual ground-truth distribution and that the samples-to-clusters transport plan can be relied upon.

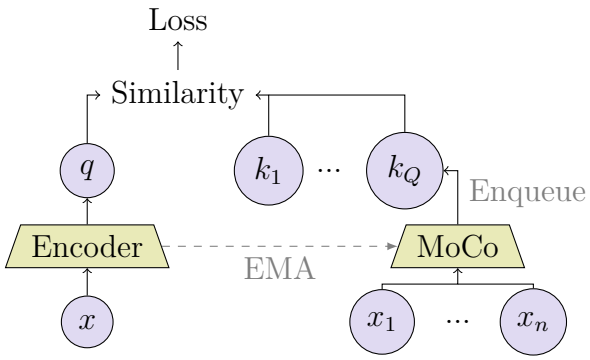


Figure 2.5: In Momentum Contrast (He et al., 2020), a queue of Q samples is built using a momentum encoder (right) whose weights are updated as an Exponentially Moving Average (EMA) of the main encoder's weights (left). Therefore at each step, only the main encoder's weights are updated by back-propagation. The similarity between the queue samples and the encoded batch samples is then used in the contrastive loss (cf. Equation 2.4).

The above joint-embedding methods tend to create more general representations than predictive methods. However, depending on the choice of augmentations, they may not perform well on all downstream tasks. For example, if a model produces the same representations for two different crops of the same image, it has then removed any spatial information about the image and is likely to perform worse in a task which requires this spatial information like semantic segmentation or object detection. To prevent this effect, Dense Contrastive Learning (DenseCL) (Wang et al., 2021) was proposed. It applies the

contrastive loss on patch-level representations instead of at the image-level. This allows the contrastive model to learn less spatially invariant representations.

Non-Contrastive Methods. As part of joint-embedding learning approaches, other methods manage to train self-supervised models without using contrastive components in their loss. We categorize them as non-contrastive methods. Bootstrap Your Own Latent (BYOL) (Grill et al., 2020) uses a teacher-student network configuration. In a teacher-student configuration, the student network is trained to match the output (or the features) of the teacher network. Such an approach is often used in knowledge distillation (Hinton et al., 2015) where the teacher and student models have different architectures (*e.g.* the size of the the student model is much smaller than that of the teacher). In BYOL, the teacher network's weights are defined as an EMA of the student network's weights. The encoders f^A and f^B are followed by two projector networks g^A and g^B whose output is used to compute the training loss. Only the student encoder f^A is then kept to extract image-level representations. A predictor network is also added on top of the student projector to prevent collapsing representations (see Figure 2.6) by adding further asymmetry between the two branches. SimSiam (Chen & He, 2021) uses two identical networks and also adds a predictor network on one of its branches. Since the two branches share the same weights, a stop gradient is used asymmetrically in the loss which maximizes the pairwise alignments between positive pairs. DINO (self-DIstillation with NO labels) (Caron et al., 2021) uses a student-teacher transformer architecture referred to as self-distillation where the teacher is defined as an EMA of the student network's weights. The student is then trained to extract the same predictions as the centered and sharpened output of the teacher network for a given positive pair.

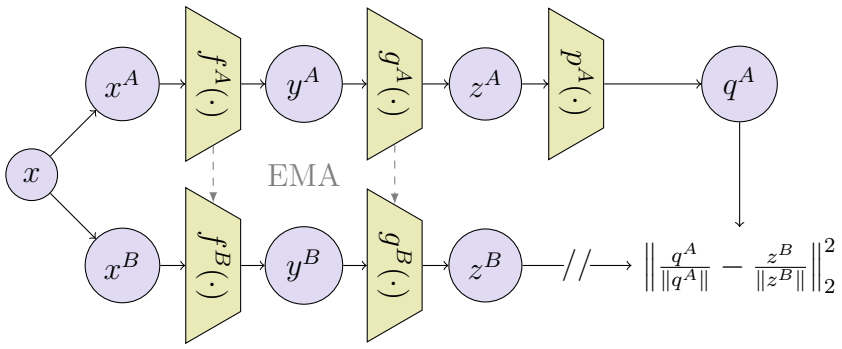


Figure 2.6: The non-contrastive BYOL (Grill et al., 2020) architecture which uses a student A and a teacher B pathways to encode the images. The teacher's weights are updated using an EMA of the student's weights. The online branch is also equipped with an additional network p^A called the predictor.

Without requiring separate weights for each branch of the teacher-student pipeline as in BYOL or SimSiam, another non-contrastive learning framework named Barlow Twins (Zbontar et al., 2021) is proposed based on the information bottleneck theory (Saxe et al., 2019). Such a method maximizes the mutual information between two views by increasing the cross-correlation of their corresponding features provided by two identical networks while removing redundant information in these representations. The principle is based on the information bottleneck. The loss function of Barlow Twins is the following:

$$\mathcal{L}_{\text{Barlow-Twins}}(\mathcal{C};\theta) = \sum_{i=1}^{N}(1-\mathcal{C}_{ii}^2) + \lambda \sum_{i=1}^{N}\sum_{j\neq i}\mathcal{C}_{ij}^2, \tag{2.5}$$

where $\mathcal{C}$ represents the cross-correlation matrix which is computed as follows:

$$\mathcal{C}_{ij} = \frac{\sum_b z_{bi}^A z_{bj}^B}{\sqrt{\sum_b (z_{bi}^A)^2}\sqrt{\sum_b (z_{bj}^B)^2}}. \tag{2.6}$$

where z^A and z^B are the outputs of two identical networks fed with the two views of an image.

Recently, a method using Variance, Invariance, Covariance regularization (VICReg) (Bardes et al., 2022) has been proposed to improve this framework. Unlike Barlow Twins, the loss terms are independent for each branchexcept for the invariance which explicitly maximizes alignment between positive pairs. This enables non-contrastive multi-modal pre-training between text and image pairs by relying on a different regularization for each pathway.

Another promising non-contrastive method is called Maximum Manifold Capacity Representation (MMCR) (Yerxa et al., 2023). Based on the manifold capacity theory, which evaluates the probability of finding a separating hyperplane for P manifolds embedded in D dimensions. It groups the spherical representations z_i^k of K views of the same images in centroids, one for each sample i. As such, these centroids have a norm which is smaller than one. A matrix of these centroids is then built to construct a basis in the embedding space.

$$C_i = \frac{1}{K}\sum_{k=1}^{K} z_i^k \tag{2.7}$$

The MMCR objective tries to maximize the nuclear norm of this matrix. The nuclear norm, also known as the trace norm, is the sum of the singular values of the input matrix. This has the effect of distributing representations around the hypersphere as well

as grouping the representations of views of the same sample together since centroids are pushed to the surface of the unit-hypersphere. Indeed, as the centroids' norms get closer to 1, all views from the samples converge to the same representation. The MMCR is defined as:

$$\mathcal{L}_{\text{MMCR}}(C;\theta) = -\|C\|_\star = -\sum_{i=1}^{\min(d,N)} \sigma_i(C), \tag{2.8}$$

where $\sigma_i(C)$ is the ith singular value of C. Since the nuclear norm is a convex envelope for the rank of C, therefore maximizing the nuclear norm effectively tries to make the centroids representations orthogonal to each other.

About Benchmarking Self-Supervised Methods. As seen in this section, self-supervised models generate representations from the data samples. These representations are made to be as discriminative as possible for a given downstream task. However, without using labels it is difficult to evaluate the quality of said representations. One can measure the effectiveness of the representation by performing an image retrieval task on the validation set only where the model can perform inference from the representations only and then select the top-k most similar images in the set. Other clustering metrics could be used such as the Rand index (Rand, 1971) or the Silhouette score (Rousseeuw, 1987). In practice, computer vision has converged on classification benchmarks on top of the representations. That is, once the model is pre-trained in a self-supervised fashion, a classifier is trained to classify the representations in the different classes relevant to the downstream task. This classification can be declined in multiple ways but the most popular methods are using a k-nearest neighbor classifier, training a linear classifier or fully fine-tuning the model to perform classification with an added linear layer. Adding a linear classifier on top of the frozen self-supervised encoder is often referred to as the linear evaluation protocol (Caron et al., 2021; Chen et al., 2020a) and is commonly used to measure the linear separability of the learned representations. However, self-supervised models which perform less effectively in the linear protocol are not necessarily worse image encoders in other benchmarks as is the case for masked auto-encoders (He et al., 2021) for example.

This two-step process of first pre-training and then learning a classifier makes it harder to evaluate the performance of self-supervised models during their training. As such, certain models are trained in a self-supervised fashion while a linear classifier is learned on top in a process called online linear probing (Garrido et al., 2023). This process

removes the need for a costly linear training and evaluation after the pre-training while offering a good proxy for the classification performance that would be reached under the linear evaluation protocol.

Another technique to diminish the computational impact of self-supervised models evaluation but also to evaluate their data efficiency is to perform the linear evaluation or fine-tuning under a limited data regime by using only a subset of the training data labels available. This setting often highlights the performance gain provided by Self-Supervised Learning compared to a single supervised training on the limited dataset.

Current Limits. As powerful pre-trained self-supervised models are released publicly (Oquab et al., 2023), their weights are used for a number of downstream tasks and become the de-facto initialization for many model architectures. However, pre-trained weights are not always available for uncommon modalities and architectures. This requires practitioners to pre-train from scratch on their respective tasks. Moreover, current methods from the computer vision literature often do not leverage the additional information which can be present in some datasets. For example, multimodal datasets contain multiple modalities for a single sample and since the methods presented in this section only operate on a single modality, they are not able to exploit the multiple modalities. As such new methods or extensions need to be developed in order to take into account these multiple modalities. Another example is that most current joint-embedding methods are projecting to the hypersphere before computing their variant of a self-supervised objective. Compared to Euclidean representations, this has the advantage of preventing the variance of representations from going to infinity. However, certain downstream tasks could benefit from using different embedding spaces. With the right formulation, recent works have successfully used hyperbolic spaces (Franco et al., 2023; Ge et al., 2023) as an embedding space for computing a self-supervised objective. In the next section, we present the background in hyperbolic representation learning and discuss why it can be a more appropriate choice of embedding spaces for tasks such as hierarchical classification.

2.2 Hyperbolic Representation Learning

While representation learning is often synonymous with Euclidean representations, other embedding spaces have recently emerged as appealing alternatives. Indeed, embedding samples into a different metric space can improve the representation capabilities (Bronstein et al., 2017). We have already seen in Subsection 2.1 that hyperspheres are often

adopted as the de-facto embedding space in Self-Supervised Learning in part due to the so-called cosine similarity which serves as a measure of alignment between data samples. Other notable embedding spaces include the cone of Positive Semi-Definite matrices (*e.g.* (Pennec, 2020)) used for covariance matrices or symmetric spaces such as the Siegel space (López et al., 2021) that naturally embed graphs. Recently, hyperbolic spaces have been gaining momentum in the research community. Hyperbolic spaces are Riemannian manifolds with a negative curvature. Interestingly, the hyperbolic distance metric is suited to embed trees with minimal distortion (Sarkar, 2011) which makes them a natural choice as an embedding space to perform representation learning on a variety of tasks. Hyperbolic spaces have successfully been used in a number of text (Dhingra et al., 2018; Ganea et al., 2018a; Nickel & Kiela, 2017; Tifrea et al., 2018), data classification (Cho et al., 2019; Fan et al., 2024), and vision (Atigh et al., 2022; Dhall et al., 2020; Ermolov et al., 2022; Ge et al., 2023; Ghadimi Atigh et al., 2021; Mettes et al., 2024) related tasks as an alternative to Euclidean or other embedding spaces.

A number of hyperbolic neural network layers have been proposed (Peng et al., 2021) to replace their Euclidean counterparts such as linear layers (Ganea et al., 2018b), convolutional layers (Shimizu et al., 2020) or even entire hyperbolic networks (van Spengler et al., 2023) where all activations are in hyperbolic space. In particular, hyperbolic representations show promising results in hierarchical tasks such as image classification (Chang et al., 2021), action recognition (Long et al., 2020; Surís et al., 2021) or image segmentation (Atigh et al., 2022) where label hierarchies are well-defined. Recently, hyperbolic spaces have been used in self-supervised representation learning by defining a hierarchy between objects and scenes (Ge et al., 2023) or for self-paced learning (Franco et al., 2023) leveraging the innate structure of the space to interpret the radius as a measure of uncertainty.

For a more detailed background about Hyperbolic spaces, we refer the reader to the books of Brannan et al. (2011) and Loustau (2020). Multiple models exist to represent hyperbolic objects. One of the most popular hyperbolic model in machine learning is that of the Poincaré ball which is defined as the points included in an hypersphere of radius $\frac{1}{\sqrt{c}}$:

$$\mathcal{H}_d^c = \{x \in \mathbb{R}^d, \|x\|_2^2 < 1/c\}, \tag{2.9}$$

where $c > 0$ refers to the curvature. Note that in the case where $c = 0$, we recover $\mathcal{H}_d^0 = \mathbb{R}^d$. In the Poincaré ball model, the hyperbolic distance between two points $x, y \in \mathcal{H}_c^d$ is defined

as:

$$d_{\mathcal{H}}(x,y) = \frac{1}{\sqrt{c}}\cosh^{-1}\left(1 + 2c\frac{\|x-y\|_2^2}{(1-c\|x\|_2^2)(1-c\|y\|_2^2)}\right). \tag{2.10}$$

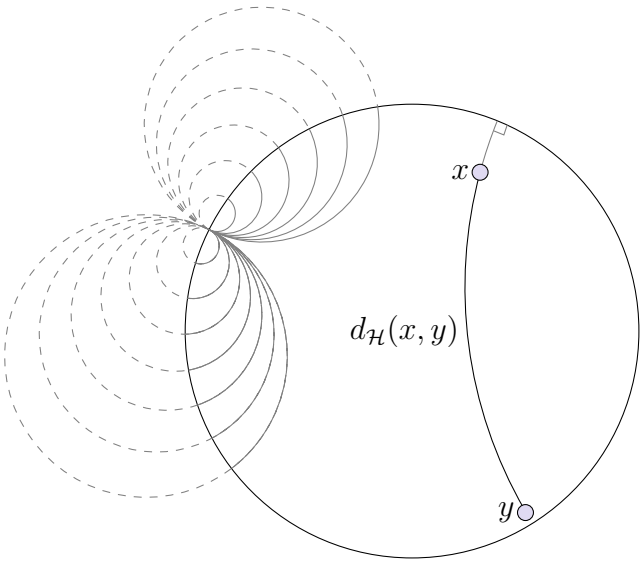


Figure 2.7: The Poincaré disk $\mathcal{H}_2^1$ and the geodesic between points x and y. $d_{\mathcal{H}}(x,y)$ describes an arc drawn for a Euclidean circle orthogonal to the disk. On the left, a set of geodesics passing through the same point at the infinity.

The hyperbolic geodesics analogous to Euclidean straight lines correspond to arcs between points drawn from Euclidean circles orthogonal to the boundary of the Poincaré ball as well as all diameters of the ball. An example of such geodesic line can be seen in Figure 2.7.

In order to project between Euclidean and hyperbolic space, one has to resort to exponential and log maps which project from, respectively to, the tangent space $T_p\mathcal{H}_d^c$ at a defined point $p \in \mathcal{H}_d^c$. An example of the exponential map on the hypersphere can be seen in Figure 2.8.

The tangent space at any point of the Poincaré ball is the Euclidean space of the same dimension $\mathbb{R}^d$, which means that points in Euclidean space can be projected

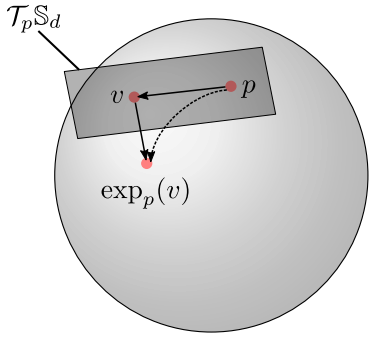


Figure 2.8: Example of the exponential map $\exp_p(\cdot)$ on a hypersphere $\mathbb{S}^d$, in this case the tangent space of a point p on the sphere is the tangent plane at point p, $T_p\mathbb{S}^d = \{x; \langle p, x - p\rangle = 0; x \in \mathbb{R}^{d+1}\}$.

on the Poincaré ball using the exponential map $\exp_p^c$:
$\mathbb{R}^d \to \mathcal{H}_d^c$ which is defined as such:

$$\exp_p^c(v) = p \oplus_c \left(\tanh\left(\sqrt{c} \frac{\lambda_p^c \|v\|_2}{2} \right) \frac{v}{\sqrt{c}\|v\|^2} \right), \tag{2.11}$$

where $\oplus_c$ corresponds to the Möbius addition, a transformation analogous to vector addition in Euclidean space but in hyperbolic space and $\lambda_p^c = 1/(2 - c\|p\|)$ is the conformal factor at point p.

Reciprocally, to project from the manifold to the tangent space at a point, the logarithm map $\log_p^c : \mathcal{H}_d^c \to T_p\mathcal{H}_d^c$ can be used. In the Poincaré ball model, it is defined as:

$$\log_p^c(x) = \frac{2}{\sqrt{c}\lambda_x^c} \tanh^{-1}(\sqrt{c}\|-x \oplus_c y\|) \frac{-x \oplus_c y}{\|-x \oplus_c y\|} \tag{2.12}$$

The exponential (*resp.* logarithm) map to project from (*resp.* to) the tangent space of the origin can be written in a more simplified manner:

$$\exp_0^c(v) = \tanh(\sqrt{c}\|v\|) \frac{v}{\sqrt{c}\|v\|}, \tag{2.13}$$

$$\log_0^c(x) = \tanh^{-1}(\sqrt{c}\|y\|) \frac{y}{\sqrt{c}\|y\|}. \tag{2.14}$$

The origin is often used as the point of reference when projecting from Euclidean representations to hyperbolic ones.

The Möbius addition between $x, y \in \mathcal{H}_d^c$ is defined as:

$$x \oplus_c y = \frac{(1 + 2c\langle x, y\rangle + c\|y\|^2)x + (1 - c\|x\|^2)y}{1 + 2c\langle x, y\rangle + c^2\|x\|^2\|y\|^2}. \tag{2.15}$$

When the Poincaré ball is used as an embedding space in computer vision, it is generally only applied in the last layers of a model whereas the previous layers operate on regular Euclidean representations, where an exponential map is used to project samples in Euclidean space to hyperbolic space. But it is also possible to perform all operations in the hyperbolic space by using hyperbolic operators for each operation in the backbone model (van Spengler et al., 2023).

Ideal Boundary. For each geodesic ray $\gamma : \mathbb{R} \to \mathcal{H}_d^c$, defined from two distinct points x and y as in Figure 2.7 such that $\gamma(t_x) = x$ and $\gamma(t_y) = y$. This geodesic can

be described from the points $\gamma(-\infty)$ and $\gamma(+\infty)$ located at the infinity of the hyperbolic space. The set of all such points at infinity is referred to as the Gromov boundary or ideal boundary. In the Poincaré ball model, the ideal boundary corresponds to the surface of the corresponding ball $\{\mathrm{p}; \|\mathrm{p}\|_2 = \frac{1}{\sqrt{c}}, \mathrm{p} \in \mathbb{R}^d\}$. In Figure 2.7, ideal points for the geodesic between x and y are located at the intersection between the geodesic and the unit-circle. The geodesic is perpendicular to the Poincaré ball at two ideal points associated with each geodesic. While ideal points are not strictly in the Hyperbolic space, they can be used as a direction with respect to which a measure of alignment can be computed with points in hyperbolic space.

Busemann function. Busemann functions (Busemann, 1955) are a measure of alignment between ideal points located at infinity and points embedded in hyperbolic space where regular distance metrics such as $d_{\mathcal{H}}$ would fall short (*i.e.* be infinite). It can be seen analogously to the Euclidean dot product which returns a measure of alignment in Euclidean space. While Busemann functions can be derived for all Riemannian manifolds of negative curvature, we discuss here Busemann functions in hyperbolic spaces. Let γ be a geodesic ray in the manifold $\mathcal{M}$, such that $d_{\mathcal{M}}(\gamma(0), \gamma(t)) = t, \forall t \geq 0$. The Busemann function $B_{\mathrm{p}} : \mathcal{M} \to \mathbb{R}$ with respect to an ideal point p located at the ideal boundary of $\mathcal{M}$ is defined as:

$$B_{\mathrm{p}}(x) = \lim_{t \to +\infty} (t - d_{\mathcal{M}}(x, \gamma(t))), \quad x \in \mathcal{M}. \tag{2.16}$$

The value of the Busemann function can be seen as a form of measure of alignment with respect to the ideal point p. In the Poincaré ball model, the Busemann function has a closed-form solution which is defined as:

$$B_{\mathrm{p}}(x) = \frac{1}{\sqrt{c}} \log \left(\frac{\|\mathrm{p} - \sqrt{c}x\|_2^2}{1 - c\|x\|_2^2} \right), \tag{2.17}$$

for $x \in \mathcal{H}_d^c$ and $\mathrm{p} \in \mathbb{S}^{d-1}$. Intuitively, the Busemann function measures the coordinates of x along the geodesic arc which contains x and has p as an endpoint.

Hyperbolic Gyroplans. Ganea et al. (2018b) introduced a hyperbolic counterpart to Euclidean hyperplanes in order to perform classification between samples embedded in hyperbolic space. These gyroplanes can be used to classify samples in the Poincaré ball. Each gyroplane is parameterized by a point $p \in \mathcal{H}_d^c$ and a normal vector $w \in T_p\mathcal{H}_d^c$ embedded in the tangent space at point p. The gyroplane decision boundary is defined as:

$$H_{p,w} = \{z \in \mathcal{H}_d^c, \langle -p \oplus_c z, w \rangle = 0\}\,. \tag{2.18}$$

From a set of gyroplans, one can define an hyperbolic classifier by computing logits from the hyperbolic distance between the gyroplans and a given sample $z \in \mathcal{H}_d^c$:

$$\zeta_y(z) = \frac{\lambda_{p_y}^c \|w_y\|_2^2}{\sqrt{c}} \sinh^{-1}\left(\frac{2\sqrt{c}\langle -p_y \oplus_c z, w_y\rangle}{(1 - c\|\langle -p_y \oplus_c z, w_y\rangle\|_2^2)\|w_y\|_2^2}\right), \tag{2.19}$$

and then a class membership probability can be computed by computing the softmax probability from the different logits ζ_y. An example of the binary decision boundary for a single gyroplan can be seen in Figure 2.9.

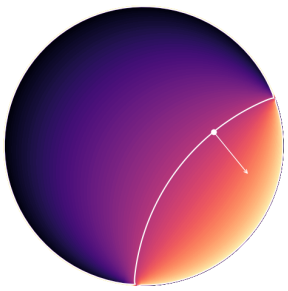


Figure 2.9: An hyperbolic gyroplan which is parameterized by a point p in the Poincaré disk and a normal vector $\vec{w} \in T_p\mathcal{H}_d^c$.

Optimization in Riemannian Manifolds. Models like the hyperbolic gyroplan have parameters which are embedded in the hyperbolic space. When optimizing these parameters using Stochastic Gradient Descent, one has to be careful since there are no guarantees that the parameters stay on their respective manifold after applying the gradient update. Therefore, optimization on Riemannian manifolds is modified to use Riemannian gradient descent (Bonnabel, 2013) where the parameters are projected back onto their manifold after the weight update. The update of weight θ at each iteration is defined as:

$$\theta_{t+1} = \exp_{\theta_t}(-\eta_t \nabla \mathcal{L}(X_t; \theta_t)), \tag{2.20}$$

where η_t is the step size, also known as the learning rate and $\nabla\mathcal{L}(X_t; \theta_t)$ can be viewed as the Riemannian gradient of the loss function at point X_t with respect to θ_t. Oftentimes, X_t will be a mini-batch sampled from the larger training dataset. In practice, since the

exponential map can be expensive to compute, one can replace it with a retraction function R_{θ_t}, that is, an estimation of the exponential map with the following property on a point v embedded within the tangent space $T_\theta\mathcal{M}$: $d_{\mathcal{M}}(R_\theta(tv), \exp_\theta(tv)) = O(t^2)$. For example, a retraction of choice for the hyper-spherical manifold $\mathbb{S}^{d-1}$ is the Euclidean addition followed by a projection on the sphere. Other gradient-based optimization methods have been adapted to the non-Euclidean setting such as Riemannian Adam by Bécigneul and Ganea (2018).

2.3 Optimal Transport

In machine learning, statistical distributions are ubiquitous. Instead of considering each sample or point as a single element, modeling deep learning problems from an empirical distribution point of view can be beneficial. Datasets or mini-batches can be modeled as empirical measures over the real unobservable distributions.

Optimal transport is a tool that can be used to compare distributions and as such has gained popularity in a number of machine learning problems such as domain adaptation (Courty et al., 2016), generative modeling (Arjovsky et al., 2017) and image comparisons (Rubner et al., 1998). In this section, we provide the reader with background information about Optimal Transport (OT) which will be used later in the document. For more detailed information, we refer the reader to the book of Villani et al. (2009). For the computational aspects of OT, more details can be found in the book of Peyré, Cuturi, et al. (2019).

2.3.1 General Optimal Transport

The Optimal Transport problem consists in finding the optimal way to transport mass from a source distribution to a target distribution. It was first introduced in Monge (1781). In Monge's formulation, a particle x from the source distribution displaces its mass $\mu(x)$ following the so-called Monge map $T(x)$ towards a particle of the target distribution ν where μ and ν are supported on Ω_μ and Ω_ν respectively. The goal is to find an optimal transport map such that it minimizes the total amount of cost to displace mass:

$$\inf_{T_{\#}\mu=\nu} \int_{\Omega_\mu} c(x, T(x))\mu(x)dx, \tag{2.21}$$

where $c : \Omega_\mu \times \Omega_\nu \to [0, +\infty]$ is a measure of distance between particle x and y from the source and target distributions respectively and $T_{\#}\mu$ denotes the push-forward of μ by T. When μ and ν are empirical distributions both containing the same amount of particles, then the Monge problem is akin to solving the linear assignment problem.

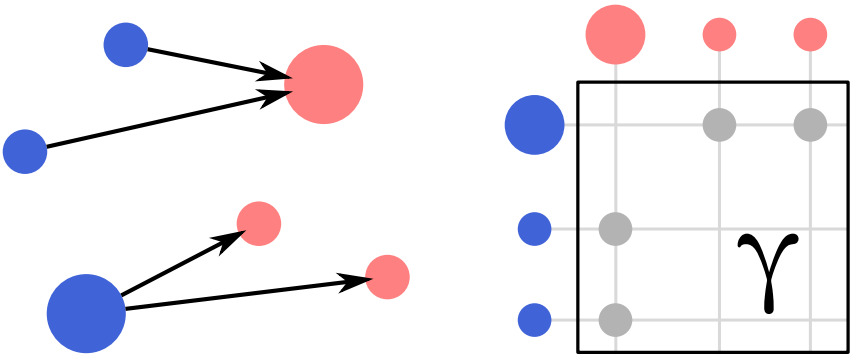


Figure 2.10: Simple discrete Optimal Transport between source and target empirical distributions each composed of three samples.

OT was then generalized by Kantorovich (1942) which relaxed the initial formulation by allowing the transport map to transport a source particle mass to multiple target particles.

$$\inf_{\gamma \in U(\mu,\nu)} \int_{\Omega_\mu \times \Omega_\nu} c(x, y)\gamma(x, y)dxdy, \tag{2.22}$$

where $U(\mu, \nu)$ is the set of joint probability measures on $\Omega_\mu \times \Omega_\nu$ with marginals μ on Ω_μ and ν on Ω_ν respectively.

While the formulations in Equations 2.21 and 2.22 operate over continuous support, in machine learning, the problem is often treated as a transport between discrete distributions where each atom in the source and target distributions can correspond to a data sample. This is because the gathered data can be thought of as being sampled from the ground truth distributions which are not observable. An example of such a problem between two point clouds supported on the plane can be seen in Figure 2.10. For a and b two vectors in the simplex of size n and n', such that $\sum_i a_{i=1}^n = 1$ and $\sum_{i=1}^{n'} b_i = 1$, they describe the probabilistic mass given to each sample of the source and target distributions respectively. Given a cost matrix C of size $N_a \times N_b$ where C_{ij} is a measure of distance between the ith sample of the source and the jth sample of the target distribution, in the discrete setting, the OT problem can be written as a linear program:

$$L_C(a, b) = \min_{P \in U(a,b)} \langle C, P \rangle_F, \tag{2.23}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius dot-product. As a result, the optimal transport plan P is also a matrix of size $n \times n'$ and elements P_{ij} indicates how much probabilistic mass must be transported from source sample i to the target sample j. An optimal solution to this problem can be found using the network simplex algorithm (Ahuja et al., 1988).

c-cyclical monotonicity. In the discrete setting, one can see OT as solving the minimum-cost flow on a bi-partite graph whose two vertex sets represent the source and target samples respectively. Let's consider an optimal transport problem between a source distribution composed of $x_i \in \mathcal{X}$ and a target distribution composed of $y_i \in \mathcal{Y}$ with cost $c : \mathcal{X} \times \mathcal{Y} \to (-\infty, +\infty]$. Now consider a particular subset of N edges on the bi-partite graph $(x_1, y_1), \ldots, (x_N, y_N)$ such that they form a cycle. Note that an edge (x_i, y_i) corresponds to a transfer of mass. If the edge belongs to non-optimal transport plan then there may exist a pair x_i, y_j such that the mass of x_i can be transported to y_j instead. Following this change, the mass of x_j cannot be transported to y_j anymore and therefore has to be moved to another y. This delta of mass is pushed all the way along the edges such that y_i actually receives its now missing mass. If the points are ordered along the changes made, that is x_1 is now transporting to y_2 and y_1 now receives mass from x_N, then the new transport plan has a lower cost if:

$$c(x_1, y_2) + c(x_2, y_3) + \cdots + c(x_N, y_1) < c(x_1, y_1) + \cdots + c(x_N, y_N). \tag{2.24}$$

A set of points (x_i, y_i), $i \in [1, N]$ is said to be c-cyclically monotone if there exists no such possible improvement. Taking an optimal transport plan, it is straightforward to see that it is c-cyclically monotone. Otherwise, there would exist a way to lower its associated cost and it would as such not be an optimal solution to the problem. As such, finding suboptimal cycles and lowering their cost by moving the mass along the cycle is a key part of the network simplex algorithm which is one way to solve an exact optimal transport problem.

Kantorovich Duality. Since empirical Optimal Transport can be written as a linear program. It exists a dual linear program whose solution will also be an optimal solution to the primal problem. The dual formulation of Optimal Transport can be written as:

$$L_C(a, b) = \max_{\mathrm{f,g}} \langle \mathrm{f}, a \rangle + \langle \mathrm{g}, b \rangle, \quad s.t. \quad \forall i, j \quad \mathrm{f}_i + \mathrm{g}_j \leq C_{i,j}. \tag{2.25}$$

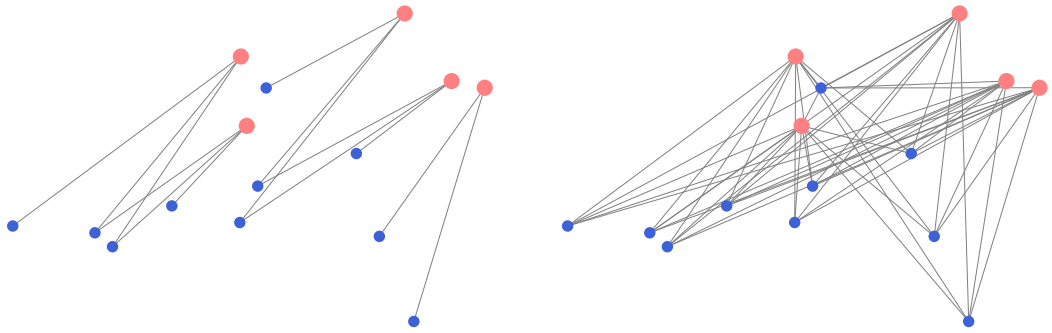
Unlike the primal solution which represents a joint probability distribution between atoms from the source and target distributions, the dual solution is in the form of two vectors f and g that represents the contribution of this point (*i.e.* its cost) to the final cost when weighted by its probabilistic mass.

The relationship of $(\mathrm{f}^\star, \mathrm{g}^\star)$, optimal solutions to the dual problem shown in Equation 2.25 with the optimal solution of the primal problem $P^\star$ is that for each pair (i, j) such that $P^\star_{i,j} > 0$, then the dual constraints are referred to as being *tight*, that is $\mathrm{f}^\star + \mathrm{g}^\star = C_{i,j}$. Reciprocally, if a pair (i, j) does not satisfy the tightness constraint by having $\mathrm{f}_i + \mathrm{g}_j < C_{i,j}$, then $P^\star = 0$ and therefore no mass is transported from source sample i to target sample j. As such, one can compute a pair of feasible duals from an optimal primal solution $P^\star$ by exploiting this property. This is done by setting an initial value for one of the dual and then by walking the underlying bi-partite graph enforcing tightness inside pairs where mass is transported in the primal solution.

Entropic Regularization. Since the computational complexity of OT remains a drawback for its adoption in Machine Learning (ML) tasks where datasets can be large, many regularizations of the OT problem have been proposed to make it more amenable for ML problems. One of the most popular is the entropic regularization (Cuturi, 2013) which modifies the original formulation to also include a term quantifying the entropy of the transport plan. The entropic regularized optimal transport is defined as:

$$L^\epsilon_C(a, b) = \min_{P \in U(a,b)} \langle P, C \rangle_F - \epsilon H(P), \tag{2.26}$$

where ϵ is a parameter to control the amount of importance given to the entropic regularization and $H(P)$ is the entropy of the coupling matrix defined as $H(P) = -\sum_{i,j} P_{i,j}(\log(P_{i,j}) - 1)$. In this case, one can see that when $\epsilon = 0$, the entropic regularized OT falls back to the regular OT problem. Interestingly, this regularization of the OT problem can be solved in $\mathcal{O}(n^2)$ complexity using the so-called Sinkhorn-Knopp algorithm (Cuturi, 2013) which can be implemented on GPUs, making it a method of choice when used in deep learning. Another advantage of entropic OT is that, unlike with regular OT, the resulting transport cost is differentiable. However, note that because of the added entropic regularization, the entropic transport plan is smoother than its exact transport counterpart. In fact, with $\epsilon \to +\infty$ the optimal entropic transport plan is the most uniform while still respecting the marginals $P_{i,j} = \frac{1}{a_i \times b_j}$. An example of entropic optimal transport plans with varying ϵ can be seen in Figure 2.11.



(a) $\epsilon = 0.01$. (b) $\epsilon = 10$.

Figure 2.11: Entropic Optimal Transport plan, as ϵ increases so does the smoothness of the optimal entropic transport plan.

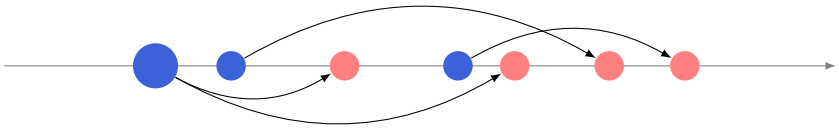
2.3.2 Sliced Wasserstein Distance

Solving the OT problem has a high complexity which motivates the proposal of new algorithms to scale OT to larger distributions. One such algorithm is the Sinkhorn-Knopp algorithm presented in Section 2.3.1. Another popular variant used to speed up Optimal Transport is called sliced optimal-transport. First, observe that while optimal transport has a generally high computational complexity of $\mathcal{O}(n^3)$ for distributions of n samples, solving OT between distributions supported on the real line $\mathbb{R}$ is easier. Indeed, on the real line, the solution comes naturally by walking along the line and transporting mass from source to target distributions. In the discrete setting, one needs to compute the empirical quantile function $F^{-1} : [0, 1] \to \mathbb{R}$ which is the inverse of the cumulative distribution function $F : \mathbb{R} \to [0, 1]$ for both the source and target distribution. In this case, the Wasserstein distance between μ and ν two distributions supported on the real line is:

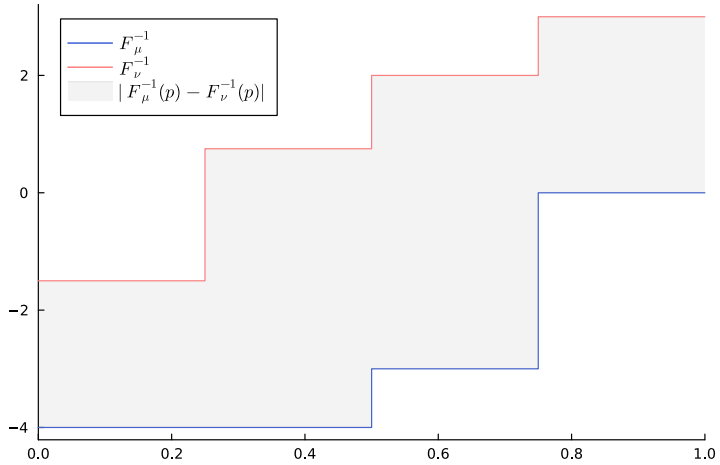
$$W_p(\mu, \nu) = \int_0^1 |F_\mu^{-1}(x) - F_\nu^{-1}(x)|^p \mathrm{d}x. \tag{2.27}$$

An example of an optimal transport problem between empirical distributions on the real line can be seen in Figure 2.12.

The Sliced Wasserstein (SW) distance, which was first introduced by Rabin et al. (2012), takes advantage of this lower complexity on the real line by projecting the dimensions in higher dimensions to lines and taking the mean cost of these 1D optimal transport costs along all these projections. Therefore, the Sliced Wasserstein (SW) distance between



(a) Samples from μ and ν on the real line.



(b) The corresponding empirical quantile functions for μ and ν.

Figure 2.12: A discrete optimal transport problem on the real line $\mathbb{R}$ and its optimal transport plan. The optimal transport cost corresponds to the gray area in subfigure 2.12b.

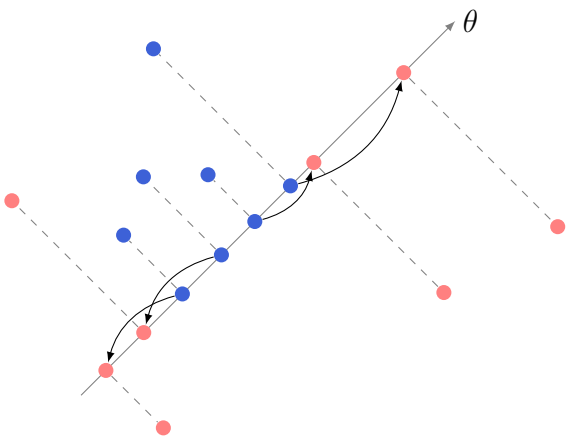


Figure 2.13: Projection on the straight line described by θ of the source and target distributions. The Optimal Transport is then solved along this line using the method based on the empirical quantile functions.

two distributions μ and ν supported on $\mathbb{R}^d$ is defined as:

$$\mathrm{SW}_p(\mu,\nu) = \int_{\mathbb{S}^d} W_p(P^\theta_\#\mu, P^\theta_\#\nu)\mathrm{d}\theta, \tag{2.28}$$

where $P^\theta_\#\mu$ represents the projection of μ on the straight line along θ. In practice, the integral is replaced with a Monte-Carlo estimation along straight lines whose directions are sampled uniformly around the unit hypersphere. An example projection of empirical distributions from the plane to a straight line can be seen in Figure 2.13. So given $\theta_i, i \in \{1,\dots,L\}$ the realization of L samples from $\Theta \sim \mathrm{Unif}(\mathbb{S}^d)$ the uniform distribution on the unit hypersphere, an estimation of the Sliced Wasserstein distance can be computed as such:

$$\mathrm{SW}_p(\mu,\nu) = \frac{1}{L}\sum_{i=1}^{L} W_p(P^{\theta_i}_\#\mu, P^{\theta_i}_\#\nu). \tag{2.29}$$

The Sliced Wasserstein distance has recently been extended to a number of Riemannian manifolds such as the hypersphere, hyperbolic spaces and the space of symmetric positive definite matrices in (Bonet et al., 2023c, 2023b, 2024) where the notion of straight lines is different than in Euclidean spaces.

2.3.3 Unbalanced Optimal Transport

Subsection 2.3.1 describes the general Optimal Transport problem, but several variants of the OT problem have been proposed over the years. One of these variants is the Unbalanced Optimal Transport (UOT) problem. General OT is referred to as being complete since all mass from the source distribution is transported to the target distribution (*i.e.* Both probabilistic masses sum up to 1). Indeed, one of the constraints of the Optimal Transport problem is respect of the marginals distributions in the set of all candidate solutions $\Gamma(\mu, \nu)$. UOT instead considers that mass which is too costly to transport should not be transported by allowing the marginal distributions of resulting transport plan to be modified with respect to the source and target distributions. The amount of modification of the marginal is quantified using a divergence between the source (*resp.* target) distributions and the resulting marginal distributions. The UOT problem can be formulated as such:

$$\inf_{\gamma} \int_{\Omega_\mu \times \Omega_\nu} c(x, y) d\gamma(x, y) dx dy + \lambda(\mathrm{KL}(\gamma_{\#}\mu \mid \mu) + \mathrm{KL}(\gamma_{\#}\nu \mid \nu)), \tag{2.30}$$

where KL refers to the Kullback-Leibler divergence and λ is a parameter to regularize the amount of distortion with respect to the source and target distributions. With this formulation, the regular Optimal Transport problem is recovered with $\lambda = +\infty$. An example of OT problem where it is too costly to transport mass can be see in Figure 2.14. The UOT can be solved with an entropic regularization (see Equation 2.26) using the algorithm proposed by Chizat et al. (2018).

In the context of Optimal Transport for machine learning, UOT has proven to be a useful tool. Since the empirical distributions observed in the world can have outliers where the OT plan would transport samples with a high cost. For example, this is the case when solving mini-batch OT since the entire support of distributions may be available at each batch (Fatras et al., 2021).

2.3.4 Optimal Transport between Incomparable Spaces

The OT problem and its unbalanced variant presented in Subsections 2.3.1 and 2.3.3 all require the existence of a cost function $c : \Omega_\mu \times \Omega_\nu \to [0, +\infty]$ which computes a distance between pairs of samples from each distribution μ and ν. The existence of this function prevents applying general OT between distributions supported on incomparable spaces. The Gromov-Wasserstein (GW) distance (Mémoli, 2011) is proposed as alternative dis-

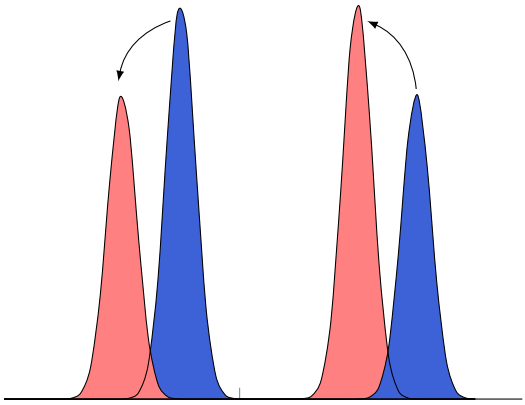


Figure 2.14: Unbalanced Optimal Transport between source and target distributions. The transport here is unbalanced since mass is created at each of the two modalities of the target instead of transporting from the source which is further away.

tance to the Wasserstein specifically for the case where source and target distributions are embedded in incomparable spaces. The core insight of the Gromov-Wasserstein distance is that instead of comparing the distance between elements in each distribution support, the GW distance compares the inner distance matrices of each distribution.

The GW transport plan is computed by comparing the inner metrics from each distribution. Therefore, the transport plan is based on the inner structure from each distribution instead of on the pairwise distances between samples from each distribution. Given two empirical distributions $\mu = \sum_{i=1}^{n} \delta_{x_i} a_i \in \mathcal{P}(\mathcal{X})$ and $\nu = \sum_{j=1}^{n'} \delta_{y_j} b_j \in \mathcal{P}(Y)$ over two metric spaces $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$ respectively with a and b, two simplex histograms of n and n' bins respectively such that $\sum_{i=1}^{n} a_i = 1$ and $\sum_{j=1}^{n'} b_j = 1$. δ_x is the Dirac measure in x. M_μ and M_ν are the matrices of pairwise distance between atoms of μ and ν respectively computed with respective distance functions $c_\mu : \mathcal{X} \times \mathcal{X} \to [0, +\infty]$ and $c_\nu : \mathcal{Y} \times \mathcal{Y} \to [0, +\infty]$. The GW distance solves the following problem:

$$GW^2(M_\mu, M_\nu, a, b) = \min_{\gamma \in U(a,b)} \sum_{i,j,k,l} |(M_\mu)_{i,k} - (M_\nu)_{j,l}|^2 \gamma_{i,j} \gamma_{k,l}. \tag{2.31}$$

Where the resulting optimal γ is a joint distribution between the source and target distribution. Note that in the case of the GW distance, there is indeed no need to compute a distance between samples from the source and target distribution. Those can therefore be embedded in incomparable spaces. An illustration of the GW distance can be seen in Figure 2.15 where $\mathcal{X} \subset \mathbb{R}^2$ and $\mathcal{Y} \subset \mathbb{S}^1$. As such, the respective distance metrics for μ

and ν are $c_\mu(x, x') = \|x - x'\|_2$ and $c_\nu(y, y') = \arccos(\langle y, y' \rangle)$.

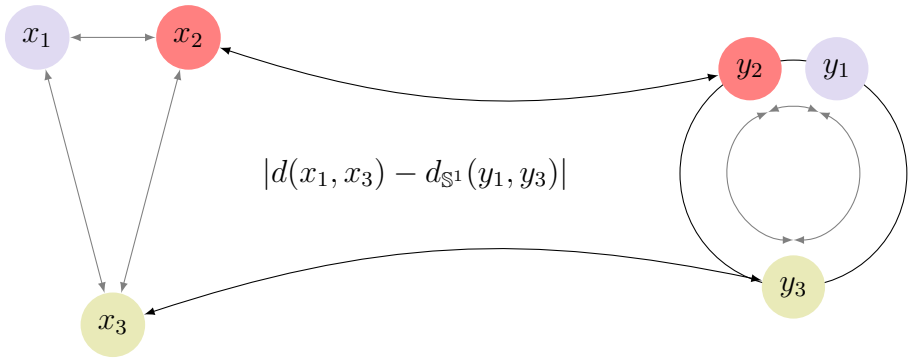


Figure 2.15: Gromov-Wasserstein distance between a source distribution on the plane equipped with the Euclidean distance and a target distribution on the circle equipped with the distance on the circle $d_{\mathbb{S}^1}(\cdot, \cdot)$. In this situation, one of the optimal solution is to transport from and to nodes of the same colors (x_1 to y_1, x_2 to y_2 and x_3 to y_3).

Following the successful applications of the GW distance to several machine learning problems, another variant of Optimal Transport on incomparable spaces was proposed. Instead of leveraging the inner metrics of the source and target distributions, Redko et al. (2020) propose to compute a coupling between features in order to project from the source embedding space to the target embedding space along with still optimizing the optimal transport plan between samples after said-projection. They call this joint-optimization problem Co-Optimal Transport (COOT).

This linear transformation can be seen as a transport plan between source and target variables. Given two empirical distributions X and X' supported on $\mathcal{X}$ (*resp.* $\mathcal{X}'$). We call π^s the transport plan between source and target samples and π^v the transport plan between variables of the source and target embedding spaces. The COOT problem is defined as follows:

$$\min_{\substack{\pi^s \in U(\mu,\nu) \\ \pi^v \in U(a,b)}} \sum_{i,j,k,l} L(X_{i,k}, X'_{j,l}) \pi^s_{i,j} \pi^v_{k,l} \tag{2.32}$$

where L is a divergence between 1D variables such that $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$. With the optimal solutions, one can use the variable-wise transport plan to project samples from $\mathcal{X}$ to $\mathcal{X}'$ using the π^v linear transformation. A simple example of the Co-Optimal Transport problem between two point clouds can be seen in Figure 2.16. In this example, the source distribution X is projected onto the real line using the feature-wise transport plan π^v. The COOT problem has also been studied in the unbalanced setting in Tran et al. (2023).

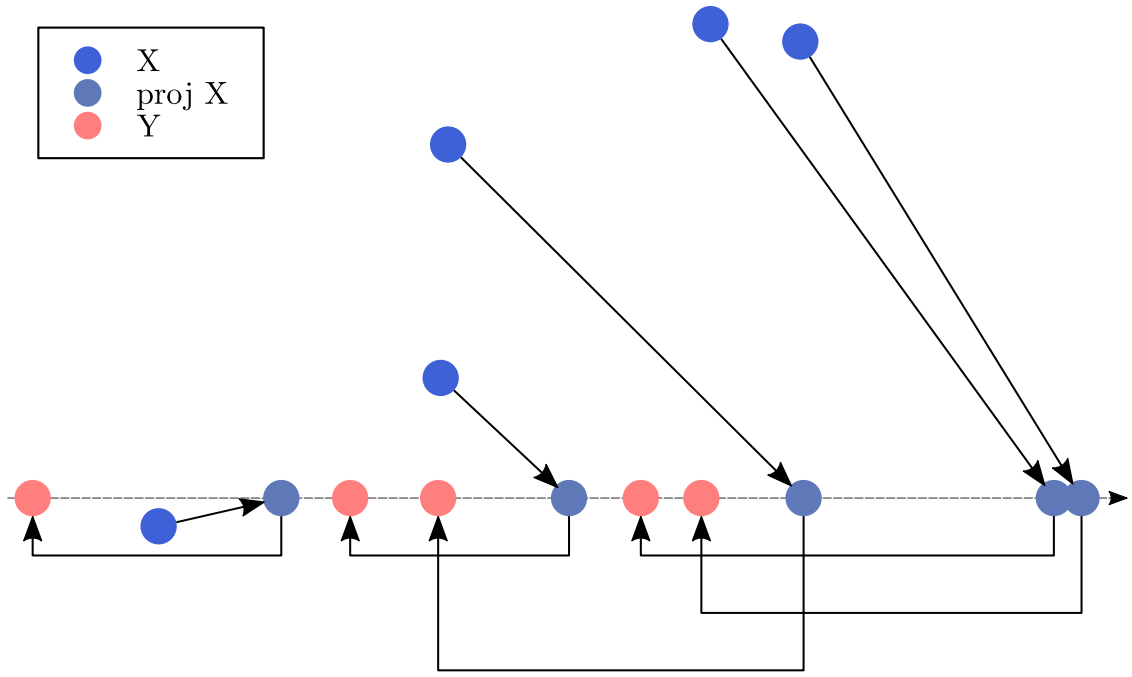


Figure 2.16: An example of a Co-Optimal Transport problem between X supported on $\mathbb{R}^2$ and Y supported on the real line $\mathbb{R}$ with uniform marginals both feature and sample-wise. Arrows from X to proj$X$ denote the variable transport plan π^v and arrows from proj$X$ to Y denote the sample-to-sample transport plan π^s on the line.

Chapter 3

Optimal Transport for Self-Supervised Learning

Contents

3.1 Transporting between Samples and Features 47

3.2 Hyperspherical Uniformity using Spherical Sliced Wasserstein 56

3.3 Robust Self-Supervised Object Detection 62

3.4 Conclusion and Discussion 68

In this chapter, we investigate how Optimal Transport can be leveraged in self-supervised representation learning. We will see that empirical distributions can be found in multiple places of the SSL framework. OT for SSL has successfully been used in the SwAV (Caron et al., 2020) and Dino-v2 (Oquab et al., 2023) methods to compute matchings between samples from a batch and a set of prototypes. We explore other ways to to model data as empirical distributions. One example is considering image representations in latent space as an empirical distribution drawn from the ground-truth distribution where OT can be useful to guide the training of the image encoder. In this example, its output distribution is guided by the OT cost. Another option is to model a single image as a distribution of patch representations supported in the embedding space. In this context, patches can be considered samples from the underlying image distribution. Both the OT cost and its associated transport plan can therefore become useful artifacts to compute meaningful pre-training objectives. Structural or computational constraints motivate us to reach for several variants of the OT problem such as entropic OT, sliced OT or unbalanced OT depending on the studied problem.

More specifically, in Section 3.1, we first consider the joint-embedding Self-Supervised Learning setting. In this framework, an image encoder is trained to produce discriminative image-level representations by using random augmentations to produce multiple views of the same image. We propose to augment this self-supervised pre-training by modeling

the generated representations as an empirical distribution embedded in latent space. We then investigate guiding the encoder's pre-training process using Optimal Transport on this distribution.

Section 3.2 starts from the observation of Wang and Isola (2020) that the classical contrastive learning loss can be decomposed in two independent components. Indeed, it can be written as the sum of an alignment loss and a uniformity loss on the hypersphere. In existing methods, the uniformity loss is implemented in a contrastive manner by maximizing the pairwise spherical distance between each representation. We introduce a new discrepancy between distributions on the hypersphere based on a sliced Optimal Transport formulation on hyperspheres. It notably has a closed-form solution when computed with respect to the uniform distribution on the sphere.

Then, in Section 3.3, we focus on the problem of Self-Supervised Learning for dense tasks such as object detection. Indeed, most methods proposed in the literature focus on generating image-level representations as opposed to patch or pixel level representations which are needed for dense tasks. Building upon the Dense Contrastive Learning (DenseCL) method proposed by Wang et al. (2021), we investigate how Optimal Transport can potentially improve the patch matching strategy in the loss for pre-training dense image encoders.

Finally in Section 3.4, we discuss our results and their potential shortcomings compared to existing methods in the literature.

3.1 Transporting between Samples and Features

3.1.1 Introduction

As seen in Subsection 2.1, Contrastive Self-Supervised Learning methods build a set of positive and negative samples for each image in a batch. Since such methods are self-supervised, the set of positives is often built from randomly augmented samples which generates positive views. Meanwhile, non-contrastive methods such as Barlow-Twins (Zbontar et al., 2021) try to maximize the amount of information shared among representations of positive samples by applying the information bottleneck to this random augmentation pipeline. In other words, it forces the cross co-variance matrix of the features to become diagonal in order to remove the redundant information present in the models representation. The key idea is that if an encoder model generates representation matrices

whose rank is equal to the dimension of the ambient space, then the entire embedding space is leveraged to maximize discrimination between samples. The relationship between contrastive and non-contrastive approaches has been studied in Garrido et al. (2023).

Starting from a batch of augmented images, we evaluate how to enforce the model to embed a view close to its corresponding positive. Instead of considering each view, its positive and their own negative views as samples, we model the problem as a transport problem where an entire batch can be seen as an empirical distribution drawn from the underlying ground-truth distribution specific to the dataset. This model enables us to leverage Optimal Transport and its variant in order to guide the pre-trained model to generate more aligned representations.

A similar approach is employed by Shi et al. (2023) where authors leverage the inverse Optimal Transport framework to extend and analyze the contrastive loss. The inverse OT formulation aims to solve the problem of finding the ground cost which will yield a transport plan as similar as possible as a target transport plan. The distance between the current transport plan and the target transport plan can be measured with a distance metric between distributions such as the Kullback-Leibler divergence. The authors also regularize the transport plan to encourage a uniform distribution of representations as this has been shown to be one of the required ingredient for successful Self-Supervised Learning of image representations by Wang and Isola (2020). While inverse OT is commonly used to optimize the ground cost matrix directly as part of a metric learning problem, Shi et al. (2023) instead optimize an image encoder model to produce representations that would generate the corresponding ground cost matrix.

More recently, Piran et al. (2024) proposed another self-supervised objective based on multi-marginal optimal transport to optimize the matching of more than two views to clusters corresponding to a single sample. Instead of inverse OT, they leverage the so-called Monge gap (Uscidda & Cuturi, 2023) which quantifies how much an estimated transport plan differs from an expected transport plan and can be used to train deep neural networks to embed samples that can then be transported accordingly.

Finally, in the context of language image pre-training, where the contrastive loss is also used (Radford et al., 2021), Shi et al. (2024) have made the connection between the contrastive loss and OT by integrating different variants such as Entropic OT (Cuturi, 2013), or Fused GW (Vayer et al., 2020). This transport formulation seems to improve over the contrastive loss in different mixed visual/language pre-training benchmarks. Given a target transport plan which is diagonal, as each image should be matched with its

corresponding text line, they jointly optimize both image and text encoders to produce representations that lead to a diagonal transport plan. The loss minimizes the Kullback-Leibler between the resulting transport plan and the diagonal ground-truth matrix.

In our case, we focus on the classical self-supervised setting. Where for each sample, two augmented views are created using random augmentations. Our starting point is indeed the simple contrastive pre-training framework from Chen et al. (2020a). Next, we consider the other side of the coin, which are non-contrastive methods such as Barlow-Twins (Zbontar et al., 2021) where the loss acts on correlations between features instead of samples. Finally, armed with methods from the OT literature, we propose a self-supervised method which combines these two viewpoints.

3.1.2 Methodology

A Contrastive Perspective. To introduce this section, we would like to consider the construction of the positive and negative sets for contrastive learning. It is known that choosing better negative samples can indeed improve the final performance of a self-supervised model (Robinson et al., 2020). In fact, in the worst case possible, the negative set is composed solely of samples from the same class as the anchor sample which prevents the model from learning discriminative representations. However in practice, the negative set contains samples from other classes and it therefore remains less costly to embed samples from the same class closer to the anchor sample even though they are negatives to each other. This phenomenon is one reason for the large batch size required for contrastive learning or the need for a queue of negative samples (He et al., 2020) which will introduce more class variety in the negatives set, making it more representative of the whole dataset.

The contrastive loss as described by Chen et al. (2020a) corresponds to softmax normalizing a pairwise similarity matrix for each row and then maximizing the resulting probability of the positive using the cross-entropy. We recall the NT-Xent loss in the following equations:

$$l(z, z^+, z^-) = -\log \frac{\exp(\langle z, z^+ \rangle / \tau)}{\sum_{z' \in z^- \cup \{z^+\}} \exp(\langle z, z' \rangle / \tau)}, \tag{3.1}$$

$$\mathcal{L}_{\text{NT-Xent}} = \frac{1}{2N} \sum_{z, z^+, z^-} l(z, z^+, z^-), \tag{3.2}$$

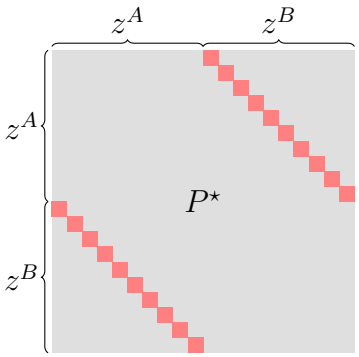


Figure 3.1: A permutation matrix $P^\star$ ($N = 10$).

where z is an augmented view with z^+ its positive and z^- its set of negatives.

Our idea is to leverage Optimal Transport in order to compute matching probabilities instead of using the row-wise softmax normalization. Instead of building a similarity matrix which is then normalized, we build a distance matrix which is used to compute an Optimal Transport transport plan. Our proposed approach can find parallels in Sinkformer (Sander et al., 2022) which replaces the softmax matrix in the self-attention layer of transformers with an OT plan (*i.e.* a bi-stochastic matrix). The difference in this case is that the desired OT plan (*i.e.* the ground truth matching) is known and given by the random augmentation process.

Thus, our goal is to train the model to output representations which will result in an empirical optimal transport plan as close as possible to the known matching. This problem statement is known as inverse Optimal Transport because of being interested in the transport cost, the matching is known and we want to find a distance matrix that will lead to the known transport plan. In case of a transport from a batch of N augmented views z^A to other augmented views z^B from the same sample, the known optimal matching is therefore the diagonal permutation matrix. Each view z_i^A should be transported to its corresponding z_i^B. If we take the formulation of computing a transport plan between all images and not only from z^A to z^B by building a $2N \times 2N$ cost matrix as done for Equation 2.4, then the optimal matching is a permutation matrix where the outer diagonals are non-zero, that is, $P^\star_{i,j} > 0$ if $i = j - N$ or $i = j + N$. The structure of the permutation matrix $P^\star$ can be seen in Figure 3.1. In order to prevent the distance matrix from having a diagonal filled with the value $d(x, x) = 0$, its diagonal is artificially filled with the value $C_{i,i} = +\infty, \forall i \in [1, \dots, 2N]$.

Given a set of N unlabelled images x, we generate two augmented views using a set of random augmentations $x_i^A = T_A(x_i)$ and $x_i^B = T_B(x_i)$. These augmented views are then embedded in latent space $\mathbb{S}^{d-1}$ using an encoder model $f_\theta : \mathcal{I} \to \mathbb{R}^d$ where $\mathcal{I}$ is the space of input images, followed by a normalization to make sure that the norm of representations is equal to 1. Thus, we define $z_i^A = \frac{f_\theta(x_i^A)}{\|f_\theta(x_i^A)\|}$ and $z_i^B = \frac{f_\theta(x_i^B)}{\|f_\theta(x_i^B)\|}$. A ground cost between representations can be computed using the cosine distance:

$$C_{ij} = 1 - \langle z_i^A, z_j^B \rangle, \quad \forall i, j \in [1, N] \times [1, N]. \tag{3.3}$$

Since we compute a matching plan between the two distributions of augmented views, we define the respective marginals a and b as uniform distributions. As such, each view is given the same amount of probabilistic mass as others and the resulting transport plan

for exact optimal transport can be a permutation matrix akin to a solution of the linear assignment problem (Kuhn, 1955).

The Optimal Transport plan between z^A and z^B is defined as:

$$\pi^\star = \underset{\pi \in U(a,b)}{\operatorname{argmin}} \langle \pi, C \rangle_F, \tag{3.4}$$

where $U(a, b)$ refers to the set of couplings with marginals equal to a and b, respectively.

Now, in order to enforce the transport plan to correspond to the known permutation matrix between samples with the full outer diagonals $P^\star$, we minimize the cross-entropy between the joint distribution described by $\pi^\star$ and the matching:

$$\mathcal{L}_{\text{Samples}}(\theta) = -\sum_{i=1}^{N} \log \pi^\star_{i,i+N} + \log \pi^\star_{i+N,i}. \tag{3.5}$$

Since the loss depends solely on the transport plan, it needs to be differentiable in order to be able to compute a gradient for the model's weights. As such, we solve the Optimal Transport from Equation 3.4 in its entropically regularized variant (see Subsection 2.3.1). On the other hand, exact OT would yield a sparse permutation matrix which would not be differentiable. In fact, with exact OT, $\mathcal{L}_{\text{Samples}}$ would be equal to $+\infty$ in the case where even one assignment does not satisfy the ground-truth matching. In the entropic variant, instead of a sparse transport plan, the resulting joint distribution can be smoother depending on the value of the ϵ parameter which regulates the importance given to the entropy of said transport plan:

$$\pi^\star = \underset{\pi \in U(a,b)}{\operatorname{argmin}} \langle \pi, C \rangle_F - \epsilon H(\pi), \tag{3.6}$$

and more importantly, the transport plan becomes differentiable and the computed cross-entropy loss is finite since the support is dense. Following this formulation of our OT based contrastive objective, we investigate the other paradigm in joint-embedding methods which is non-contrastive methods in which we believe OT methods can provide improvements.

Transport between Features. The other side of the coin in joint-embedding methods are non-contrastive objectives such as Barlow-Twins (Zbontar et al., 2021) (see Subsection 2.1). Instead of using a contrastive loss, the Barlow-Twins objective tries to make the model produce representations which have de-correlated features. This objective has two effects, since the cross-correlation is computed between two augmented batches of

views z^A and z^B (see Equation 2.6) then the objective implicitly aligns representations for augmented pairs (z_i^A, z_i^B) of the sample x_i. The second effect is that of leveraging the entire embedding space by limiting the correlation between features by minimizing the terms of the cross-correlation matrix which are not in the diagonal because they measure the cross-correlation of two different features (see Equation 2.5). Instead of this cross-correlation based objective, we propose to use Optimal Transport to compute matching between features and as with the contrastive case, we would want each feature in z^A to be transported to the corresponding feature in z^B. In this setting, we define a distance metric between features (*i.e.* variables) instead of samples inspired by the cross-correlation from Equation 2.6:

$$C_{ij} = 1 - \sum_{b=1}^{N} z_{bi}^A z_{bj}^B, \quad \forall i, j \in [1, d] \times [1, d], \tag{3.7}$$

where d corresponds to the number of variables of the latent representations such that $z_i^A \in \mathbb{S}^{d-1}$. In this case, instead of being of size $2N \times 2N$, the distance matrix is actually of size $d \times d$. As such, the optimal matching is actually a diagonal permutation matrix. We then can compute an optimal transport plan between features using this ground-cost matrix similarly to Equation 3.4. With the resulting variable-to-variable transport plan, the loss is to again minimize the anti-diagonal terms, thus maximizing the assignment of features to their known counterpart in the target augmented set z^B. With this approach, the hypothesis that the cross-correlation used in Barlow-Twins is a too aggressive regularization for a self-supervised objective and that instead that having feature correlations in such a way that their are c-cyclically monotone may be enough. We refer to this objective as $\mathcal{L}_{\text{Feature}}$ which can be noted as such:

$$\mathcal{L}_{\text{Feature}} = -\sum_{i=1}^{d} \log \pi_{i,i}^{\star}, \tag{3.8}$$

where in this case $\pi^\star$ is the transport plan between features. It is computed with respect to the ground cost matrix C defined in Equation 3.7.

Unifying Contrastive and Non-Contrastive Formulations. Our next step in the research process is trying to unify both formulations since they only are transposed versions of each other. That is, can we train an encoder model to produce representations whose sample-wise transport is diagonal and its feature-wise transport plan is diagonal as well?

One tool in the Optimal Transport toolbox that was presented in Section 2.3 which can solve such problems is Co-Optimal Transport (COOT) (Redko et al., 2020) (see Subsection 2.3.4). Indeed, recall that COOT computes both a sample-to-sample and a feature-to-feature transport plan in a joint manner such that the transport cost is minimized between the linear projection of the source distribution through the feature plan and the target distribution. The objective then becomes to have the two jointly optimized transport plans diagonal by minimizing anti-diagonal terms in each plan. The sample-to-sample plan and the variable-to-variable transport plan are called π^s and π^v respectively. They can be obtained by finding an optimal solution to the following COOT problem:

$$\min_{\substack{\pi^s \in U(\mu,\nu) \\ \pi^v \in U(a,b)}} \sum_{i,j,k,l} L(z^A_{i,k}, z^B_{j,l}) \pi^s_{i,j} \pi^v_{k,l}. \tag{3.9}$$

Following this definition, the COOT-based objective is defined as:

$$\mathcal{L}_{\text{COOT}}(\theta) = -\lambda_{\text{Samples}} \sum_{i=1}^{N} \left(\log \pi^s_{i,i+N} + \log \pi^s_{i+N,i} \right) - \lambda_{\text{Features}} \sum_{i=1}^{d} \log \pi^v_{i,i}, \tag{3.10}$$

similarly to the sample and feature-wise counterparts but with an added λ_{Samples} and $\lambda_{\text{Features}}$ to weight between the two terms. Note that since COOT solves a series of OT problems as part of its optimization process (Redko et al., 2020), one can use entropic optimal transport in the inner loop of the algorithm. In this case, the value of the entropic regularization parameter ϵ needs to be adjusted.

3.1.3 Experiments

In order to evaluate the performance of our optimal transport based self-supervised losses, we first pre-train an encoder. Then, given its frozen weights, we apply the linear probing protocol by training a linear classifier on top. Once trained, we evaluate the performance of the linear classifier on the test set of the dataset. Results can be seen in Table 3.1. We train and evaluate the performance of methods on the CIFAR10 (Krizhevsky, 2009) and STL10 (Coates et al., 2011) datasets.

The CIFAR10 dataset is composed of 60 000 images distributed among 10 classes each of size 32×32 pixels. Its small scale makes it an interesting choice to test the initial performance of a self-supervised method.

The STL10 dataset is composed of 100 000 unlabelled images extracted from the larger ImageNet dataset, it also contains a training split with 500 training images distributed among 10 classes. All images are coloured and of size 96×96 pixels. The unlabelled split makes the STL10 a dataset of choice to test the performance of self-supervised methods since the self-supervised pre-training can be done on the unlabelled and training splits whereas the downstream finetuning or linear classification is trained only on the labelled training split. Authors have mentioned that the dataset is inspired by the CIFAR10 dataset.

We perform self-supervised pre-training on different SSL methods, namely SimCLR (Chen et al., 2020a), Barlow-Twins (Zbontar et al., 2021) and Maximum Manifold Capacity Representation (Yerxa et al., 2023). For each methods, we use a batch size of 1024 samples. For Barlow-Twins, the λ is set to 0.0051 as proposed in the original paper. For $\mathcal{L}_{\text{Feature}}$ and $\mathcal{L}_{\text{Sample}}$, we set the ϵ of the entropic OT solver to 0.01 with 5 iterations. For $\mathcal{L}_{\text{COOT}}$, we pick $\lambda_{\text{Samples}} = \lambda_{\text{Features}} = 1$ for simplicity. The resulting pre-trained models are then compared on evaluation benchmarks. In order to evaluate, we compare the models on a finetuning benchmark where the pre-trained weights serve as initialization for a finetuning over the entire labelled dataset. We also perform an evaluation on the linear evaluation protocol where the self-supervised model weights are frozen and a simple linear classifier is trained on top of these representations (see Subsection 2.1 from more details). Results for the linear evaluation protocol can be seen in Table 3.2. In this results, we can see that non-contrastive methods such as Barlow-Twins (Zbontar et al., 2021) and MMCR (Yerxa et al., 2023) perform better than other methods included in the benchmark, including our proposed methods based on Optimal Transport without having as much hyper-parameters as the OT based methods. Among OT based methods, the feature wise transport plan yields the best performance. While the performance of the best OT method is on par with SimCLR (Chen et al., 2020a) it does not reach the same performance as more recent non-contrastive methods such as Barlow-Twins or MMCR.

3.1.4 Discussion

The results in these experiments are not able to show that these OT based objectives perform better than current state-of-the-art methods. In this case, it seems that the relaxed nature of the Optimal Transport based objectives does not provide a performance improvement. We note that Shi et al. (2023) test only against contrastive methods such as the triplet loss (Schroff et al., 2015) or SimCLR. Piran et al. (2024) focuses on the problem

Method	CIFAR10		STL10	
	Acc@1	Acc@5	Acc@1	Acc@5
SimCLR (Chen et al., 2020a)	74.17%	98.55%	59.24%	96.63%
Barlow-Twins (Zbontar et al., 2021)	87.65%	99.67%	73.78%	98.70%
MMCR (Yerxa et al., 2023)	**88.11%**	**99.79%**	**80.44%**	**99.14%**
$\mathcal{L}_{\text{Feature}}$	82.71%	99.45%	71.44%	98.49%
$\mathcal{L}_{\text{Sample}}$	77.45%	98.70%	71.63%	98.34%
$\mathcal{L}_{\text{COOT}}$	75.20%	98.56%	42.84%	93.29%

Table 3.1: Finetuning accuracy results. Acc@1 refers to the top-1 accuracy while Acc@5 corresponds to the top-5 accuracy.

Method	CIFAR10		STL10	
	Acc@1	Acc@5	Acc@1	Acc@5
SimCLR (Chen et al., 2020a)	73.90%	98.21%	54.53%	95.79%
Barlow-Twins (Zbontar et al., 2021)	82.67%	99.37%	70.86%	98.69%
MMCR (Yerxa et al., 2023)	84.06%	99.27%	73.18%	98.53%
$\mathcal{L}_{\text{Feature}}$	81.34%	99.23%	73.41%	98.68%
$\mathcal{L}_{\text{Sample}}$	81.29%	99.32%	68.78%	98.00%
$\mathcal{L}_{\text{COOT}}$	52.05%	94.31%	40.85%	92.31%

Table 3.2: Accuracy results on the linear evaluation protocol. Acc@1 refers to the top-1 accuracy while Acc@5 corresponds to the top-5 accuracy.

of more than two views and exhibits better performance in this case than SimCLR and BYOL (Grill et al., 2020). As such, we recon that solely optimizing the transport plan may not be enough and that instead it could be combined with the distance matrix. Optimizing with the distance matrix would have a more direct effect on the representations than using only the transport plan in the objective. We also can see that OT methods have successfully been used to formulate pseudo-labels in SwAV (Caron et al., 2020) and Dino-v2 (Oquab et al., 2023) where the online branch is trained to predict such pseudo-labels. So our approaches may yield better results when used with a similar an asymmetric setup with an online branch and an offline branch updated with an Exponentially Moving Average (EMA).

In the next section, we consider the contrastive learning from another viewpoint. Indeed, the contrastive loss can be decomposed in two components. For one of those, a variant of Optimal Transport turns out to be a natural fit.

3.2 Hyperspherical Uniformity using Spherical Sliced Wasserstein

3.2.1 Introduction

As seen in Section 2.1, many self-supervised methods rely on spherical projections to compute distances between representations. This normalization enables usage of the so-called cosine similarity as a measure of alignment between representations. The goal of contrastive learning is both to pull representations of positive samples closer together while pushing representations of so-called negatives further away on the hypersphere (*i.e.* minimize alignment). Having only the first term maximizing the alignment of positives would not be enough since it can be minimized in a trivial manner. Indeed, if the encoder model only outputs a constant representation independent of the input, then alignment is maximized for every sample pairs but the representations lose their informative nature and cannot be used to solve a downstream task. As such, negatives are used to prevent the model representations from collapsing to a single point.

The contrastive objective from (Chen et al., 2020a) shown in Equation 2.4 can be rewritten in two terms, one for the positives and another for the negatives:

$$\mathcal{L}_{\text{NT-Xent}} = -\frac{1}{2N}\sum_{z,z^+,z^-}\log\left(\frac{\exp(\langle z,z^+\rangle/\tau)}{\sum_{z'\in z^-\cup\{z^+\}}\exp(\langle z,z^-\rangle/\tau)}\right) \tag{3.11}$$

$$= -\frac{1}{2N}\sum_{z,z^+,z^-}\log(\exp(\langle z,z^+\rangle/\tau)) - \log\left(\sum_{z'\in z^-\cup\{z^+\}}\exp(\langle z,z^-\rangle/\tau)\right) \tag{3.12}$$

$$= -\frac{1}{2N}\sum_{z,z^+,z^-}\underbrace{\langle z,z^+\rangle/\tau}_{\text{Alignment}} \underbrace{- \log\left(\sum_{z'\in z^-\cup\{z^+\}}\exp(\langle z,z^-\rangle/\tau)\right)}_{\text{Uniformity}}. \tag{3.13}$$

These two terms lead Wang and Isola (2020) to investigate the impact of each. They suggest that two ingredients are required to perform representation learning successfully with positives, Alignment and Uniformity as can be shown in Equation 3.13 for the NT-Xent loss. Alignment means making representations of positive samples closer by maximizing the cosine similarity between z and z^+ in this case. The second one prevents the sample from collapsing to a single point. They call this term uniformity. Unlike SimCLR (Chen et al., 2020a), they propose an uniformity term based on maximizing

the Gaussian potentials between pairwise points. The two objectives proposed by these methods are based on pairwise distance maximization between representations of negatives. This lead us to investigate the use of a uniformity metric which is not solely based on pairwise distance maximization but leverages a divergence with the hyperspherical uniform distribution instead.

3.2.2 Spherical Sliced Wasserstein

The Spherical Sliced Wasserstein (SSW) distance (Bonet et al., 2023b) is a distance between distributions embedded on the hypersphere which leverages the sliced optimal transport distance on the circle (Delon et al., 2010). On the circle, the optimal transport plan is also easier to compute similarly to the real line. It requires computing the empirical quantile functions but additional precaution is required because of the circular nature of the space and therefore one could start the optimal transport assignment at any point of the circle. An example of circular Optimal Transport problem between two simple empirical distributions can be seen in Figure 3.2.

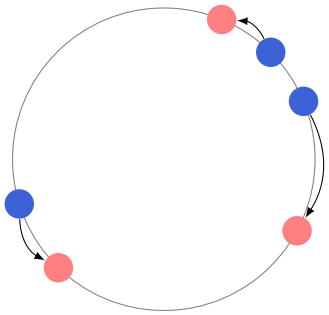


Figure 3.2: A simple empirical Optimal Transport problem on the circle $\mathbb{S}^1$.

Unlike the Sliced Wasserstein distance defined in Equation 2.28, SSW uses projections on the circle. Therefore, similarly to SW, SSW is defined for $p \geq 1$ between locally absolutely continuous measures μ and ν as an integral over all directions:

$$\mathrm{SSW}_p^p(\mu, \nu) = \int_{\mathbb{S}^d} \mathrm{W}_p^p(\mathring{P}_{\#}^{\theta}\mu, \mathring{P}_{\#}^{\theta}\nu)\mathrm{d}\theta, \tag{3.14}$$

where $P_{\#}^{\theta}$ is the geodesic projection on the great circle generated by θ and then projected onto $\mathbb{S}^1$ (*i.e.* the circle). This projection is defined as:

$$\mathring{P}^{\theta}(x) = \frac{\theta^\top x}{\|\theta^\top x\|}. \tag{3.15}$$

An example of the great circle projection applied to points located on the sphere can be seen in Figure 3.3.

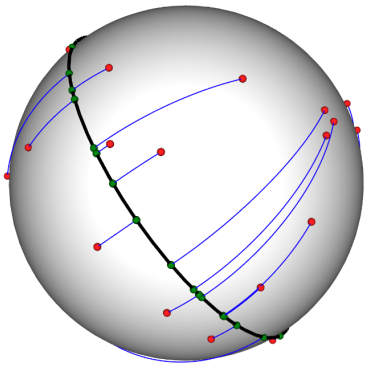


Figure 3.3: The great circle projection $\mathring{P}^{\theta}(x)$ on a set of points located on the sphere $\mathbb{S}^2$.

In Equation 3.14, the Wasserstein distance is defined with the geodesic distance on the circle which measures the length of the arc between two points. When dealing with empirical distributions, SSW can be estimated by computing the expectation over a set of $L \geq 1$ uniformly sampled directions $\theta_i \sim \text{Unif}(\mathbb{S}^d), i \in \{1, \dots, L\}$ in a similar manner as the Sliced Wasserstein distance presented in Equation 2.29:

$$\text{SSW}_p^p(\mu, \nu) = \frac{1}{L} \sum_{i=1}^{L} W_p(\mathring{P}_{\#}^{\theta^i} \mu, \mathring{P}_{\#}^{\theta_i} \nu) \tag{3.16}$$

Interestingly for us, SSW has a closed-form solution when the Spherical Sliced Wasserstein distance is taken between an empirical distribution and the uniform distribution on the hypersphere. Indeed, the push forward through $\mathring{P}^{\theta}$ of the spherical uniform distribution $\nu = \text{Unif}(\mathbb{S}^{d-1})$ remains a uniform distribution on $\mathbb{S}^1$.

This naturally leads us to leverage SSW for Self-Supervised Learning using the uniformity loss presented in Table 3.3 which has a lower computational complexity than the other uniformity losses used in (Chen et al., 2020a; Wang & Isola, 2020) since they are based on pairwise comparisons and therefore scale non-linearly with regard to the number of samples.

A simple choice for the alignment loss is to minimize the mean squared Euclidean distance between pairs of different augmented versions of the same image. A Self-Supervised Learning network is pre-trained using this alignment loss to which is added an uniformity term. Our overall self-supervised loss can be defined as:

Method	$\mathcal{L}_{\text{uniform}}(z^A) + \mathcal{L}_{\text{uniform}}(z^B)$	Complexity
SimCLR (Chen et al., 2020a)	$\frac{1}{2N}\sum_{i=1}^{N}\log\sum_{j\neq i}\exp(\frac{\langle \hat{z}_i, \hat{z}_j\rangle}{\tau}),\ \hat{z}=\text{cat}(z^A, z^B)$	$O(N^2d)$
Wang and Isola (2020)	$\sum_{z\in\{z^A,z^B\}}\log\frac{2}{n(n-1)}\sum_{i>j}\exp(-t\|z_i - z_j\|_2^2)$	$O(N^2d)$
SSW-SSL (Ours)	$\frac{1}{2}(\text{SSW}_2^2(z^A,\nu) + \text{SSW}_2^2(z^B,\nu))$	$O(LN(d+\log N))$

Table 3.3: Comparison of contrastive methods and their respective uniformity objective where $z^A, z^B \in \mathbb{R}^{n\times d}$ are representations from two augmented versions of the same set of images and $\nu = \text{Unif}(\mathbb{S}^{d-1})$ is the uniform distribution on the hypersphere. L is the number of projections for SSW.

$$\mathcal{L}_{\text{SSW-SSL}} = \underbrace{\frac{1}{N}\sum_{i=1}^{N}\|z_i^A - z_i^B\|_2^2}_{\text{Alignment loss}} + \frac{\lambda}{2}\Big(\underbrace{\text{SSW}_2^2(z^A,\nu) + \text{SSW}_2^2(z^B,\nu)}_{\text{Uniformity loss}}\Big), \tag{3.17}$$

where $z^A, z^B \in \mathbb{R}^{n\times d}$ are the representations from the network projected on the hypersphere of two augmented versions of the same images, $\nu = \text{Unif}(\text{S}^{d-1})$ is the uniform distribution on the hypersphere and $\lambda > 0$ is used to balance the two terms.

3.2.3 Experiments

Method	Encoder output	$\mathbb{S}^2$
Supervised	82.26	81.43
SimCLR (Chen et al., 2020a)	**66.55**	<u>59.09</u>
Wang and Isola (2020)	60.53	55.86
SW-SSL, $\lambda = 1, L = 10$	62.65	57.77
SW-SSL, $\lambda = 1, L = 3$	62.46	57.64
SSW-SSL, $\lambda = 20, L = 10$	<u>64.89</u>	58.91
SSW-SSL, $\lambda = 20, L = 3$	63.75	**59.75**

Table 3.4: Linear evaluation on the CIFAR10 (Krizhevsky & Hinton, 2009) dataset. The features are taken either on the encoder output or directly on the sphere $\mathbb{S}^2$. The supervised method results show the upper bound of the performance.

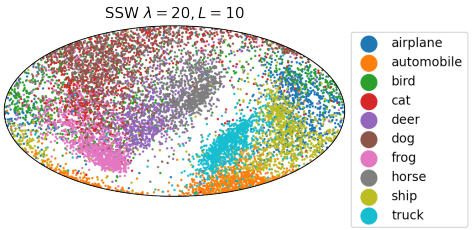
As an experiment, we evaluate the encoder performance on linear classification after self-supervised pre-training. We test using our proposed $\mathcal{L}_{\text{SSW-SSL}}$ and a variant of it which relies on the Sliced Wasserstein (SW) distance between the point clouds an the uniform distribution on the hypersphere which we refer to as $\mathcal{L}_{\text{SW-SSL}}$. We use a ResNet18 (He et

al., 2016) encoder which outputs 1024 features that are then projected onto the sphere S^2 using a last fully connected layer followed by a ℓ^2 normalization. We pretrain the model for 200 epochs using minibatch stochastic gradient descent (SGD) with a momentum of 0.9, a weight decay of 0.001 and an initial learning rate of 0.05. We use a batch size of 512 samples. The images are augmented using a standard set of random augmentations for SSL: random crops, horizontal flipping, color jittering and gray scale transformation as done in Wang and Isola (2020). For the trade-off parameter λ, we $\lambda = 20$ for SSW and $\lambda = 1$ for SW. To evaluate the performance of representations, we use the common linear evaluation protocol where a linear classifier is fitted on top of the pre-trained representations and the best validation accuracy is reported. The linear classifiers are trained for 100 epochs using the Adam (Kingma & Ba, 2014) optimizer with a learning rate of 0.001 with a decay of 0.2 at epoch 60 and 80. We compare our methods with two other contrastive objectives, Chen et al. (2020a) with the normalized temperature-scaled cross-entropy (NT-Xent) loss and Wang and Isola (2020) which proposes to decompose the objective in two distinct terms $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$. We recall the respective uniformity loss of each method in Table 3.3. As one can see in Table 3.4, our method achieves here comparable performances to two state-of-the-art approaches, yet slightly under-performing compared to Chen et al. (2020a). We suspect that a finer validation of the balancing parameter λ is needed. Especially since the representations on Figure 3.4 are not completely uniformly distributed around the sphere after pre-training compared to other contrastive methods.

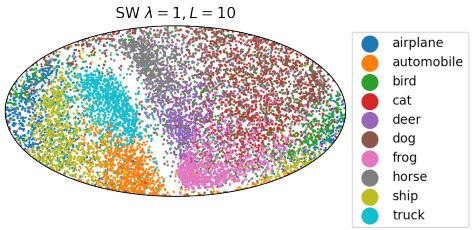
3.2.4 Discussion

While the accuracy scores are interesting when performing classification directly on the sphere, SSW does not show a performance superior to other uniformity methods based on contrastive interactions between samples. Nevertheless, these results show that other metrics can be explored to enforce uniformity on the hypersphere to create promising contrastive learning approaches without explicit distances between negative samples.

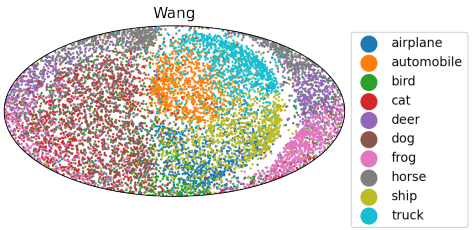
After having investigated regular image-level self-supervised methods, we turn to pre-training methods for dense tasks such as object-detection or semantic segmentation in the next section.



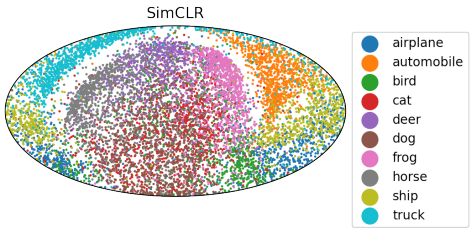
(a) SSW-SSL.



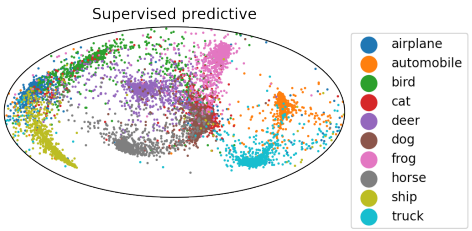
(b) SW-SSL.



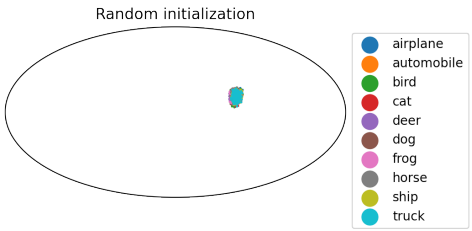
(c) Wang and Isola (2020).



(d) SimCLR (Chen et al., 2020a).



(e) Supervised prediction.



(f) Random initialization.

Figure 3.4: The CIFAR10 (Krizhevsky & Hinton, 2009) validation set on $\mathbb{S}^2$ after pre-training.

3.3 Robust Self-Supervised Object Detection

3.3.1 Introduction

As seen in Section 2.1, self-supervised methods have achieved great success at learning discriminative image-level representations without relying on human provided annotations but rather by generating its own pseudo-labels to use in a self-defined training objective. Models pre-trained in a self-supervised fashion can then be used in a variety of downstream tasks oftentimes with better performance than training from scratch or leveraging an off-the-shelf model pre-trained on a larger dataset such as the ImageNet dataset (Russakovsky et al., 2015).

However, when self-supervised models trained using joint-embedding methods are tested on dense downstream tasks (*i.e.* object detection, semantic segmentation,...), the performance increase is not as important as for image-level downstream tasks such as image classification. Indeed, almost all proposed self-supervised objectives focus on generating discriminative image-level representations. Moreover, the transformation invariance which is mandated by joint-embedding methods (see Section 2.1) forces models trained in a self-supervised fashion to generate spatially invariant representations in order to align representations from images with different viewpoint of the same sample. This spatial invariance goes against the requirement for dense tasks which require discriminative patch or pixel level representations. As such, Wang et al. (2021) propose a self-supervised objective which trains an image encoder to produce better spatial representations suited for dense tasks. Authors introduce a new self-supervised objective which operates between image patches unlike image-level methods which operate on a single representation per image. They name their proposed method Dense Contrastive Learning (DenseCL).

When training a convolutional neural network as a dense image encoder, generating dense representations essentially means extracting features before the last global pooling layer. In the context of Vision Transformer (ViT) (Dosovitskiy et al., 2020), patches are naturally represented by each of the visual tokens extracted from the image. Each element of these representations can be traced back to its original patch in the image. The goal of such dense methods is therefore to produce discriminative representations at the patch level to finally use them in tasks such as object detection or semantic segmentation where discriminative representations at the patch level are important for correct image localization.

Finding Positive Patches. Consider a dense image encoder $f_\theta : \mathbb{R}^{c\times h\times w} \to \mathbb{R}^{\mathbf{c}\times\mathbf{h}\times\mathbf{w}}$,

which generate dense representation for images. Similarly, to image-level Self-Supervised Learning, for each image, augmented views are generated which are then fed in the encoder $z_i^A = f_\theta(T^A(x_i)), z_i^B = f_\theta(T^B(x_i))$. For each patch j in augmented view z_i^A of sample i, a corresponding patch t_+ to be used as positive is searched in the positive view z_i^B:

$$t_+(j) = \underset{k}{\text{argmax}}\ \text{sim}(z_{i,j}^A, z_{i,k}^B), \tag{3.18}$$

where $\text{sim}(\cdot, \cdot)$ corresponds to the cosine similarity. This positive patch is then used during training as a form of pseudo-label as a positive in a dense contrastive loss whose definition is:

$$\mathcal{L}_{\text{Dense}}(z^A, z^b; \theta) = -\sum_{i=1}^{N} \sum_{j=1}^{\text{h}\times\text{w}} \log \left(\frac{\exp(\text{sim}(z_{i,j}^A, z_{i,t_+(j)}^B)/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}(z_{i,j}^A, z_k^B)/\tau)} \right), \tag{3.19}$$

where z_k^B is a shorthand notation for the image-level representation of sample k augmented with B.

This matching strategy between patches is simple yet effective and improves the performance of image-level methods on dense downstream tasks with several points. But this strategy can also result in sub-optimal matching between patches. Indeed, consider the case where the two augmented views have been generated with only a single overlapping area. Each augmented view therefore has exclusive patches which are not present in the other view. Following the matching rule from Equation 3.18, each patch will be matched with its most similar in the other augmented view even though this may not make semantic sense because the loss forces two unrelated patches to have more similar representations.

Another problem encountered by the authors is that of "*the chicken and egg*" problem. In simple terms, the signal provided by the matching part is initially random during the training and cannot really be relied upon to self-improve the model. However, once the model has trained for long enough, the matching signal can be more reliably used to further improve the model. Therefore, the dense objective is supported by using a global contrastive objective which we refer to as the Global loss:

$$\mathcal{L}_{\text{Global}}(z^A, z^B; \theta) = \frac{-1}{N} \sum_{i=1}^{N} \log \left(\frac{\exp(\text{sim}(z_i^A, z_i^B)/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(z_i^A, z_j^B)/\tau)} \right). \tag{3.20}$$

The model is therefore trained with a loss that combines both $\mathcal{L}_{\text{Dense}}$ and $\mathcal{L}_{\text{Global}}$ with an equal weight during the training.

3.3.2 Optimal Matching

In order to alleviate these problems, we choose to rely on Optimal Transport to improve the patch matching part of the algorithm. That is, instead of making each patch closer to its most similar patch in the other image, we formulate the matching problem as an Optimal Transport one. The first key observation is to consider a dense representation of an image as a distribution of patches. OT then integrates nicely in this framework by providing a natural way to compute a distance between two dense image representations. One can then naturally compute an OT distance between the two augmented views z^A and z^B. Starting by computing the distance matrix between patch representations in each views:

$$C_{i,j} = -\text{sim}(z_i^A, z_j^B). \tag{3.21}$$

And then find a solution to the OT problem between the two distributions with uniform marginals a, b respectively:

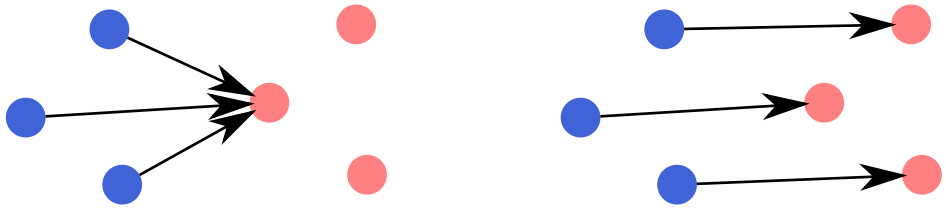
$$\pi^\star = \underset{\pi \in U(a,b)}{\text{argmax}} \ \langle C, \pi \rangle_F. \tag{3.22}$$

The resulting transport plan can be used as an alternative matching strategy to the one described in Equation 3.18:

$$ot_+(i,j) = \underset{k,l}{\text{argmax}} \ \pi^\star_{(i,j),(k,l)}. \tag{3.23}$$

On its own, this strategy does not offer better matching than the one described in Equation 3.18 which was proposed in (Wang et al., 2021). Indeed, with this new strategy the overall cost between patches is actually higher since Equation 3.18 essentially describes a relaxation of marginals in Equation 3.22. A comparison of the resulting match on a toy point-cloud can be seen on Figure 3.5.

Therefore, we decide to leverage Unbalanced Optimal Transport in a refined matching strategy. As explained in Subsection 2.3.3, UOT computes an optimal transport plan which does not strictly need to respect the marginals in the case where it is too costly to transport mass between samples. Consider the following case, an image with two objects is augmented is two views using random augmentations which include random cropping. However, the random cropping generate one view with object A and another view with object B with little to no overlap between the two images. With the most-similar patch



(a) Matching strategy based on most similar patch from Equation 3.18.

(b) Matching strategy based on the Optimal Transport plan from Equation 3.23

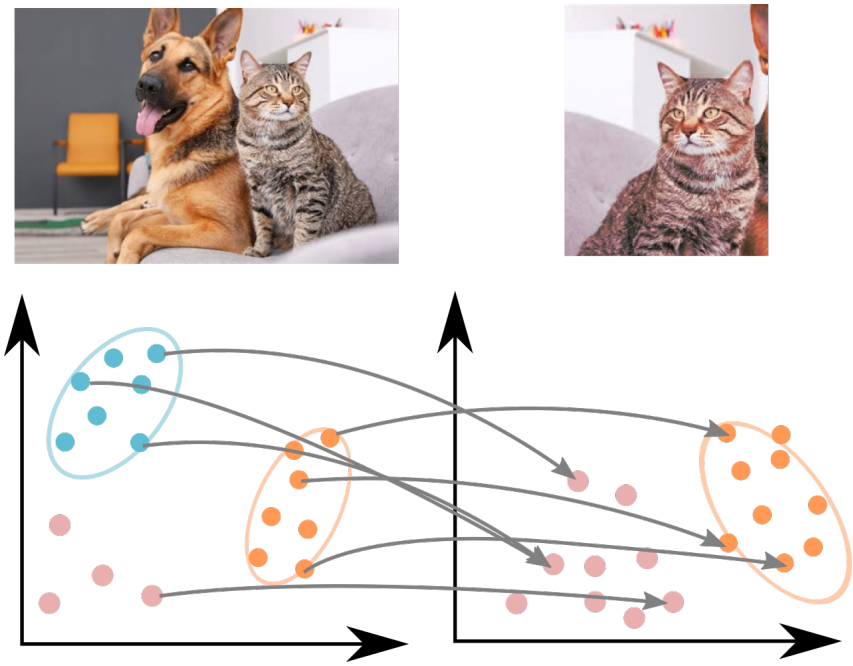
Figure 3.5: Comparison of patch matching strategy, patch representations from z^A are represented in blue while patch representations from z^B are represented in red, an edge between two patches signifies that the two patches are matched.

matching strategy, patches from one view could be matched to patches in the other view even though they have no semantic relationship, and only because they happen to be most similar. On the contrary, using Unbalanced Optimal Transport allows patches which are too dissimilar to patches in the other images to not be matched and to not have as much impact in the dense contrastive loss computation. A visual interpretation of such problem can be seen in Figure 3.6.

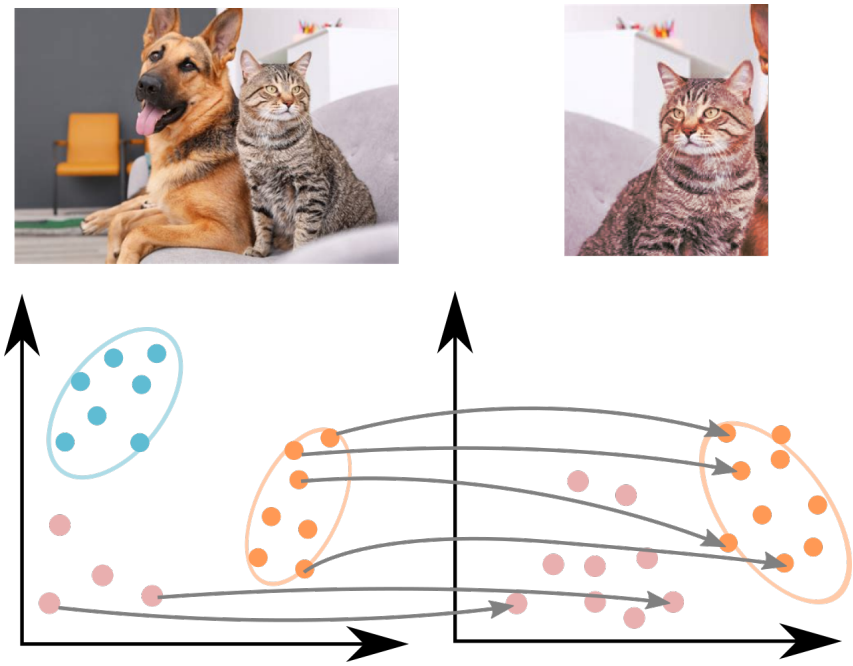
A Soft Contrastive Loss. One difference between the hard matching from maximum in Equation 3.18 and using OT for matching is that OT-based matching provides the model with a soft-matching. Indeed, multiple patches can be matched with an anchor patch. Said differently, the probabilistic mass of a given anchor patch can be transported to multiple other target patches in the other view. This soft-matching allows us to do a modification to the contrastive loss as we know it by using the amounts of probabilistic mass transported as a target vector for each anchor patch. We name this modified loss $\mathcal{L}_{\text{Soft-Dense}}$ as it target soft labels instead of hard ones, its formulation is based on the cross-entropy between the target vector which is a row of the matching optimal transport plan and the softmax of the distances with the corresponding plans:

$$\mathcal{L}_{\text{Soft-Dense}}(z^A, z^b; \theta) = -\sum_{i=1}^{N} \sum_{j=1}^{\text{h}\times\text{w}} \sum_{k=1}^{\text{h}\times\text{w}} (\text{h}\times\text{w}) \times \pi^{\star}_{j,k} \log \left(\frac{\exp(\text{sim}(z^A_{i,j}, z^B_{i,k})/\tau)}{\sum_{l=1}^{N} \exp(\text{sim}(z^A_{i,j}, z^B_{l})/\tau)} \right). \quad (3.24)$$

One can see that having a sparse transport plan which is a permutation matrix will recover $\mathcal{L}_{\text{Dense}}$ from Equation 3.19 since for each row i of $\pi^{\star}$ there will only be one j such that $\pi^{\star}_{i,j} = 1$. This difference is identical to that between categorical cross-entropy and cross-entropy where labels are one-hot encoded in classification.



(a) Matching with argmax.



(b) Matching with Unbalanced Optimal Transport.

Figure 3.6: Example of augmentation that could be problematic for the cropping augmentations. One of the views only contains the cat and not the dog. Therefore, patches from the dog (represented in blue) will be matched to the most similar (for example background patches which are represented in red) whereas the cat patches effectively match across the two views. With UOT, it is too costly to transport patches from the dog to the background and therefore they do not participate in the dense contrastive loss.

Method	AP	AP50	AP75
Global Only	46.42	72.45	49.74
DenseCL	**53.18**	**79.50**	**58.68**
Uniform	52.06	78.40	56.72
Unbalanced OT ($\lambda = 0.05$)	52.84	79.17	58.05
Unbalanced OT	52.50	78.96	57.21
Exact OT	52.38	78.38	58.03
Entropic OT ($\epsilon = 10$)	52.18	78.52	56.69
Entropic OT ($\epsilon = 1000$)	52.23	78.69	56.67

Table 3.5: Object detection results on the Pascal VOC 2007 test dataset.

3.3.3 Experiments

In order to experiment whether or not this optimal transport-based matching strategy improves the final performance of the model, we perform pre-training experiments on the COCO dataset (Lin et al., 2014). After pre-training the model with a mixture of dense contrastive loss and image level contrastive loss, the model is finetuned to the task of object detection on the Pascal VOC dataset (Everingham et al., 2015) on the 2007 and 2012 splits and finally the performance of the model is evaluated on the Pascal VOC 2007 test dataset. The backbone model is a ResNet50 (He et al., 2016) model which is complemented with the relevant head in order to be able to perform object detection.

The results of the experiments can be seen in Table 3.5. The baseline method is running only with the global level self-supervised loss for which we can see that the performance is not on par with model pre-trained on dense methods. However, when comparing the different matching strategy and their associated performance, we can see that the choice of matching strategy does not affect the performance in a significant manner. Moreover, choosing a completely uniform matching plan $\pi^{\star}_{ij} = \frac{1}{\mathrm{h} \times \mathrm{w}}$ does not result in a particularly worse performance compared to other OT-based methods. As such, it seems that indeed maximum is a good matching strategy proposed by Wang et al. (2021) as OT-based methods introduce hyper-parameters that are hard to tweak since training happens in a self-supervised setting where one cannot always get his hands on a validation set to finetune the hyper-parameters.

3.3.4 Discussion

Compared to the maximum similarity matching strategy proposed in DenseCL, OT based matching does not improve the final performance. Firstly, the added complexity and hyper-parameters introduced by the different variant make it hard to tune compared to the parameter-less maximum similarity matching. Because it is a self-supervised methods, it first must be pre-trained before being able to evaluate the performance needed to judge a method. As such, since dense downstream tasks are more computationally expensive, fewer experiments can be performed to investigate other options. Given our results, we can consider that since a uniform matching and exact optimal transport yield similar performances, any complete Optimal Transport method is unlikely to yield much better results, and that future efforts should be focused on the unbalanced variants. From this observation, we can question if this is a manifestation of the so-called "*Chicken and Egg*" problem where a sparse transport plan would penalize the model early in its training because of its inaccuracies, on the other hand a completely uniform transport plan gives no specific signal but to cluster all patches from the same image together which has a neutral effect with respect to the image-level objective. As such, one possible area of research would be to investigate if these transport based matchings could improve the performance of an encoder model already trained using the classical DenseCL method. Even if the benefits from OT to augment the DenseCL method are not so clear, there are still interesting research avenues for dense Self-Supervised Learning and Optimal Transport, especially with the recent rise in popularity of Vision Transformers (Dosovitskiy et al., 2020) for which the modeling of an image as a distribution of patches comes naturally (Sander et al., 2022).

3.4 Conclusion and Discussion

Its increasing popularity in machine learning makes of Optimal Transport a tool of choice when developing self-supervised methods. As such, we have investigated its usage in several key area of Self-Supervised Learning, being on the formulation of contrastive and non-contrastive methods, as a objective to enforce hyperspherical uniformity, or as a matching strategy between patches of augmented views for dense SSL. While the results are not always able to supersede state-of-the-art methods in their respective area, OT provides an interesting framework to tackle this issues. Since it is already successful in a variety of existing self-supervised methods such as SwAV (Caron et al., 2020) or Dino-

v2 (Oquab et al., 2023), we believe that optimal transport can still play an important role in the future of self-supervised representation learning.

Our proposed methods introduce additional hyper-parameters to the model which should be set for each dataset. However, running experiments in Self-Supervised Learning is quite costly since one needs to do the long pre-training before the evaluation benchmark. As such, the performance cannot be evaluated during the pre-training phase. This makes hyper-parameter tuning more compute intensive since there is no possibility of exiting early of failed experiments. Also, we consider that tuning extensively in order to maximize the performance on a single dataset to be oftentimes be too costly in term of compute but also of energy consumed for our capabilities. We note that energy consumed directly translates to carbon emissions which we would like to minimize in our research. For example, pre-training a DenseCL model requires more than 20 hours on a multi-GPU node. This computational cost has to be repeated for as many values of the hyper-parameters tried. Finally, we remark that outside of academic benchmark datasets, practitioners often do not have a complete enough training/validation set to perform a valid hyper-parameter tuning.

Chapter 4

Multi-Modal Contrastive Learning

Contents

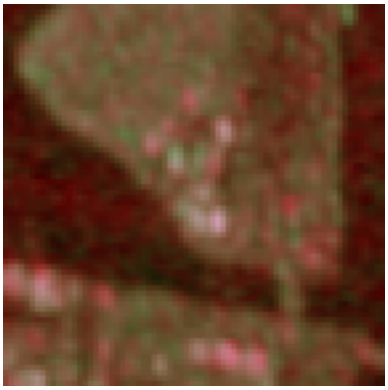
4.1 Uni-Modal Self-Supervised Learning in Remote Sensing . . . 71

4.2 Multi-Modal Self-Supervised Learning 76

4.3 Multi-Modal Supervised Contrastive Learning 81

4.4 Conclusion and Discussion 87

Following the background to self-supervised learning in Subsection 2.1, we now study a specific task that is common in remote sensing, multi-modal image classification. This section focuses on leveraging the multi-modal nature of remote sensing dataset to improve the representation learning performance in both self-supervised and supervised classification tasks. While most computer vision datasets and tasks focus on a single modality, remote sensing datasets are sometimes multi-modal as their images have been captured with different satellites. A multi-modal dataset has samples for which multiple modalities are available. For example, a common instance of multi-modal datasets in remote sensing is composed of images captured from the pair of Sentinel-1 and Sentinel-2 satellites. An example of such a pair can be seen in Figure 4.1. In this section, we investigate how SSL methods can be adapted to multi-modal remote sensing datasets. Interestingly, remote sensing multi-modal datasets are often aligned, meaning that for a given sample all modalities for the dataset are available. In this chapter, we explore and propose a method to leverage this specificity in order to learn good representations for these pairs and to perform multi-modal classification. First, in Section 4.1, we perform an analysis of the state of self-supervised methods applied to uni-modal remote sensing datasets. Next, in Section 4.2, we propose a framework for multi-modal self-supervised representation learning using a variant of contrastive learning. And finally, in Section 4.3, we generalize this framework and apply it to the setting of supervised learning, improving upon the baseline



(a) Sentinel-1.

(b) Sentinel-2.

Figure 4.1: Example of a multi-modal pair of Sentinel-1 and Sentinel-2 images for a sample of the *Waste Water Treatment* class extracted from the Meter-ML (Zhu et al., 2022) dataset.

finetuning using categorical cross-entropy.

4.1 Uni-Modal Self-Supervised Learning in Remote Sensing

Before evaluating the state of representation learning for multi-modal image classification, we first investigate existing self-supervised techniques applied to uni-modal remote sensing datasets. These datasets possess numerous differences with natural image datasets and as such can require specific self-supervised pre-training methods (Berg et al., 2022; Wang et al., 2022a) to reach improvements similar to those seen in the computer vision literature (Jing & Tian, 2020).

In order to evaluate the state of off-the-shelf pre-training methods for remote sensing, we perform self-supervised pre-training experiments on the Resisc-45 (Cheng et al., 2017) (high resolution scenes) and the EuroSAT (Helber et al., 2019) (low resolution Sentinel-2 scenes). They are not the largest datasets in the remote sensing domain but the most common ones that have been exploited in the literature for benchmarking, comparing and reproducing results. Resisc-45 contains 31500 RGB images extracted from Google Earth with 45 scene classes, each with 700 images. The images are of size 256×256 pixels and the spatial resolution varies from 0.3m to 30m depending on the scene class. However, the precise resolution of each image is unknown. In our experiments, we use the public

train/validation/test split proposed by Neumann et al. (2019) which uses 60% (more than 18k) of the images for training, 20% (more than 6k) for validation and 20% for test. It should be noted that Resisc-45 has very high intra-class and inter-class diversity, making it a common benchmark to evaluate and compare scene classification methods.

The EuroSAT dataset is collected from Sentinel-2 images and contains 10 scene categories. We also apply the public train/validation/test split proposed by Neumann et al. (2019) to perform experiments on this dataset, from which one could easily reproduce the results. There are 16200 (60%) training images, 5400 (20%) validation images and 5400 (20%) test images. The dataset is proposed with the 13 spectral bands captured from the Sentinel-2 sensor. Each image has the size of 64×64 pixels and a spatial resolution of 10m. The medium size and the interest in publicly-available Sentinel-2 data make EuroSAT a popular low-resolution dataset to conduct scene classification experiments.

For both datasets, we use only the RGB channels to be able to utilize methods from the literature which are presented in Section 2.1. We pre-train models using the SimCLR (Chen et al., 2020a) and MoCo-v2 (Chen et al., 2020b) for contrastive methods and BYOL (Grill et al., 2020) and Barlow Twins (Zbontar et al., 2021) for non-contrastive methods. To measure the performance of such methods, we use the linear evaluation protocol whereby models are first pre-trained in a self-supervised fashion and then a linear classifier is trained on top of the frozen pre-trained models.

After training the linear layer, we evaluate the top-1 classification performance and Cohen's Kappa coefficient (κ) (Cohen, 1960) which takes into account the probability of randomly selecting the ground truth value: $\kappa \in [-1, 1]$ with 1 meaning a perfect prediction and -1 means a completely different prediction. The numerical definition of the κ score can be seen in Appendix A.2. Results can be seen in Table 4.1. As observed, self-supervised methods perform much better than the randomly initialized model: 82.55%-85.37% compared to 45.45% on Resisc-45 and 92.59%-95.59.37% compared to 63.48% on EuroSAT. Meanwhile, the supervised ImageNet initialization performs on par or better than self-supervised pre-training due to the fact that this model was trained on a huge number of images (14 millions labelled images) against using only around 16k-18k images of each studied dataset. On Resisc-45, it yields an accuracy about 5% higher than the best performing SSL method (MoCo-v2). Nevertheless on EuroSAT, the two non-contrastive SSL models (BYOL and Barlow Twins), both perform better than the ImageNet initialization. One explanation could be the fact that the small size of EuroSAT images is not well-suited for supervised ImageNet models (initially pre-trained on 224×224 pixel images) who tend

to drop a lot of channels through the use of pooling layers. In the meantime, the SSL models have been adapted to handle the small size of 64×64 pixels as previously described in the experimental setup. EuroSAT also contains object which have a much smaller size than on ImageNet's natural images, thus requiring more fine-grained features. We note that similar behaviors have been observed by Jain et al. (2022) on these two datasets. In terms of comparing the four SSL methods, their behaviors are not the same on the two datasets. MoCo-v2 gives the best score on Resisc-45 but its performance is lower than BYOL and Barlows Twins on EuroSAT. Meanwhile, Barlow Twins performs very well on EuroSAT but stands behind MoCo-v2 and BYOL on Resisc-45.

Pre-training method	Resisc-45		EuroSAT	
	Acc.	$100 \times \kappa$	Acc.	$100 \times \kappa$
Random initialization	45.65±0.84	43.43±0.89	63.48±0.16	59.33±0.19
ImageNet supervised	**90.32±0.00**	**89.93±0.00**	94.46±0.00	<u>93.84±0.00</u>
SimCLR (Chen et al., 2020a)	82.55±0.68	81.84±0.71	92.59±0.05	91.76±0.05
MoCo-v2 (Chen et al., 2020b)	<u>85.37±0.15</u>	<u>84.78±0.15</u>	93.78±0.07	93.08±0.08
BYOL (Grill et al., 2020)	85.13±0.07	84.52±0.31	<u>94.92±0.12</u>	94.34±0.13
Barlow Twins (Zbontar et al., 2021)	83.14±0.30	82.44±0.31	**95.59±0.17**	**95.08±0.19**

Table 4.1: Classification performance (accuracy and *kappa* (κ) coefficient) on the Resisc-45 and the EuroSAT datasets under the linear evaluation protocol (3 runs). Best results are displayed in **bold** while the second best are <u>underlined</u>.

While the linear protocol is a good measure of the discriminative qualities of an encoder's representations, it is not the only commonly used benchmark for evaluating the performance of SSL methods. As such, we also test the pre-trained models under the benchmark of finetuning. This means that during the finetuning phase, model's weights are updated as well as those of a classifier on top of the representations. We perform this benchmark under a varying amount of labelled data, that is, by using only 1%, 10% and finally 100% of the available labels and samples.

The results from the finetuning experiments can be observed in Table 4.2. With this setting, we again confirm the improvement from using SSL pre-training compared to random initialization. Indeed, on the EuroSAT dataset with only 1% of labels, self-supervised methods perform better than the randomly initialized model fine-tuned with 10% of labels. Additionally, on the Resisc-45, self-supervised models also systematically yield a better performance than randomly initialized models. Again, the ImageNet supervised model outperforms the SSL models, especially on the high resolution Resisc-45 dataset

when fewer labels are available. Meanwhile, the gap is not as large on the lower-resolution EuroSAT due to the specific small size of the images. We also remark that when the number of labels increases (from 1% to 10%, then 100%), the performance gaps between the random initialization, SSL models and ImageNet supervised become closer. However in practice, finetuning strategy is usually adopted in the context of limited number of labels. When compared against the model trained from scratch with a random initialization, SSL methods are largely outperforming when the same number of labelled samples are available. Therefore, in case that the ImageNet initialization is not a trivial option due to weights not being available for a particular network architecture, using SSL pre-training is a reliable alternative to improve the training performance without the need for additional labels as also highlighted in the finetuning results from Jung et al. (2021). In our experiments, MoCo-v2 and BYOL exhibit similar performance levels while SimCLR is slightly under-performing. One reason might be the relatively low batch size of 512 used during the pre-training. MoCo-v2 does not suffer as much from this low batch size because the negative queue is still relatively large with 3072 samples. BYOL, being a non-contrastive method, also suffers less from smaller batch sizes than standard contrastive approach like SimCLR. We also find that the performance of the Barlow Twins pre-training seems to be more reliant on hyper-parameters (i.e. learning rate value and scheduler) than the other methods, leading to a more variable performance depending on the dataset and the percentage of labelled samples for training.

Pre-training method	Resisc-45			EuroSAT		
	1%	10%	100%	1%	10%	100%
Random initialization	32.39±0.69	67.68±0.39	91.05±0.32	53.64±0.38	76.76±0.67	96.49±0.03
ImageNet supervised	**58.79±0.29**	**89.27±0.28**	**96.75±0.03**	**85.62±0.21**	**95.43±0.11**	**98.70±0.02**
SimCLR (Chen et al., 2020a)	41.14±0.94	78.22±0.25	93.01±0.32	77.46±0.38	92.04±0.15	97.62±0.05
MoCo-v2 (Chen et al., 2020b)	<u>50.83±0.39</u>	<u>80.71±0.16</u>	93.45±0.38	<u>82.67±0.06</u>	93.06±0.18	98.15±0.07
BYOL (Grill et al., 2020)	49.30±1.64	78.92±0.11	93.39±0.20	82.74±0.43	<u>94.15±0.35</u>	<u>98.36±0.04</u>
Barlow Twins (Zbontar et al., 2021)	42.85±1.25	73.42±0.94	<u>95.03±0.26</u>	81.60±0.41	93.14±0.12	96.52±0.10

Table 4.2: Classification performance on the Resisc-45 and EuroSAT datasets under the finetuning evaluation protocol (3 runs). Pre-trained models are fine-tuned with a limited number of labelled samples (1%, 10% and 100% respectively). For each proportion of labelled data, the best results are displayed in **bold** while the second best are <u>underlined</u>.

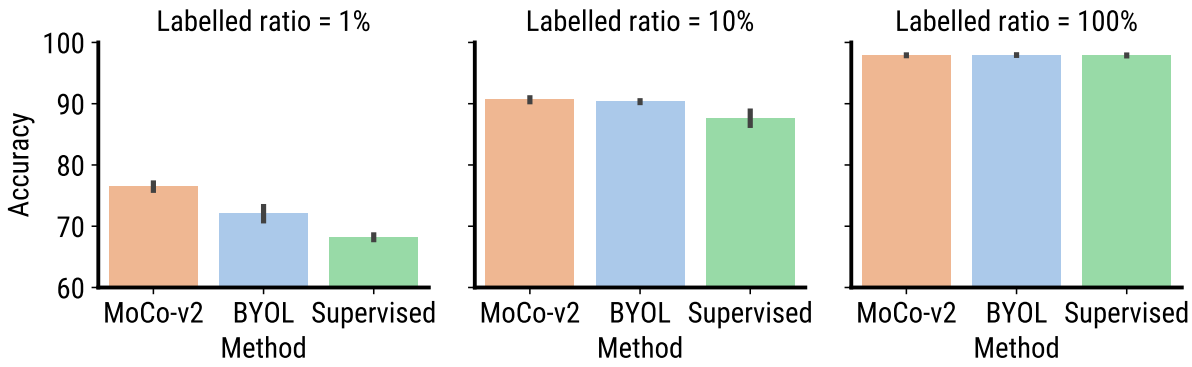
Transferring Representations. Following these experiments, we also investigate the capability of self-supervised pre-training in a transfer setting. That is, does a self-supervised model pre-trained on a dataset with similar but not identical visual features can be used on another dataset? To this end, we perform transfer learning experiments

where a model is first trained in a self-supervised fashion on a dataset and is then used as a weights initialization for a subsequent dataset.

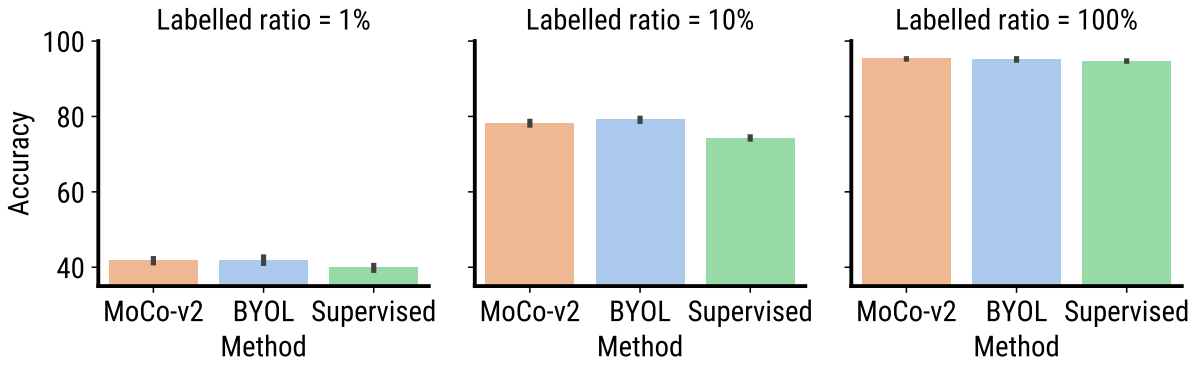
We compare this self-supervised model initialization with weights from a model that was trained in a supervised manner on the initial dataset. As for the previous experiments, we vary the percentage of labelled data. This variation highlights the relative performance gap between the different models initializations. Indeed, experiments using less labelled data will put more importance on the model initialization during finetuning. Experiments are performed again between the Resisc-45 and EuroSAT datasets both using RGB images. They both contain satellite images although with different resolutions and label sets. Therefore a model trained on one dataset is unable to make predictions on the other dataset.

The results for these experiments can be seen on Figure 4.2. The orange, blue and green colors represent the performance using MoCo-v2, BYOL and supervised method for pre-training respectively. Figure 4.2a shows the results observed when transferring models Resisc-45 $\rightarrow$ EuroSAT whereas Figure 4.2b shows transfer performance of EuroSAT $\rightarrow$ Resisc-45. One can see that indeed, when using a low amount of labelled data, a performance difference between self-supervised and supervised model as initialization is present. In fact, by training without labels, SSL approaches provide more generalized features which are more relevant in downstream tasks with a domain shift, whereas supervised models usually learn more dataset-specific representations. This confirms that self-supervised models have better transfer capabilities than their supervised counterparts. With larger quantities of labelled data, the gap between supervised and self-supervised model initialization shrinks as the finetuning reaches its optimal performance.

Beyond ImageNet initialization. While these experimental results showcase the performance of the so-called ImageNet initialization, pre-trained model weights are not always available especially when images are not composed of RGB channels and of size similar to ImageNet (224×224 pixels). Therefore, in the next sections of this chapter we set to explore the setting of multi-modal classification on multispectral images which have more than 3 channels and therefore where off-the-shelf models are not available to provide a strong baseline initialization.



(a) Pre-training on the Resisc-45 dataset and finetuning/evaluating on the EuroSAT dataset.



(b) Pre-training on the EuroSAT dataset and finetuning/evaluating on the Resisc-45 dataset.

Figure 4.2: Comparison of finetuning performance of supervised and self-supervised models on another dataset.

4.2 Multi-Modal Self-Supervised Learning

4.2.1 Methodology

In Section 4.1, experiments show that ImageNet initialization remains a strong baseline when pre-trained models are available. However, they are not always available as is the case for multi-modal datasets where a sample is composed of multiple views from different sensors. These sensor captures have been co-registered to cover the same location. One task for such datasets is then to classify the nature of the sample from the multiple captures. These pairs can then be seen as different views of the same location and incorporated in a self-supervised learning pipeline. In this section, we build upon existing works (Jain et al., 2022; Scheibenreif et al., 2022b; Wang et al., 2022b) which focus on learning on only two modalities to develop a framework for self-supervised pre-training

across several image modalities. In our experiments, we work with up to three but there could be an arbitrary number of modalities if available.

In this section, we focus on contrastive methods due to their popularity in SSL. Also, they have been until now the most widely used SSL approaches in remote sensing (Berg et al., 2022). The most common version of the contrastive loss is called NT-Xent (Chen et al., 2020a). It maximizes the alignment between positive views from the same sample while minimizing the alignment with views from other samples present in the same batch using a softmax cross-entropy. Given an encoder model $f_\theta(\cdot)$ parameterized by θ, the loss for a single view is the NT-Xent shown in Equation 2.4.

As a reminder, the NT-Xent loss (for normalized temperature cross-entropy), described in Equation 2.4, is the result of the cross-entropy between the distribution generated by the softmax over distances and a pseudo-label of positives such that only the distance with respect to positive samples is diminished and the distance with negative samples is increased when minimizing the loss.

To adapt the contrastive learning framework to multi-modal datasets, we propose to treat a multi-modal sample as multiple augmented views of the same sample. In this case, the sensing with different sensors can be seen as a form of augmentation since the underlying ground truth label remains the same. Moreover, one can still use random augmentations such as color jittering but with caution since remote sensing images may not be as augmentable as natural images. For example, applying color jittering on a multispectral image has no guarantee to generate a plausible multispectral image with the same label.

One consideration about computing representations for multi-modal images is that more often than not, the multiple modalities composing a remote sensing dataset are heterogeneous. That is, they do not have the same resolution and the same channels. As such, they cannot be directly encoded using the same backbone model. Hence, we propose to use an encoder per modality and train each model to project in a common latent representation space which becomes domain and augmentation invariant.

In a setting with M modalities, the joint learning process is as follows. After encoding each image using a modality-specific image encoder $f_m(\cdot), m \in [1, M]$ and projecting the feature representations into the hypersphere (*i.e.* feature space) $z_{i,m} = \frac{f_m(x_{i,m})}{\|f_m(x_{i,m})\|}$ (*i.e.* normalized to unit vectors). We now consider $\mathrm{z}_i = \mathrm{cat}(z_{i,1}, \ldots, z_{i,M})$ the set of all fused representations in the batch. For a view representation z_i, we call its set of multimodal positives $P_{\mathrm{MM}}(i)$ which contains the indices in z of representations of the same sample but

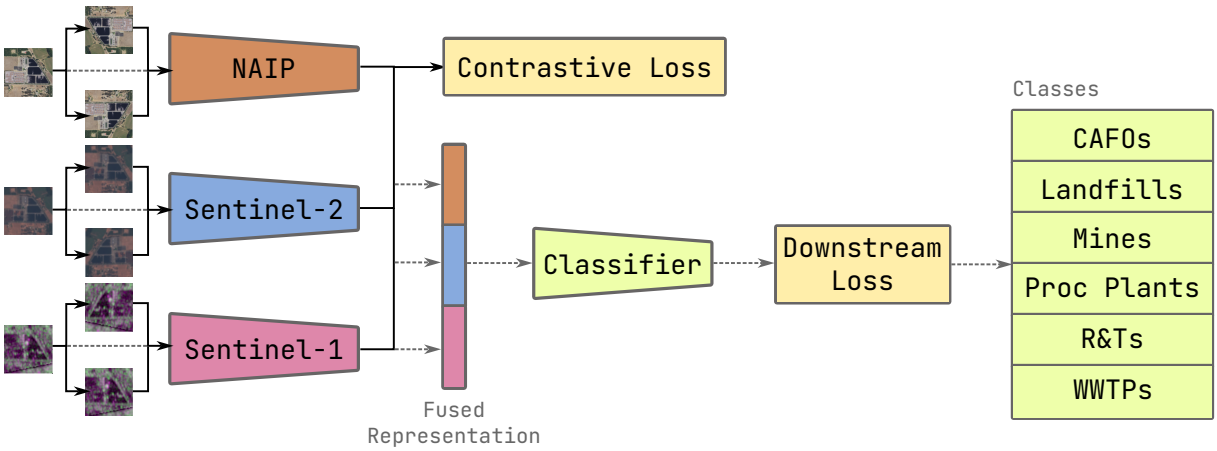


Figure 4.3: Architecture of our multi-modal pre-training and finetuning processes. Black arrows represent the forward data flow during pre-training while dashed gray arrows represent the forward data flow during finetuning. To ease reading, we illustrate a specific use-case of methane source classification using the from the Meter-ML (Zhu et al., 2022) dataset with three modalities: NAIP, Sentinel-2 and Sentinel-1 images. In the downstream classification task, methane emitting sources are divided into six classes.

from other modalities. The pre-training loss used for multimodal datasets is the following for a batch of N samples:

$$\mathcal{L}_{\text{MM-SSL}} = \sum_{i=1}^{N} \frac{-1}{|P_{\text{MM}}(i)|} \sum_{j \in P_{\text{MM}}(i)}^{M} \log \frac{\exp(\langle \mathrm{z}_i, \mathrm{z}_j \rangle / \tau)}{\sum_{k \neq i} \exp(\langle \mathrm{z}_i, \mathrm{z}_k \rangle / \tau)}, \tag{4.1}$$

where $\tau > 0$ is a temperature parameter used to control the overall sharpening of the distribution produced by the softmax operator. Due to its construction, Equation 4.1 is very similar to Equation 2.4 which defines the NT-Xent loss as used in (Chen et al., 2020a). Our overall architecture can be seen in Figure 4.3.

After pre-training encoder models with $\mathcal{L}_{\text{MM-SSL}}$, the f_m backbones can be used on their own as a discriminative initialization after self-supervised pre-training. During the downstream tasks, a single feature can be used for a geographical location by fusing representations from each input modalities. We choose to generate these image-level representations from the multiples modality-specific representations by concatenating. Though the fusing method could be improved in future work on multi-modal representation learning.

4.2.2 Experiments

Pre-training Experiments. In order to evaluate the performance of our proposed framework, we perform a series of multi-modal self-supervised pre-training with the different combinations of modalities available in the Meter-ML (Zhu et al., 2022) dataset. Then after pre-training, we perform a final finetuning on a subset or all modalities used during the pre-training and evaluate the performance on the test set of the dataset.

The Meter-ML (Zhu et al., 2022) dataset used in our experiments for methane source classification contains multiple modalities for each geographic coordinate. Each methane-emitting present facility in the dataset accordingly has corresponding Sentinel-1, Sentinel-2 and NAIP sensor captures. To experiment with both optical and SAR data as well as both low and high resolution, we pick sensor views from Sentinel-1 (VH and VV) and Sentinel-2 (RGB and NIR) at 10-m resolution as well as NAIP (RGB and NIR) at 1-m resolution. The proposed architecture is composed of a backbone for each modality used during pre-training. We use an AlexNet (Krizhevsky et al., 2017) for Sentinel-1 (S1) and Sentinel-2 (S2) and a ResNet18 (He et al., 2016) model for NAIP. All experiments are implemented using the Pytorch (Ansel et al., 2024) deep learning framework. To evaluate the impact of artificial augmentations, we compare self-supervised pre-training with artificial augmentations and without. When artificial augmentations are used, we generate two versions of the same image for each modality using random augmentations. Our set of augmentations includes random horizontal and vertical flips and a random cropping with a conservative resized crop with a scale of at least 90% of the original image. With artificial augmentations, it results in each augmented view having a randomly augmented positive in the same modality as well as multiple augmented positives in other available modalities whereas without artificial augmentations, the view from a modality only has corresponding positives in other modalities. The Negative class samples present in the Meter-ML dataset are only used as negatives in the contrastive loss (see Equation 4.1). Models are pre-trained for 120 epochs. When using multiple modalities during finetuning, representations from each backbone are concatenated to produce a single feature vector for the sample which is then fed to the classifier. For methane source classification, we finetune models for 100 epochs. The training and validation sets are set following the official split of the Meter-ML dataset. Results can be seen in Table 4.3.

From the table, self-supervised pre-training consistently improves the performance compared to randomly initialized models. The multi-backbone architecture also scales with the number of modalities even when certain modalities are removed for the down-

Pre-training	Downstream						
	S1	S2	NAIP	S1 + S2	S1 + NAIP	S2 + NAIP	S1 + S2 + NAIP
None	47.37%	64.29%	62.03%	65.04%	63.16%	68.42%	65.79%
S1	51.13%	-	-	-	-	-	-
S2	-	70.30%	-	-	-	-	-
NAIP	-	-	66.92%	-	-	-	-
S1 + S2	53.76%	71.80%	-	71.80%	-	-	-
S1 + NAIP	56.39%	-	70.68%	-	**72.18%**	-	-
S2 + NAIP	-	71.43%	68.42%	-	-	72.18%	-
S1 + S2 + NAIP	**58.65%**	**72.56%**	**72.93%**	**72.18%**	68.80%	**73.31%**	**73.68%**
S1 + S2 ⋆	50.00%	69.55%	-	65.04%	-	-	-
S1 + NAIP ⋆	55.26%	-	70.30%	-	60.90%	-	-
S2 + NAIP ⋆	-	**72.56%**	69.92%	-	-	72.93%	-
S1 + S2 + NAIP ⋆	52.63%	71.05%	69.92%	65.41%	65.79%	69.92%	69.17%

Table 4.3: Accuracy using different pre-training and finetuning scenarios. When pre-training is None, it refers to randomly initialized baseline models. Pre-trainings annotated with ⋆ denotes using no random augmentations and using only view (the original) by modality during pre-training. In this setting, the self-supervised pre-training cannot be done on a single modality.

stream task. As an example, pre-training with Sentinel-1 and Sentinel-2 and then performing finetuning and evaluation on Sentinel-1 alone yields a final performance of 53.76% whereas performing only on Sentinel-1 yields a performance 51.13%. The best overall performance is obtained when combining all modalities for the downstream classification task. Each modality therefore contains information relevant to the classification task that the methane source classifier is able to exploit. It is interesting to highlight that using fusion only during the downstream task and with a random initialization leads to a worse performance of 65.79% than when using only Sentinel-2 data during pre-training and finetuning which reaches 70.30% of accuracy. This means that this modality could provide the most important information for classification and that self-supervised pre-training allows a discriminative initialization, which gives a better performance on the scene classification downstream task.

Our experiments without artificial augmentations show interesting results. Mainly, when pre-training with modalities which are different in nature like SAR and optical data (S1+S2, S1+NAIP), the results are better with artificial augmentations. This phenomenon suggests that the single positive from the other modality is hard to align to. Indeed, adding artificial augmentations improves the pre-training performance by also offering in-modality positives. For example, when pre-training on the three available

modalities and finetuning on the same three modalities, the performance increases from 69.17% to 73.68% when adding random augmentations to the pre-training process. With only optical pairs from different modalities (when pre-training with Sentinel-2 and NAIP for example), the drop in performance from removing random augmentations is less severe, dropping from 72.18% to 69.92%. Consequently, in this case, artificial augmentations have less impact on the downstream performance. In any case, the pre-training performance with only a single modality and random augmentations performs worse than pre-training with multiple modalities, but remains better than random initialization for our chosen downstream task of scene classification with finetuning.

Now that we have successfully experimented with a multi-modal pre-training method, we generalize this scheme to the supervised setting in Section 4.3. This generalization enables leveraging label information if available in the final finetuning phase which currently only uses categorical cross-entropy as its classification objective.

4.3 Multi-Modal Supervised Contrastive Learning

4.3.1 Methodology

Following section 4.2 which introduces a self-supervised framework to pretrain multi-modal models on multi-modal dataset, we apply this formulation to the supervised setting. Introduced by Khosla et al. (2020), Supervised Contrastive (SupCon) learning brings the representation learning lessons from self-supervised learning to supervised representation learning where labels are available. With the extra supervision signal, the core principle of SupCon learning is to extend the set of positives with samples from the same class when computing the contrastive loss. This addition prevents a known problem with SSL methods where negatives are actually picked from the same ground truth class and pushing negatives far away which thus reduces the clustering of samples from the same class. More optimal solutions for negative sampling have been explored in Chuang et al. (2020) and Robinson et al. (2020).

Figure 4.4 depicts our supervised contrastive framework between samples from two different classes each with two modalities. Samples from the same class are pulled closer together whereas negative samples from a different class are pushed further away. The hypersphere is used as a support for the learnt representations. In SupCon, Khosla et al. (2020) propose to extend the set of positives used for contrastive learning with all samples

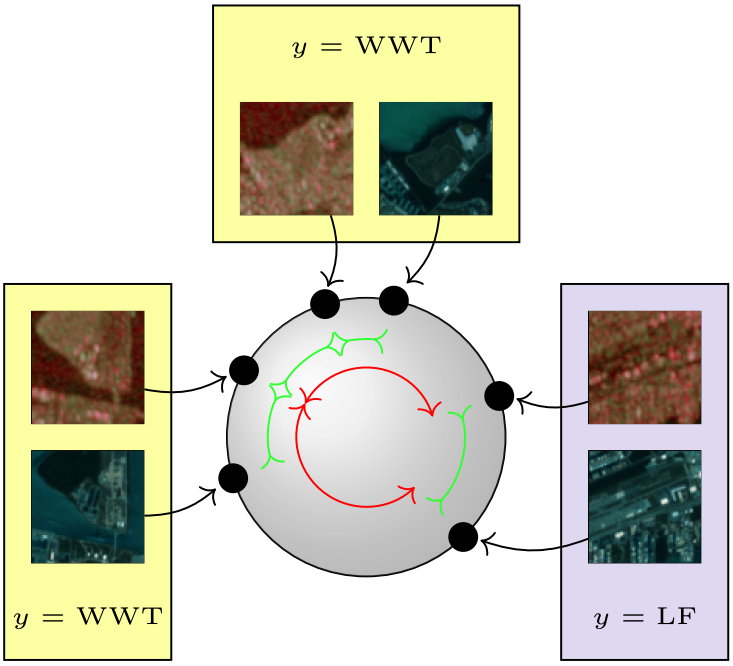


Figure 4.4: Example of our proposed multi-modal supervised contrastive learning framework with samples from the class *Waste Water Treatment* (WWT) and *Landfill* (LF) optimized on the unit-circle $\mathbb{S}^1$.

from the same class.

We define $P_{\mathrm{S}}(i) = \{j; y_j = y_i, j \neq i\}$ as the set of all images from the same class as given sample i. The SupCon method therefore uses this set of positives for an image instead of randomly augmented versions from the same image. The SupCon loss is defined similarly to the NT-Xent loss from Equation 2.4 but using P_{S}:

$$\mathcal{L}_{\text{SupCon}} = \sum_{i=1}^{N} \frac{-1}{|P_{\mathrm{S}}(i)|} \sum_{p \in P_{\mathrm{S}}(i)} \log \frac{\exp(\langle z_i, z_p \rangle / \tau)}{\sum_{k \neq i} \exp(\langle z_i, z_k \rangle / \tau)}, \tag{4.2}$$

In our multi-modal extension of supervised contrastive learning, we leverage this set and extend it with all views from all modalities. Formally, we define the set of positives for a sample i as $P_{\text{MM-S}}(i)$:

$$P_{\text{MM-S}}(i) = P_{\text{MM}}(i) \cup \bigcup_{j \in P_{\mathrm{S}}(i)} P_{\text{MM}}(j). \tag{4.3}$$

$P_{\text{MM-S}}(i)$ is therefore composed of all views of the samples belonging to the same class. Using this positive set for each sample, we can define the multimodal supervised

contrastive loss by once again embedding the multimodal images and concatenating them in a single vector z as done in Section 4.2 (Equation 4.1) as follows:

$$\mathcal{L}_{\text{MM-S}} = \sum_{i=1}^{N} \frac{-1}{|P_{\text{MM-S}}(i)|} \sum_{j \in P_{\text{MM-S}}(i)} \log \frac{\exp(\langle \mathrm{z}_i, \mathrm{z}_j \rangle / \tau)}{\sum_{k \neq i} \exp(\langle \mathrm{z}_i, \mathrm{z}_k \rangle / \tau)}. \tag{4.4}$$

With this definition, the standard SupCon (Khosla et al., 2020) becomes the specific case with a single modality ($M = 1$) where $P_{\text{MM}}(i) = \{i\}$. Moreover, the joint multimodal self-supervised learning (Berg et al., 2023) presented in Section 4.2 becomes another specific case where each image has a different ground-truth label ($y_i = i \Leftrightarrow P_{\text{S}}(i) = \varnothing$).

While $\mathcal{L}_{\text{MM-S}}$ is a representation learning loss, it also requires a classification loss to evaluate the quality of representations and perform classification inference. We therefore choose to combine this loss with the categorical cross-entropy:

$$\mathcal{L}_{\text{train}} = \alpha \mathcal{L}_{\text{MM-S}} + (1 - \alpha) \mathcal{L}_{\text{CE}}, \tag{4.5}$$

where $0 \leq \alpha < 1$ is the weighting coefficient to balance the importance of the two loss terms. $\mathcal{L}_{\text{CE}}$ refers to the categorical cross-entropy loss. This loss can be seen as combining a representation learning objective with a classification objective. In our experiments, we set $\alpha = 0.5$ to train the proposed framework, giving equal importance to each term. An overview of the complete training architecture can be seen in Figure 4.5. Additionally, by setting $\alpha = 0$, one can ignore the multimodal SupCon term to only train using the standard cross-entropy loss for classification, falling back to the well-known categorical cross-entropy training on the representations obtained from fusing the multi-modal representations.

4.3.2 Experiments

In order to evaluate the improvement that multi-modal supervised contrastive learning can bring to the table, we conduct experiments on multimodal scene classification using two public datasets, including the 2020 IEEE Datafusion Contest (Yokoya et al., 2020) (DFC2020) and the Meter-ML (Zhu et al., 2022) dataset. Following the setting of Scheibenreif et al. (2022a), for the first experiment on DFC2020, the pre-training phase was performed on the large-scale SEN12MS (Schmitt et al., 2019) dataset which contains 180,662 pairs of spatially-aligned Sentinel-1/Sentinel-2 observations without labels. Then, in the downstream task, 986 and 5,128 paired Sentinel-1/2 from DFC2020 were

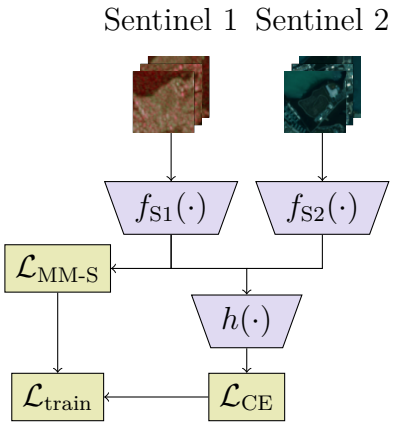


Figure 4.5: An overview of multi-modal supervised contrastive training pipeline on a Sentinel-1/Sentinel-2 pair multi-modal dataset.

used for finetuning and test, respectively. This dataset involves 8 land-use scene classes including *Forest, Shrubland, Grasland, Wetland, Cropland, Urban/Built-up, Barren and Water*. For both datasets, we use the official train/test splits. For a fair comparison to the baseline in Scheibenreif et al. (2022a), we adopted the ResNet-50 model as encoder and exploited the pre-trained weights provided by the authors using their multimodal SSL pretraining approach. With the finetuning, we also used the same hyper-parameters as the baseline. We note that the authors also leveraged the Swin Transformers as encoder in their experiments. As it achieved lower performance than ResNet-50 in finetuning (Scheibenreif et al., 2022a), we did not consider Swin Transformer backbone in this work.

Regarding the second dataset which is Meter-ML (Zhu et al., 2022), which contained three image modalities including Sentinel-1/2 and NAIP observations. In this study, we only consider pairs of Sentinel-1/2 images to perform the downstream classification task to show the impact of SupCon learning. Nevertheless, one can easily extend this approach to more modalities. For experimental settings, we follow the settings set in Section 4.2, including the use of AlexNet as encoder as well as the number of epochs, *i.e.* set to 120 and 100, used for SSL pre-training and finetuning, respectively.

For both experiments on the two datasets, we perform the downstream finetuning using the combined loss function in Equation 4.5 and set $\alpha = 0.5$ to balance the two loss terms. We set $\alpha = 0$ to yield the baseline performance using only the cross-entropy loss. Each experiment is conducted for 5 runs. As previously mentioned, to study the impact of the multimodal SupCon term, we also conduct experiments on the Meter-ML dataset to perform a sensitivity analysis to parameter α in Equation 4.5.

The results for the downstream task on the DFC2020 dataset can be observed in Table 4.4. We also report the performance from the paper (Scheibenreif et al., 2022a) for a comparison, although our reproduced results were marginally higher. The table shows that the proposed multimodal SupCon learning provided better performance than the standard use of categorical cross-entropy. The overall accuracy was significantly increased with a gain of 4.33% (72.62% compared to 68.29%) and the average accuracy over all classes was also improved for about 1.5% (55.87% compared to 54.39%). This confirms the effectiveness of the proposed frameworks, in particular the use of supervised contrastive loss to better leveraging class information and multiple data modalities during the finetuning phase.

Class	CE (Scheibenreif et al., 2022a)	CE (Our impl.)	**SupCon+CE (Proposed)**
Forest	65±8	63.49±5.90	**75.54±5.01**
Shrubl.	56±11	**58.31±4.59**	52.26±4.42
Grassl.	9±6	**12.60±7.53**	6.18±4.93
Wetland	15±8	14.43±11.55	**20.90±7.11**
Croplan	45±8	52.65±7.94	**59.65±5.07**
Urban	**95±1**	91.14±3.48	89.94±2.63
Barren	39±3	43.91±2.7	**44.09±3.49**
Water	**99±1**	98.63±0.41	98.42±1.25
Overall	67±2	68.29±1.66	**72.62±1.31**
Average	53±2	54.39±0.72	**55.87±0.89**

Table 4.4: Top 1 Accuracy Performance on the DFC2020 dataset. We compare our proposed multi-modal supervised contrastive objective with categorical cross-entropy finetuning used in Scheibenreif et al. (2022a) which we have reimplemented.

From Table 4.5, the classification results on the Meter-ML dataset again confirms the superior performance of the proposed approach. We observe that for a large majority of classes (*i.e.* 5 out of 6 classes), exploiting the multimodal supervised contrastive learning significantly improves the final downstream performance. A gain of more than 3% was finally achieved for the average accuracy (73.83% compared to 70.74%). We note that, as done in Berg et al. (2023), the *Negative* class was excluded from the evaluation since it contains samples which are not methane emitting facilities.

Finally, to enrich our experiments and demonstrate the consistency of the proposed method to other supervised classification loss functions. We have performed a comparative

Class	CE (Our impl.)	**SupCon+CE (Proposed)**
CAFOs	97.44 ± 1.59	**97.44 ± 0.95**
Landfills	50.23 ± 3.15	**59.53 ± 4.82**
Mines	**60.00 ± 5.24**	59.00 ± 8.40
Proc Plants	57.36 ± 5.10	**66.31 ± 5.71**
R&Ts	77.91 ± 3.99	**78.61 ± 1.97**
WWTs	81.50 ± 2.00	**82.10 ± 4.20**
Average	70.74 ± 1.74	**73.83 ± 2.00**

Table 4.5: Top 1 Accuracy Performance on the Meter-ML dataset. Our proposed method is compared with categorical cross-entropy (CE) finetuning.

Method	Accuracy
WCE Loss	71.23±0.69
SupCon + WCE (ours)	**74.74±0.55**
Focal Loss	71.50±0.93
SupCon + Focal (ours)	**75.19±0.76**

Table 4.6: Accuracy performance on the Meter-ML dataset when combining our proposed multi-modal SupCon loss with other finetuning losses. The weighted cross-entropy loss is weighted inversely to the occurrence of a class in the training dataset.

study by combining our SupCon loss with the Weighted Cross-Entropy (WCE) loss and the Focal loss (Lin et al., 2017). We note that these two loss functions have been proved to be more effective than the standard CE when dealing with unbalanced classes, which is the case of the Meter-ML dataset. Table 4.6 shows the significant improvement by using our approach, confirming the its consistency to this multimodal classification task.

These results show the impact of leveraging the supervised signal to improve representations when training the downstream task classifier. We can imagine using both self-supervised and supervised multi-modal contrastive learning as part of a training pipeline where a large subset of the data is not annotated and therefore self-supervised pre-training can play an important role in initializing the model's weights such that it has good initial performance.

On the sensitivity of the α parameter. The α hyper-parameter in Equation 4.5 is used to weight between the classification loss ($\mathcal{L}_{\text{CE}}$) which impacts both the classifier and the weights of the encoder models and the multi-modal supervised contrastive loss

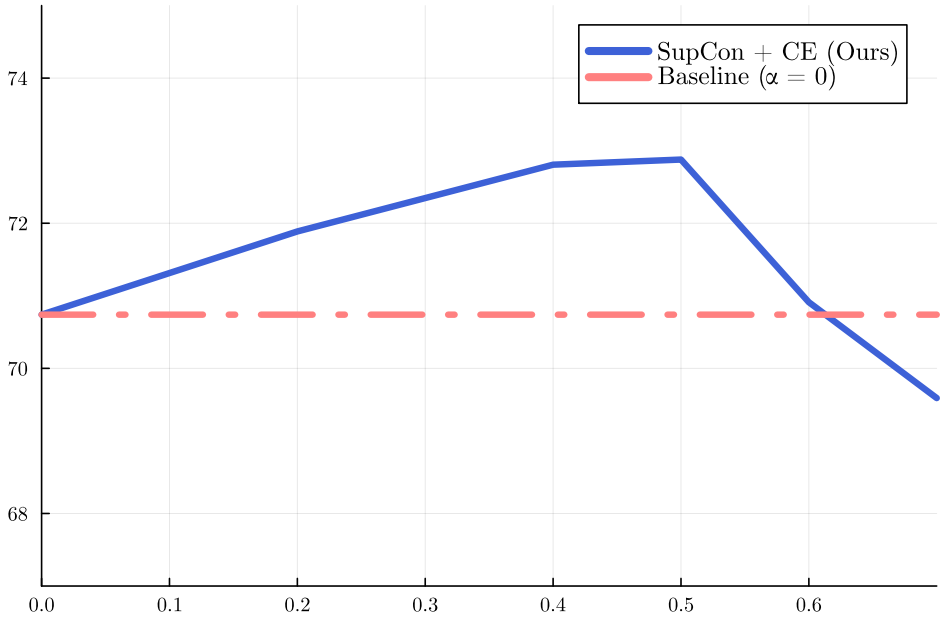


Figure 4.6: Performance on the Meter-ML dataset (Zhu et al., 2022) of our proposed supervised multi-modal framework when varying the value of the α hyper-parameter in the formulation of $\mathcal{L}_{\text{train}}$.

$\mathcal{L}_{\text{MM-S}}$. Naturally, $\alpha = 0$ implies $\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{CE}}$ which corresponds to the baseline finetuning which we compared our method to in Table 4.4 and 4.5. Conversely, if one were to choose $\alpha = 1$, then there would be no more classification loss and one could expect the resulting classification performance to be that of a random classifier since the classifier would not be impacted by the gradient of $\mathcal{L}_{\text{train}}$. Since our results for Table 4.4 and 4.5 were achieved with $\alpha = 0.5$, we suspect that there exists an optimal value for $\alpha \in [0, 1[$. As such, we run an ablation study by performing multiple run under the same protocol but by varying the value of α in the interval $[0, 1[$. The results of this ablation study can be seen in Figure 4.6. With this study, we see that the optimal value for the Meter-ML dataset is indeed $\alpha = 0.5$ and that the performance decreases rapidly if α is too big.

4.4 Conclusion and Discussion

In this chapter, we covered the task multi-modal image classification which can occur in remote sensing datasets. To this end, we propose a self-supervised framework to learn discriminative representations for multi-modal datasets. Secondly, the proposed framework

can be generalized to the setting of Supervised Contrastive learning (Khosla et al., 2020). When labels are available, this improves upon the commonly used cross-entropy finetuning used to prepare a self-supervised model to solve a downstream classification task. In the future, one could explore the task of multi-modal classification in cases where remote sensing images are not always co-registered (*i.e.* all modalities are not always available for a sample). One could draw on the foundation model research where multi-modal language and text models share a common latent space (Mizrahi et al., 2024; Radford et al., 2021). This proves to be an harder problem than the one studied in this chapter but we hope that our work is a step in the right direction for the research in multi-modal scene classification.

Chapter 5

Horospherical Learning with Hierarchical Prototypes

Contents

5.1 Hyperbolic Classification using Horospheres 90

5.2 Hierarchical Initialization 92

5.3 Experiments 97

5.4 Conclusion and Discussion 103

In this chapter, we consider the problem of classifying samples whose labels can be organised in a hierarchy. A so-called hierarchical dataset can be built by grouping labels which have similar semantic meaning under a common ancestor. For example, the ImageNet (Russakovsky et al., 2015) dataset labels can be organised based on the WordNet (Miller, 1998) taxonomy. Hierarchical groupings can also be based on the biological taxonomic rank as is the case for the CUB200 dataset (Wah et al., 2011) and the iNaturalist 2019 dataset (Van Horn et al., 2018). In other cases, such as when dealing with the task of semantic segmentation, the hierarchy can come naturally from the spatial inclusion of one class into its superclass. Therefore, simply by varying the fine graining level, one can build a hierarchy of labels. For example, in a semantic segmentation dataset, the superclass *Human body* can have children classes for each of the corresponding body parts.

To that end, we introduce a hyperbolic classification scheme that can then be used for image classification and semantic segmentations tasks. On its own, this hyperbolic classification method does not leverage the hierarchical nature of the data apart from using hyperbolic embeddings to perform the classification. To explicitly leverage the tree likeness of hyperbolic spaces, we then propose a hierarchically informed scheme to position ideal prototypes which are used as parameters of our horospherical classifier. In this initialization scheme, prototypes corresponding to labels near each other in the label

hierarchy should also be positioned nearby in the ideal prototype embedding space which is the hypersphere in the case of using the Poincaré ball model to represent hyperbolic embeddings.

5.1 Hyperbolic Classification using Horospheres

In Subsection 2.2, we introduced the Busemann function as a measure of distance between points embedded in hyperbolic space and points located at the infinity (*i.e.* ideal points). This function was successfully leveraged by Ghadimi Atigh et al. (2021) to propose a hyperbolic classification scheme based on ideal prototypes. In their proposed framework, a backbone encoder model is trained to minimize the Busemann distance between an image representation and the prototype corresponding to its class label. During inference, the prototype whose Busemann distance function with a sample is minimal is taken as the prediction.

We propose to build upon their work and introduce another hyperbolic classification layer based on the Busemann function. More specifically, the level-sets of the Busemann function are called *horospheres* (or *horocycles* in two dimensions). Given an ideal point p, a horosphere is said to be centered at p if it is generated as the level-set of B_{p}.

In our work, we propose to update the prototype's position and add an additional scalar parameter which parameterizes the level-set generating this horosphere. This parameter acts analogously to the bias term in Euclidean hyperplanes as it acts on the radius of the horosphere. This allows the horospherical classifier to model a larger set of decision boundaries. Therefore, a parameterized horosphere can be defined from its prototypes $\mathrm{p} \in \mathbb{S}^{d-1}$ and its bias term $a \in \mathbb{R}$. We define this horosphere level-set as:

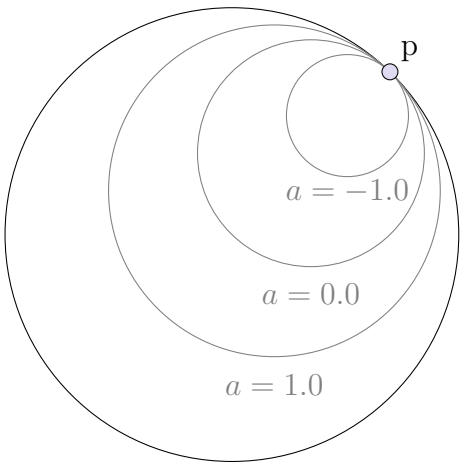


Figure 5.1: Multiple horocycles centered at the same ideal point p in the Poincaré disk.

$$H_{\mathrm{p},a} = \{x \in \mathcal{H}^c_d, -B_{\mathrm{p}}(x) + a = 0\}, \qquad (5.1)$$

where B_{p} is the Busemann function (see Equation 2.17). In the Poincaré ball model, such a level-set corresponds to an Euclidean hypersphere which is tangent to the Poincaré ball at p and has a radius of $r(a) = \frac{1+\tanh(a/2)}{2}$ (see Supplementary A.3). Examples of different horocycles in the Poincaré disk can be seen in Figure 5.1.

In order to prevent samples from moving too close to the boundary, one common solution is to use Euclidean clipping before projecting in hyperbolic space (Guo et al., 2022) to restrict representations to only a subset of the Poincaré ball. However even with this clipping, the majority of representations end up close to the new boundary such that only comparing samples based on their orientation can be enough (Moreira et al., 2023). In Ghadimi Atigh et al. (2021), the authors instead use an additional regularization term which prevents representations' radii from growing too close to the boundary. We include a similar regularization, not as a loss term, but directly in the logit $\xi_{\mathrm{p},a}$ for an hypersphere parameterized with p, a:

$$\xi_{\mathrm{p},a}(x) = -B_{\mathrm{p}}(x) + a + \phi(d) \cdot \log\left(1 - \|x\|_2^2\right), \tag{5.2}$$

where $\phi(d)$ weights the impact of this regularization depending on the dimension d of the hyperbolic space $\mathcal{H}_d^c$. Adding this regularization to the scoring function $\xi_{\mathrm{p},a}$ penalizes points from moving too close to the boundary and therefore the level set $\{x \in \mathcal{H}_d^c, \xi_{\mathrm{p},a}(x) = 0\}$ is not strictly an hypersphere tangent to the Poincaré ball at point p. Note that with $\phi(d) = 0$, we fall back to the parameterized definition from Equation 5.1. In practice and similar to Ghadimi Atigh et al. (2021), we use a single scalar hyper-parameter $\lambda \geq 0$ which is fixed during training to parameterize regularization such that $\phi(d) = \lambda \times d$. We provide an analysis of the sensitivity of this parameter in Appendix A.3.

While the logit function $\xi_{\mathrm{p},a}$ defined in Equation 5.2 can be used to perform binary classification by computing $P(\hat{y} = 1) = \sigma(\xi_{\mathrm{p},a}(x))$, with $\sigma : \mathbb{R}^d \to [0, 1]$ being the logistic function, it can also be adjusted for multi-class predictions. Indeed, horospheres can be adapted in a multi-class classification setting by having a horosphere per class and using a softmax normalization to obtain class membership probabilities. A multi-class horospherical layer consists of K horospheres and we define the probability of a sample $x \in \mathcal{H}_d^c$ with respect to one of the K classes as the softmax over the resulting K logits:

$$P(\hat{y} = k \mid x, \mathrm{p}, a) = \frac{\exp(\xi_{\mathrm{p}_k,a_k}(x))}{\sum_{l=1}^{K} \exp(\xi_{\mathrm{p}_l,a_l}(x))}, \forall k \in \{1..K\}. \tag{5.3}$$

This horospherical multi-class classifier can then be trained using the negative log-

likelihood loss over the training data x, y:

$$\mathcal{L}_{\mathrm{p},a}(x, y) = \frac{1}{N} \sum_{i=1}^{N} -\log P(\hat{y} = y_i \mid x_i, \mathrm{p}, a). \tag{5.4}$$

During training the ideal prototypes position is updated using stochastic Riemannian gradient descent (Bonnabel, 2013) which leverages the exponential map to project gradients to the manifold (see Subsection 2.2) whereas Euclidean parameters from the backbone are optimized using regular Stochastic Gradient Descent (SGD). There is a Riemannian version of the Adam (Kingma & Ba, 2014) optimizer proposed in (Bécigneul & Ganea, 2018) which can be used instead of the Riemannian gradient descent to keep prototypes on the hypersphere. Even though the prototypes are allowed to move during the training in order to minimize the training loss, it can be costly or impossible for multiple prototypes to exchange positions on the hypersphere. As such, the initial position of ideal prototypes can largely impact the training process. Therefore, we propose in Subsection 5.2 an initialization scheme to improve the downstream classification performance of the horospherical classifier.

5.2 Hierarchical Initialization

(a) *Pine Warbler.* (b) *Prairie Warbler.* (c) *Sooty Albatross.*

Figure 5.2: Example birds from the CUB200 (Wah et al., 2011) dataset. 5.2a and 5.2b both are in the *Parulidae* biological family whereas 5.2c is in the *Diomedeidae* family.

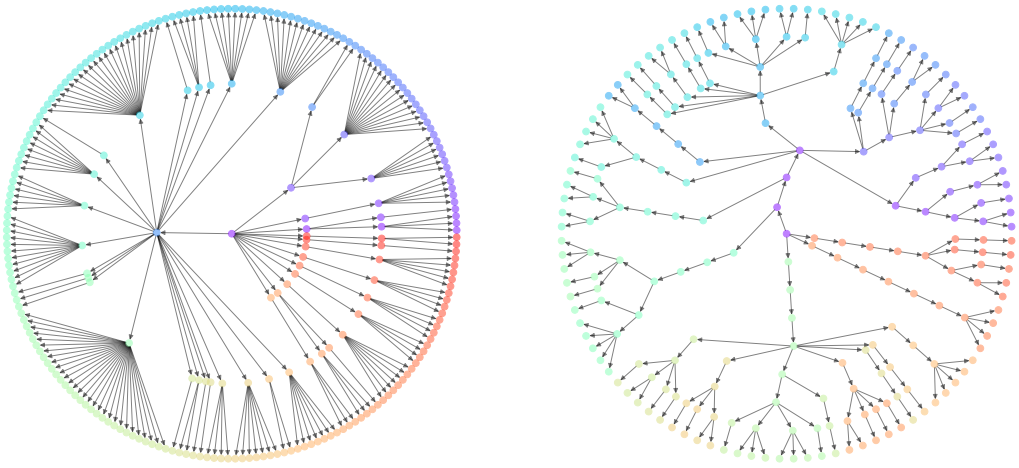
In Section 5.1, a new hyperbolic classification model based on horospheres was presented. The proposed horospherical classification leverages the hyperbolic geometry through the Busemann function but does not explicitly leverage the ability of hyperbolic spaces to embed trees with minimal distortion. However, this property can be useful in order

to perform classification on hierarchical datasets. That is, datasets where labels are organised in a hierarchy. As an example, the ImageNet dataset (Russakovsky et al., 2015) labels are organised following the WordNet (Miller, 1998) hierarchy. This means that a label can be more related semantically to one label than to another depending on the distance between the two labels in the label hierarchy. One example of such a hierarchy for the CUB200 dataset (Wah et al., 2011) can be seen in Figure 5.3a where the lower level of the hierarchy (*i.e.* leaves) corresponds to bird species, the level above to biological families and the first level to biological orders. The notion of distance between labels can be defined in several ways, for example, Mettes et al. (2019) and Liu et al. (2020) use the distance between textual word embeddings of labels. When a label hierarchy is available, the distance between two labels can be defined as the shortest path in the tree induced by the hierarchy between the two labels. As such, in CUB200, the two species *Pine Warbler* (see Subfigure 5.2a) and *Prairie Warbler* (see Subfigure 5.2b) have a distance of 2 because they are member of the same biological family, *Parulidae*. On the other hand, *Prairie Warbler* and *Sooty Albatross* (see Subfigure 5.2c) do not share the same biological order so their distance is 6, the maximum in the CUB200 hierarchy. As can be seen in Figure 5.2, in this case, biological families directly ties to visual similarity because birds from the same family exhibit similar visual features but also are more likely to appear in a similar environment. Therefore, we suppose that making prototypes of the same family close to each other in latent space can lead to better performance as the decision boundary between leaf labels will only focus on the intrinsic differences within that superclass. Said differently, it will classify based on what really differentiates a *Pine Warbler* from a *Prairie Warbler*, knowing they both are members from the *Parulidae* biological family.

In this subsection, we introduce a hierarchically informed scheme to initialize ideal prototypes which are the starting point for horospherical classification. This enables prototypes to be already hierarchically positioned upon the start of training which can help the model reduce the gravity of its classification errors. Indeed, if semantically related prototypes are located close to each other, then classification errors will fall in semantically related classes instead of completely unrelated ones, which should ideally improve the hierarchical performance of the models as well as its baseline performance.

5.2.1 Uniform ideal prototypes.

In order to fully leverage the embedding space, we initialize the layer with prototypes uniformly distributed around the hypersphere $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d, \|x\|_2 = 1\}$. While uniformly



(a) CUB200 Hierarchy. (b) CIFAR100 Hierarchy.

Figure 5.3: The CUB200 (Wah et al., 2011) and CIFAR100 (Krizhevsky & Hinton, 2009) label hierarchies. The CUB200 hierarchy contains three levels, representing the biological order, family and species of labels whereas the CIFAR100 hierarchy contains 100 classes with 9 levels.

distributing around the circle can be done in a closed-form by simply dividing the circle in K slices such that $\mathrm{p}_i = [\cos(2i\pi/K), \sin(2i\pi/K)]^\top$, $\forall i \in [1, K]$. However, there exists no such solutions for higher dimensions. Computing uniform positions for a set of points on the hypersphere can be done in different manners (Bonet et al., 2023b; Wang & Isola, 2020) as was also shown in Section 3.2. For this section, we choose to optimize a set of K points $\left\{\mathrm{p}_i \in \mathbb{S}^{d-1}; i = 1, \ldots, K\right\}$ on the sphere using the uniform loss proposed by Wang and Isola (2020). This uniform loss is based on the pairwise Gaussian potentials between each point. It effectively maximises the pairwise distance between each pair $\mathrm{p}_i, \mathrm{p}_j$:

$$\mathcal{L}_{\text{unif}}(\mathrm{p}) = \log \left(\frac{1}{K(K-1)} \sum_{i \neq j} \exp(-\|\mathrm{p}_i - \mathrm{p}_j\|_2^2) \right). \tag{5.5}$$

After optimization, this set of points can be used as ideal prototypes in the Poincaré ball model. Whereas the Busemann classifier (Ghadimi Atigh et al., 2021) randomly assigns prototypes to class labels, we now consider the problem of assigning a class label to each prototype in a hierarchically informed manner to improve the performance of the classification model.

5.2.2 Hierarchical ideal prototypes

On one side, we have a set of ideal prototypes uniformly distributed around the hypersphere $\mathbb{S}^{d-1}$. On the other side, we have a label hierarchy represented as a tree. As such, there is no straight forward way to compute a distance between an ideal prototype and a label since they are embedded on different support. Garnot and Landrieu (2021) work around a similar problem by providing a scale-free variant of their distortion loss.

Instead, in this context of incomparable spaces, one can resort to the Gromov-Wasserstein distance (Mémoli, 2011). The GW distance is used to compute a transport plan P between samples from two empirical distributions whose supports are embedded in incomparable spaces (see Subsection 2.3.4 for more background).

The tree metric is taken to be the length of the path between two nodes in the hierarchy. Therefore, two leaves i and j with the same parent will have a distance $(M_{\mathbb{T}})_{i,j} = 2$. For the sphere metric, we use the cosine distance $(M_{\mathbb{S}})_{i,j} = 1 - \langle \mathrm{p}_i, \mathrm{p}_j \rangle$ which is an increasing function of the length of the arc between p_i and p_j and $0 \leq (M_{\mathbb{S}})_{i,j} \leq 2$.

Given $M_{\mathbb{T}}$ and $M_{\mathbb{S}}$, the distance metrics on the tree and on the sphere respectively, one can compute the optimal transport plan between atoms from $\mathbb{T}$ and $\mathbb{S}$:

$$P^\star = \underset{P \in U(a,b)}{\operatorname{argmin}} \sum_{i,j,i',j'} |(M_{\mathbb{T}})_{i,i'} - (M_{\mathbb{S}})_{j,j'}|^2 P_{i,j} P_{i',j'} \tag{5.6}$$

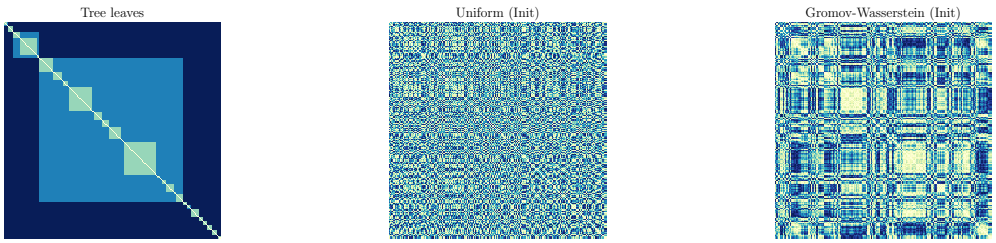
where $a_i = b_i = 1/K$ are uniform marginal distributions for both labels in the hierarchy and prototypes on the sphere respectively.

Since both a and b have the same cardinality, the optimal transport plan $P^\star$ is a permutation matrix which can be use to assign prototypes to nodes in the hierarchy. Therefore, label i will be assigned to prototype $j^\star$ as such:

$$j^\star = \underset{j \in \{1..K\}}{\operatorname{argmax}} P^\star_{i,j}. \tag{5.7}$$

The different metrics displayed in a visual form can be seen in Figure 5.4. In this case, we can see that the sphere metric after applying our proposed assignment strategy exhibits a diagonal structure similar to that of the tree metric.

In order to visualize this assignment more explicitly, we display the uniform prototypes before the assignment using a hierarchical color map in subfigure 5.5a. After the hierarchical assignment, the same display can be used. In Figure 5.5, one can notice that after the hierarchical positioning step, prototypes of similar colors and therefore of similar



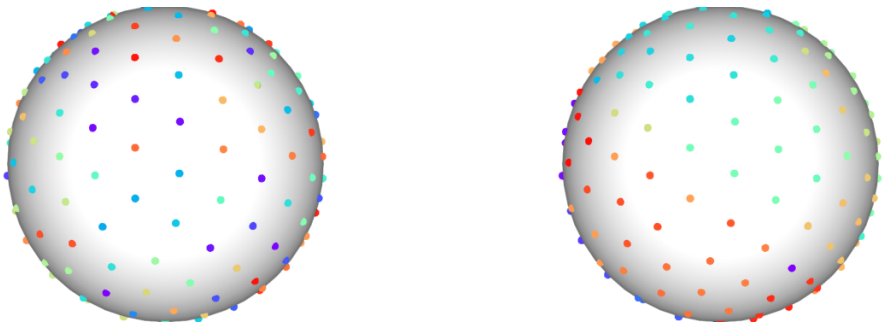
(a) Tree metric $M_{\mathbb{T}}$ as the longest path between labels in the hierarchy.

(b) Sphere metric $M_{\mathbb{S}}$ metric of randomly assigned prototypes.

(c) Sphere metric $M_{\mathbb{S}}$ metric of prototypes assigned using our proposed method.

Figure 5.4: Comparison of the different distance matrices, axes are sorted such that leaves closer in tree also appear closer in the axis. Hence the diagonal pattern in the tree metric which can be seen in Subfigure 5.4a.

hierarchical similarities are close together on the sphere.



(a) Uniform prototypes.

(b) Gromov-Wasserstein prototypes.

Figure 5.5: Example of the prototypes before and after our proposed assignment strategy on the sphere $\mathbb{S}^2$.

Additionally, we consider the simpler case of positioning hierarchical prototypes on the boundary of the Poincaré disk in two dimensions (*i.e.* the unit circle $\mathbb{S}^1$). Firstly, uniformly distributing points on the circle can be done in a closed form by slicing the circle in arcs of equal length $2\pi/K$ as explained in Section 5.2.1. The hierarchical structure can also be embedded on the circle without having to compute an optimal transport plan, by using the graph visualization algorithm used to display the hierarchies in Figure 5.3 called *twopi* introduced by Wills (1999). The algorithm evenly divides the circle in K slices and positions each prototype by assigning slices to nodes with an angle proportional to their

number of successors. We rely on this method in two dimensions.

5.3 Experiments

As we have seen in Section 5.1, our proposed horospherical classifier can perform multi-class classification on samples embedded in Hyperbolic space. In order to evaluate its classification performance against other hyperbolic and Euclidean baselines, we run classification benchmarks on several tasks. We first perform image level classification over hierarchical datasets. And secondly, we run benchmarks on semantic segmentation tasks on images as well as point clouds where label hierarchies are derived from the semantic meaning of labels.

5.3.1 Linking Visual and Hierarchical Similarities

To test this hypothesis, we perform a simple experiment. Given a perceptual image similarity measure $s(\cdot, \cdot)$ which takes an image and returns a similarity score based on their visual resemblance. Our experiment consists in comparing the mean similarity between images which have the same predecessors in the hierarchy compared to mean similarities between images which have different predecessors. If the hierarchy indeed has a relation with the visual similitude then we expect to see higher similarities in images which have a more common ancestor. Said differently, there should be a correlation between the length of the longest path between two classes on the hierarchy and the average distances between their respective images.

First, we define a visual distance metric between images based on a ResNet18 (He et al., 2016) model pretrained on the ImageNet 1k dataset (Krizhevsky et al., 2017) and whose last linear classification layer has been removed to produce latent representations for each input image. Indeed, pre-trained networks show interesting properties for measuring visual similarities between images (Zhang et al., 2018) and are in practice more robust than hand-crafted perceptual metrics. Since the ResNet is not trained in a hierarchically-aware fashion we consider that it can be used as a reliable image encoder. We then measure the Euclidean distance between these latent representations of two images as a visual distance measure. We refer to the encoder model as $r_{18} : \mathcal{I} \to \mathbb{R}^{512}$ where $\mathcal{I}$ refers to the space of images corresponding to that of the dataset. Then, the distance can be defined as:

$$d(x, y) = \|r_{18}(x) - r_{18}(y)\|. \tag{5.8}$$

Now, given a leaf class i of a dataset $\Omega = \mathcal{X} \times \mathcal{Y}$ of size N, we refer to X_i as the set of samples with class label i such that $X_i = \{x_k; y_k = i, k \in [1, N]\}$ with x_k being the k-th sample of the dataset with ground-truth label y_k.

$$D_{\text{Intra-Class}}(i) = \frac{1}{|X_i|^2 - |X_i|} \sum_{x \in X_i} \sum_{y \in X_i} 1_{x \neq y} d(x, y). \tag{5.9}$$

Then, we introduce the function $\mathrm{p}_h(i)$ which returns the ancestor of leaf class i at height h. Therefore, for a leaf class at height h then $p_h(i) = i$. Following this definition, we compute the mean distance with samples from other classes which have a similar ancestor at height h that we refer to as $D^h_{\text{Intra-Class}}$:

$$D^h_{\text{Extra-Class}}(i) = \frac{1}{|X_i| \times (\sum_{p_h(j)=p_h(i); i \neq j} |X_j|)} \sum_{x \in X_i} \sum_{j; p_h(j)=p_h(i); i \neq j} \sum_{y \in I_j} d(x, y). \tag{5.10}$$

For each leaf class in the hierarchy, we measure multiple mean distances with increasingly hierarchically-distant samples. First, we compute the mean intra-class similarity by taking the mean of all pairwise distances between samples of the samples of the same class. Then, we measure the mean distance between samples from a class and all samples from the same family which we refer to *Intra-Family*. Next, we compute the mean distance with all samples but from the same biological order *Intra-Order*. Finally, we compute the mean distance with samples outside the biological order, which we refer to as *Extra-Order*. As more hierarchically-distant samples are included, the mean distance between samples increases as can be seen in Figure 5.6. This is a promising result to confirm the benefit of leveraging hierarchical information in order to perform better visual classification.

5.3.2 Image Classification

In order to test the performance in image-level classification, we select image datasets with different heights of label hierarchies. We compare our proposed methods with the Euclidean baseline using a regular linear layer to perform the final classification. We also perform experiments on the prototype learning method from Garnot and Landrieu (2021) which learns Euclidean prototypes using a training loss which minimize distortion

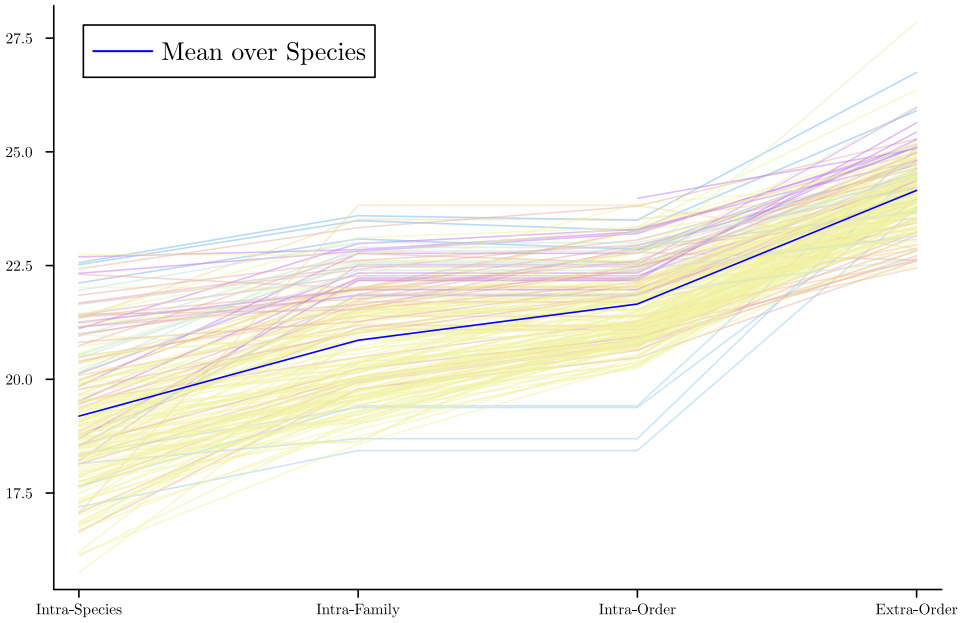


Figure 5.6: Evolution of the different means of distances either from samples from the same species (Intra-Species), same family (Intra-Family), same order (Intra-Order) or different order (Extra-Order) on the CUB200 dataset. As the level of granularity becomes higher in the hierarchy, the mean distance between images increases. Series are coloured based on the biological order.

between the label hierarchy and the prototypes' positions. As an hyperbolic classification baseline, we select the hyperbolic neural networks (Ganea et al., 2018b) which remain one of the most popular method for classification in hyperbolic spaces. Finally, we also compare with the Busemann learning approach introduced by Ghadimi Atigh et al. (2021) in which the prototypes are fixed but the Busemann function is also used to measure a distance between prototypes and hyperbolic representations.

We first select the CIFAR10 and CIFAR100 (Krizhevsky & Hinton, 2009) datasets which contains 10 and 100 classes of common visual objects, respectively. For the CIFAR10 dataset, we divide the labels in two simple super-classes, *vehicle* and *animal*, giving a hierarchy of depth 2 (see Figure A.3). For the CIFAR100, we use an available hierarchy which has 9 levels in its deepest branch (see Figure 5.3b). Branches do not all have the same length but for ease of use we insert inert nodes where needed to extend branches which are too short to the desired length of the longest branch (*i.e.* the height of the tree).

We also choose to experiment on the CUB-200 (Wah et al., 2011) which contains 200

bird species. It is composed of 5994 training images and a test set of 5794 images. Each species belongs to a single family which in turn belongs to an order. This gives us a hierarchy of height 3 with 252 nodes and 200 leaves, which can be seen in Figure 5.3a.

As the backbone, we use a ResNet32 (He et al., 2016). Models are trained for 1110 epochs for CIFAR datasets and 2110 for the CUB200 dataset, as done in (Ghadimi Atigh et al., 2021). The Riemannian variant (Bécigneul & Ganea, 2018) of the Adam optimizer (Kingma & Ba, 2014) implemented in the Geoopt toolbox (Kochurov et al., 2020) is used for optimizing the parameters for the Horospherical method. For other methods without trainable parameters on manifolds, we use the Adam optimizer (Kingma & Ba, 2014). The curvature is set to $c = 1.0$ for all hyperbolic methods. The Gromov-Wasserstein transport plan for the hierarchical positioning is computed using the Python Optimal Transport toolbox (Flamary et al., 2021). All experiments are done with 3 runs.

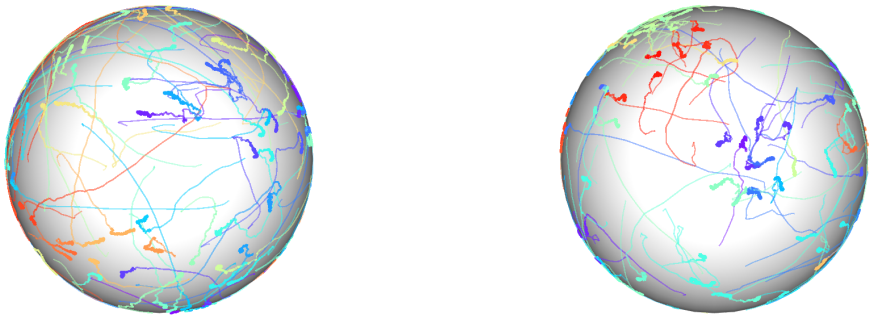
During training on the CUB200 dataset and in 3 dimensions, we save the position of the prototypes on the sphere. This allows us to display the trajectories of prototypes during training in Figure 5.7. In the figure, we can see that the prototypes initialized randomly move all around the sphere while hierarchically positioned prototypes are already clustered with prototypes of similar color and therefore of similar parent class. Even though the method only affects initialization and applies no training regularization with respect to the hierarchy, it seems that the clustering remains compliant with the hierarchy during training. This is in line with the hypothesis that the biological hierarchy of the CUB200 dataset can be of use to solve visual classification tasks.

The results can be seen in Table 5.1. We can see that the performance is improved with hierarchically positioned prototypes and horospherical classifier, especially when the embedding dimension is small such as with $d \leq 4$. On its own, the performance is improved for the Busemann method when initialized with hierarchically-positioned prototypes compared to using randomly assigned ideal prototypes distributed uniformly around the hypersphere.

Dataset	CIFAR10 (Krizhevsky & Hinton, 2009)				CIFAR100 (Krizhevsky & Hinton, 2009)				CUB200 (Wah et al., 2011)			
Dimension	2	3	4	50	2	3	4	50	2	3	4	50
Euclidean	89.8±0.2	90.6±0.1	90.8±0.2	**91.0±0.1**	45.9±0.5	53.3±0.8	56.2±0.2	62.5±0.3	24.3±1.3	46.8±0.7	50.5±1.5	51.8±0.8
Hyperbolic Neural Networks (Ganea et al., 2018b)	86.7±5.4	90.2±0.3	90.4±0.2	89.3±0.6	38.7±0.7	43.9±3.8	44.5±0.2	55.9±0.5	24.4±6.0	22.3±12.0	16.9±1.5	33.6±2.1
Metric Guided Prototypes (Garnot & Landrieu, 2021)	89.8±0.3	90.4±0.2	90.6±0.3	91.0±0.2	54.2±0.8	57.9±0.6	58.9±0.4	62.4±0.4	31.0±2.8	48.8±1.4	53.2±1.9	**57.3±0.7**
Busemann - Uniform (Ghadimi Atigh et al., 2021)	89.7±0.3	90.6±0.2	90.7±0.2	88.6±0.1	49.7±0.7	48.7±2.3	50.2±1.5	63.3±0.6	27.6±7.4	40.0±2.0	45.0±1.7	49.5±3.7
Busemann - Smart (Ours)	89.8±0.1	90.4±0.5	90.7±0.1	88.6±0.1	**55.7±0.5**	54.8±0.4	55.4±0.3	**63.3±0.3**	**53.3±1.6**	54.3±0.4	**58.0±1.1**	50.4±0.6
Horospherical - Smart (Ours)	**90.1±0.3**	**90.7±0.2**	**90.8±0.1**	90.8±0.2	53.7±0.8	**58.7±0.3**	**60.0±0.1**	60.9±0.1	38.7±4.3	**56.4±1.6**	56.3±0.8	56.6±0.6

Table 5.1: Results on image classification. Best results are displayed in **bold** while the second best are underlined.

In order to evaluate the classification performance, we measure the Average Hierarchi-



(a) Prototypes trajectories when initialized randomly.

(b) Prototypes trajectories with hierarchically-informed initialization.

Figure 5.7: Comparison of the trajectories between random (a) and hierarchically-informed (b) prototype positioning.

cal Cost (AHC) (Kosmopoulos et al., 2015) of the different methods on the datasets. The AHC is a measure of performance used to quantify the hierarchical costs of errors. For a hierarchy of labels with C leaf labels, we build a symmetric distance matrix $D \in \mathbb{R}^{C \times C}$, where $D_{i,j}$ corresponds to the shortest path in the hierarchy between labels i and j. Given predictions $z_i \in \{1, \ldots, C\}$ from a model of N test samples and ground $y_i \in \{1, \ldots, C\}$ from the dataset with $i \in \{1, \ldots, N\}$, The Average Hierarchical Cost is defined as:

$$AHC(z, y) = \frac{1}{N} \sum_{i=1}^{N} D_{y_i, z_i}. \tag{5.11}$$

Note that unlike accuracy, the lowest the AHC is, the better the hierarchical classification performance. An AHC of 0 corresponds to an actual Top 1 accuracy of 100%. The Average Hierarchical Cost of baseline methods and our proposed horospherical classifier with hierarchical initialization can be seen in Table 5.2. In the results, we notice that the hierarchically-initialized horospherical classifier has a lower AHC than other methods but is on par with the Busemann classifier (Ghadimi Atigh et al., 2021) in lower dimensions ($d = 4$) while also being on par with metric guided prototypes (Garnot & Landrieu, 2021) in higher dimensions ($d = 50$). Interestingly, the AHC increases for all methods when increasing the embeddings' dimensions.

Method	AHC↓	
	$d = 4$	$d = 50$
Euclidean	2.12	2.18
Metric-Guided (Garnot & Landrieu, 2021)	1.72	**1.80**
Busemann (Ghadimi Atigh et al., 2021)	**1.63**	2.07
Horospherical (Ours)	**1.63**	**1.80**

Table 5.2: Average Hierarchical Cost of different methods on the CUB200 dataset.

5.3.3 Semantic Segmentation

Since our horospherical classifier can be used in place of a regular Euclidean linear layer, we choose to measure its performance for dense tasks such as image semantic segmentation and point cloud semantic segmentation. In the context of such dense tasks, label hierarchies can be derived from either the semantics or by grouping together labels which are finer annotations of a bigger object. For example, in an image segmentation task, the individual body parts could be the leaf nodes whose parent is the super class body.

Similarly to image classification, we compare our method with the baseline Euclidean classification. As an hyperbolic baseline, we choose to evaluate the performance of the Hyperbolic Image Segmentation from Atigh et al. (2022).

For image segmentation, we consider the Cityscapes dataset (Cordts et al., 2016) which contains 19 classes divided in 7 super-classes (*flat, construction, object, nature, sky, human*, and *vehicle*) as can be on the hierarchy presented in Figure A.5. We use a DeepLab-v3+ (Chen et al., 2018) backbone trained with the official split for 240 epochs. We set $\lambda = 0.5$. For all hyperbolic methods, c is set to 1.0. Experiments are done with 3 runs.

For point cloud segmentation, we perform experiments on the NuScenes (Caesar et al., 2020) dataset which contains 40k frames, sampled from 1000 driving sequences in Boston and Singapore, with a rotating LiDAR. Points are classified into 16 semantic classes and 1 ignore class. We use the official train split for training, and report the results obtained on the publicly available validation set. For the class hierarchy we leverage the NuScenes class description and split into 5 branches (*movable object, vehicle, pedestrian, flat* and *static*). We additionally split the vehicle branch into 4-wheeled and 2-wheeled. This leads to a hierarchy with 3 layers which can be seen in Figure A.4. As a point cloud processing backbone, we use the commonly used sparse-voxel Minkowski U-Net (Choy et al., 2019), with a 10 cm voxel size. We set $\lambda = 0.1$.

Dataset	NuScenes (Caesar et al., 2020)			Cityscapes (Cordts et al., 2016)		
Dimension	$d = 2$	$d = 3$	$d = 128$	$d = 2$	$d = 3$	$d = 128$
Euclidean	54.0±2.4	67.4±0.3	**70.5±0.5**	35.9±0.0	60.9±4.6	**78.8±0.4**
HIS (Ghadimi Atigh et al., 2021)	40.2±5.4	58.0±2.7	69.5±0.2	41.3±4.8	45.1±7.7	77.9±0.2
Horospherical - Smart (Ours)	**68.4±0.1**	**68.7±0.0**	69.2±0.3	**73.5±0.4**	**76.1±0.1**	78.2±0.4

Table 5.3: Results on semantic segmentation (%mIoU). Best results are displayed in **bold** while the second best are underlined.

Semantic segmentation results can be seen in Table 5.3. Similarly to the results in image classification, our proposed horospherical model with hierarchical initialization performs well, especially in lower dimensions. Finally, as seen previously in Atigh et al. (2022), when applied to higher dimensions, the gap between Euclidean and hyperbolic representations shrinks and Euclidean classification exhibits better performance. Nevertheless, these results are promising for the future of hyperbolic classification in dense segmentation tasks.

5.4 Conclusion and Discussion

In this work, we contribute to the task of classification over hierarchical dataset. Our solution is built upon the ability of hyperbolic spaces to embed tree-like structure with minimal distortion. The proposed hyperbolic classifier can be used in place of Euclidean classifiers in order to solve tasks such as image classification and semantic segmentation.

A future research direction is investigating improvements to our framework by adding additional hierarchical regularization during the model training similar in vain to the distortion measure proposed by Garnot and Landrieu (2021). Indeed, in its current state, our method only leverages the hierarchical prior in its initialization phase and has no hierarchical knowledge during its training as the hyperbolic classification is only performed over leaf nodes.

Chapter 6

Conclusion

Contents

6.1 Overview of Contributions 104

6.2 Perspectives 106

6.1 Overview of Contributions

In this thesis, we considered the process of representation learning for images. To this end, we have explored both computer vision and remote sensing applications which each require bespoke methods. Moreover, our work also carefully considers the specificities of each dataset. As such, our proposed methods are often tailored to specific properties that an image dataset can possess. This contributions are aligned with our initial objectives of looking at self-supervised learning to reduce the need for large amounts of unlabelled data as well as to propose methods more considerate about the possible hierarchical or multi-modal nature of images in a dataset.

In Chapter 3, we have explored the use of Optimal Transport for self-supervised representation learning. This methodological work has pushed us to introduce a new formulation of the self-supervised joint-embedding representation learning objective using OT plans between features or samples. We have also researched in Section 3.2, the ability of the spherical sliced Wasserstein to enforce hyperspherical uniformity in a self-supervised training objective with a lower computational complexity. Notably, the existence of a closed-form when computing the Spherical Sliced Wasserstein (Bonet et al., 2023b) has been the basis of our work. Next up in Section 3.3, we have looked into the use of OT in dense self-supervised learning in order to improve the downstream performance of self-supervised models in dense tasks such as object detection or semantic segmentation.

Next, in Chapter 4, we have focused on the more applied problem of learning efficient representations for remote sensing multi-modal datasets which contain co-registered im-

ages. This leads us to first evaluate the state of self-supervised representation learning in remote sensing. Next, we have analyzed the setting of joint-contrastive learning for multi-modal data by considering each modality as an augmented view in the traditional contrastive learning setting in Section 4.2. With our promising results in multi-modal self-supervised representation learning, we have generalized this multi-modal contrastive objective to the supervised setting by proposing a supervised learning framework largely inspired by the Supervised Contrastive (SupCon) method (Khosla et al., 2020).

Finally in Chapter 5, we have studied the problem of hierarchical classification and explored how hyperbolic spaces can be used in such cases to improve over Euclidean baselines due to their natural ability to embed trees with minimal distortion. This work leads us to introduce a horospherical classifier based on horospheres which are level-sets of the Busemann function. Compared to the Busemann classifier introduced by Ghadimi Atigh et al. (2021) which has been a source of inspiration for our work, we have introduced a hierarchically-informed manner to initialize these prototypes on the hypersphere. This initialization scheme improves the hierarchical classification performance compared to randomly distributing prototypes around the hypersphere.

While at the beginning of the thesis, the number of newly published methods for self-supervised learning was frankly a bit overwhelming, it seems that the computer vision community is broadly converging to the same performant methods (Oquab et al., 2023; Zbontar et al., 2021) and scaling those to propose self-supervised models whose weights are openly available. In this context and with only working with relatively small models, our contributions may not seem too aligned with this trend. However, this ever scaling movement is not accessible to lesser equipped computer labs. Though we are not to be pitied, thanks to the compute grants provided by IDRIS on the Jean-Zay supercomputer. As such, we choose to work on proposing methods which could yield improvements on smaller models. In our experiments, there is often a trade-off between running many compute heavy experiments in order to tune hyper-parameters and gaining little insight into the response of the methods with respect to different parameters. As such, we try to launch experiment runs to gain a better understanding of our methods but still remaining aware of the potential computational, energetic and environmental costs.

As hinted by the sudden popularity of self-supervised methods, the well-known ImageNet initialization is not the preferred method for initialising model's weights. Instead, reaching for open models pre-trained in a self-supervised fashion is now becoming the most popular initialization as those model oftentimes have an easier time generalizing to

different tasks than what they were initially trained on.

6.2 Perspectives

Our work presented in this thesis opens up a series of perspective future works which we describe in this section. Finally, we also try to position our work within the more global and recent trends in representation learning for images, computer vision and remote sensing at large.

Optimal Transport for Self-Supervised Learning. While we only reached mixed results in our research in integrating Optimal Transport and Self-Supervised Learning during chapter 3, OT remains a promising tool in representation learning because many parts of the problem can be coined as distances between distributions where one can conveniently reach for Optimal Transport and its variants. We note that other successful approaches of OT in self-supervised representation learning often leverage the entropic OT problem via the Sinkhorn-Knopp algorithm as demonstrated in Caron et al. (2020) or Oquab et al. (2023) in part due to its numerical scaling abilities compared to other variants.

Multi-Modal Learning. The setting we studied in Chapter 4 concerns the classification of multi-modal datasets whose samples are co-registered images which have been captured by different sensors. A particularly challenging problem to tackle would be to consider the setting where not all samples have captures in all modalities, resulting in a dataset harder to align across modalities. In this case, the samples which contain all modalities could be referred to as anchors. The goal would be to leverage such anchors in order to reach better performance than when performing the training on either modalities.

Horospherical Learning. Hyperbolic spaces have already shown that they can provide interesting benefits compared to Euclidean spaces in relevant tasks. Following the promising results from our horospherical classifier, we would like to bring similar improvements to hierarchical classification to self-supervised learning. While hyperbolic spaces have already been used successfully in SSL (Ge et al., 2023), we believe that hyperbolic representations can still bring further improvements to the world of self-supervised representation learning. Moreover, our contribution to the task of hierarchical classification can be still be improved and leverage in other tasks where hierarchy plays a role such as few- or zero-shot learning. In this case, the association of having well-defined ideal prototypes for each class and a hierarchical positioning for them could be an asset when

unseen classes are introduced. We also note that our positioning scheme is currently only used during the layer initialization. As such, there is a possible improvement by regularizing the movement of prototypes during training according to the hierarchy in a manner inspired by the scale-free distortion loss from Garnot and Landrieu (2021). In this case, the cost of computing Gromov-Wasserstein remains marginal since it would only be a function of the number of labels in the dataset and not dependent on the number of samples in the dataset. Thus, the added computation time would be constant during the training and scale conveniently with a dataset's size.

On a more global level, it seems that leveraging existing large self-supervised models is becoming more popular. Indeed, a rising trend in representation learning is to release so-called *foundation models* which are trained on very large amounts of data. This scaling enables these models to perform well in a variety of downstream tasks. Such models have already been proposed in the remote sensing community (Gao et al., 2022; Jakubik et al., 2023). The multi-modal alignment of the latent space between different modalities make these models able to perform well with multiple modalities such as text and images (Radford et al., 2021) or multiple visual modalities (Mizrahi et al., 2024). As such, training self-supervised models from scratch can be less beneficial than being able to embed one's data in the latent space of an existing foundation model. Nevertheless, we hope that our works can stimulate further research in using OT for SSL, multi-modal representation learning or hyperbolic representation learning for hierarchical classification. Indeed, tasks on datasets with features such as a hierarchical organisation of labels or multi-modal nature still need to be fulfilled.

Appendix A

Appendix

Contents

A.1 List of Publications 108

A.2 Appendix of Chapter 4 109

A.3 Appendix of Chapter 5 109

A.1 List of Publications

A.1.1 Journal Articles

- Berg, P., Pham, M.-T., & Courty, N. (2022). Self-supervised learning for scene classification in remote sensing: Current state of the art and perspectives. *Remote Sensing*, *14*(16), 3995

- Berg, P., Uzun, B., Pham, M.-T., & Courty, N. (2024c). Multimodal supervised contrastive learning in remote sensing downstream tasks. *IEEE Geoscience and Remote Sensing Letters*

A.1.2 Conference Articles

- Bonet, C., Berg, P., Courty, N., Septier, F., Drumetz, L., & Pham, M.-T. (2023b). Spherical sliced-wasserstein. *International Conference on Learning Representations*

- Berg, P., Pham, M.-T., & Courty, N. (2023). Joint multi-modal self-supervised pre-training in remote sensing: Application to methane source classification. *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*, 6624–6627

- Berg, P., Michele, B., Pham, M.-T., Chapel, L., & Courty, N. (2024a). Horospherical learning with smart prototypes. *Proceedings of the British Machine Vision Conference (BMVC)*

- Lê, H.-A., Berg, P., & Pham, M.-T. (2024). Box for mask and mask for box: Weak losses for multi-task partially supervised learning. *Proceedings of the British Machine Vision Conference (BMVC)*

A.1.3 Communications without Proceedings

- Bonet, C., Berg, P., Courty, N., Septier, F., Drumetz, L., & Pham, M.-T. (2023a). Sliced-wasserstein spherique. In *Conférence sur l'apprentissage automatique (cap)*

- Berg, P., Pham, M.-T., & Courty, N. (2024b). Apprentissage contrastif multi-modal : Du pré-entrainement auto-supervisé à la classification supervisée. In *Reconnaissance des formes, image, apprentissage et perception (rfiap)*

A.2 Appendix of Chapter 4

Cohen's κ score. The κ score (Cohen, 1960) measures the agreement between two distributions of categorical items. It is often considered more robust than accuracy because it takes into account the probability of making the right prediction randomly. It is defined as such:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{A.1}$$

where p_o is probability of agreement between the two distributions, that is among N predictions over C classes $y_i \in \{1, \dots, C\}, z_i \in \{1, \dots, C\}$, $\forall i \in \{1, \dots, N\}$, $p_o = \frac{1}{N} \sum_{i=1}^{N} 1_{\{y_i = z_i\}}$. p_e, on the other hand, is defined as $p_e = \frac{1}{N^2} \sum_{i=1}^{N} (\sum_{j=1}^{N} 1_{\{y_j = i\}}) \times (\sum_{k=1}^{N} 1_{\{z_k = i\}})$. As for the accuracy, a higher κ score means a better performance.

A.3 Appendix of Chapter 5

Impact of $\phi(d)$ regularization. In order to evaluate the impact of the regularization introduced in Equation 6 we perform an experimental study of varying the value of the

$\phi(d)$ parameter. Since we use a constant regularization defined as $\phi(d) = \lambda \times d$, we vary the value of the λ parameter in the range $[0, 2]$.

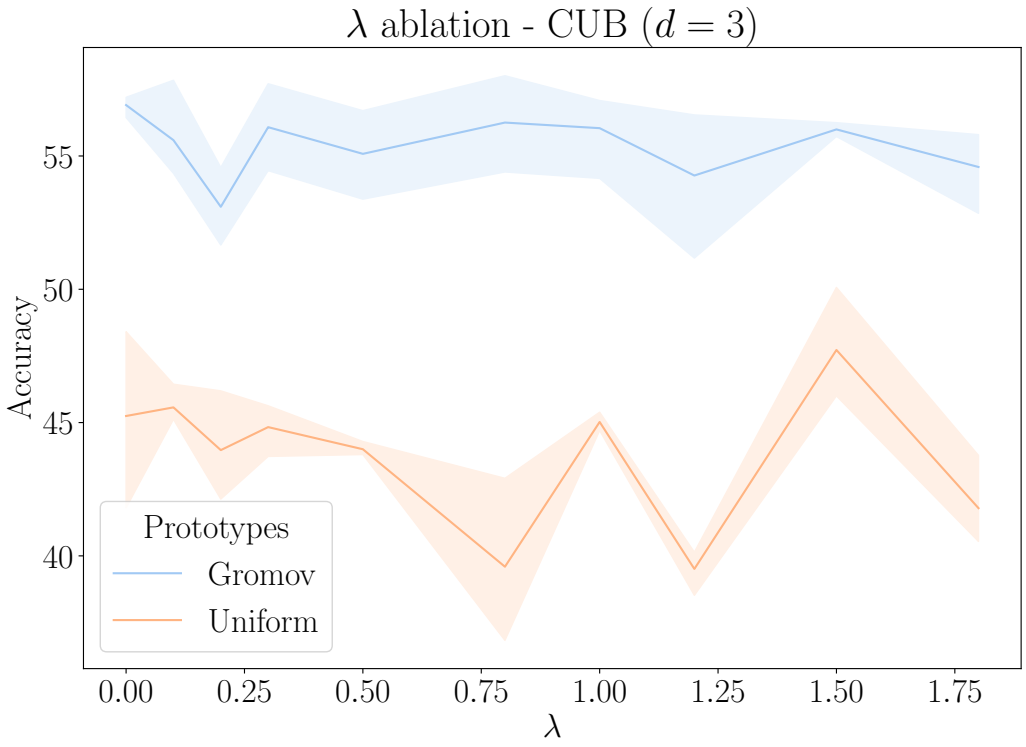


Figure A.1: Evolution of the accuracy when moving the λ regularisation parameter. We observe that varying the value of the parameter has little impact on the resulting performance.

Impact of the bias a. The bias a in Equation 6 acts on the radius of each horosphere. We experiment with disabling the bias parameter in order to evaluate its impact on the final performance of the model. As can be seen in Table A.1, the bias provides in 2 dimensions an increase in performance but seems to be of less importance for higher dimensions.

Radius of a Horosphere. Given a horosphere parameterized by an ideal prototype $\mathrm{p} \in \mathbb{S}^{d-1}$ and a bias $a \in \mathbb{R}$. We can compute the radius of the said horosphere. Remember that horospheres in the Poincaré ball model are hyperspheres tangent to the boundary of the ball. To compute the radius dependent on a, we will find the two points of the horosphere which are located on the Poincaré ball radius, one of this point is p, and we refer to the other one as x. By taking p as our first base vector, x has a single non-null

Table A.1: Performance when enabling or disabling biases during training of horospherical classifiers on the CUB dataset.

Method	Bias	Dimensions 2	3	4	50
Horospherical - Smart		34.4±6.2	56.4±0.8	56.0±0.4	**57.5±0.2**
Horospherical - Smart	✓	**38.7±4.3**	56.4±1.6	**56.3±0.8**	56.6±0.6

dimension x_0.

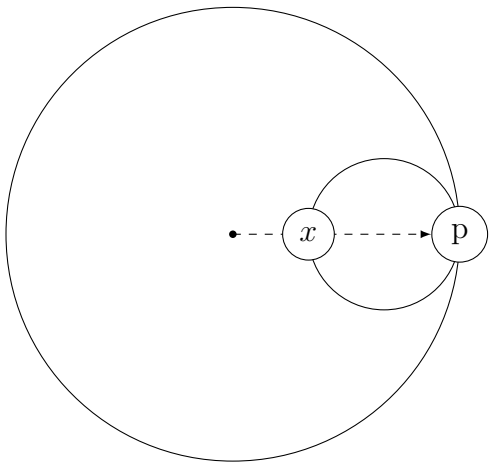


Figure A.2: The position of x along the vector p is x_0 such that $x_0 = \langle x, \mathrm{p} \rangle$.

$$-B_{\mathrm{p}}(x) + a = 0 \tag{A.2}$$

$$\log\left(\frac{\|\mathrm{p} - x\|^2}{1 - \|x\|^2}\right) - a = 0 \tag{A.3}$$

$$\|\mathrm{p} - x\|^2 = \frac{1 - \|x\|^2}{\exp(-a)} \tag{A.4}$$

$$(1 - x_0)^2 = \frac{1 - x_0^2}{\exp(-a)} \tag{A.5}$$

$$1 - 2x_0 + x_0^2 = \frac{1 - x_0^2}{\exp(-a)} \tag{A.6}$$

$$1 - 2x_0 + x_0^2 = (1 - x_0^2)\exp(a) \tag{A.7}$$

$$1 - \exp(a) - 2x_0 + (1 + \exp(a))x_0^2 = 0. \tag{A.8}$$

We find the roots for this polynomial using the quadratic formula:

$$x_0 = \frac{2 \pm \sqrt{4\exp(2a)}}{2(1+\exp(a))} \tag{A.9}$$

$$= \frac{2 \pm 2\sqrt{\exp(2a)}}{2(1+\exp(a))} \tag{A.10}$$

$$= \frac{2(1 \pm \exp(a))}{2(1+\exp(a))} \tag{A.11}$$

$$= \frac{1 \pm \exp(a)}{1+\exp(a)} \tag{A.12}$$

$$= \begin{cases} 1. & \text{if } x = \text{p.} \\ -\tanh(a/2) & \text{otherwise.} \end{cases} \tag{A.13}$$

Since we are interested in the solution other than the trivial $x = p$, the radius of a horosphere with a bias term a is:

$$r(a) = \frac{1+\tanh(a/2)}{2}. \tag{A.14}$$

Note that with a bias equal to 0, the diameter of the corresponding horosphere is equal to 1 which is consistent with the fact that the Busemann function is equal to 0 at the origin.

Label Hierarchies. Here, we include the hierarchies used for positioning prototypes in our experiments on different datasets. Other hierarchies can be seen in Figure 5.3a and Figure 5.3b.

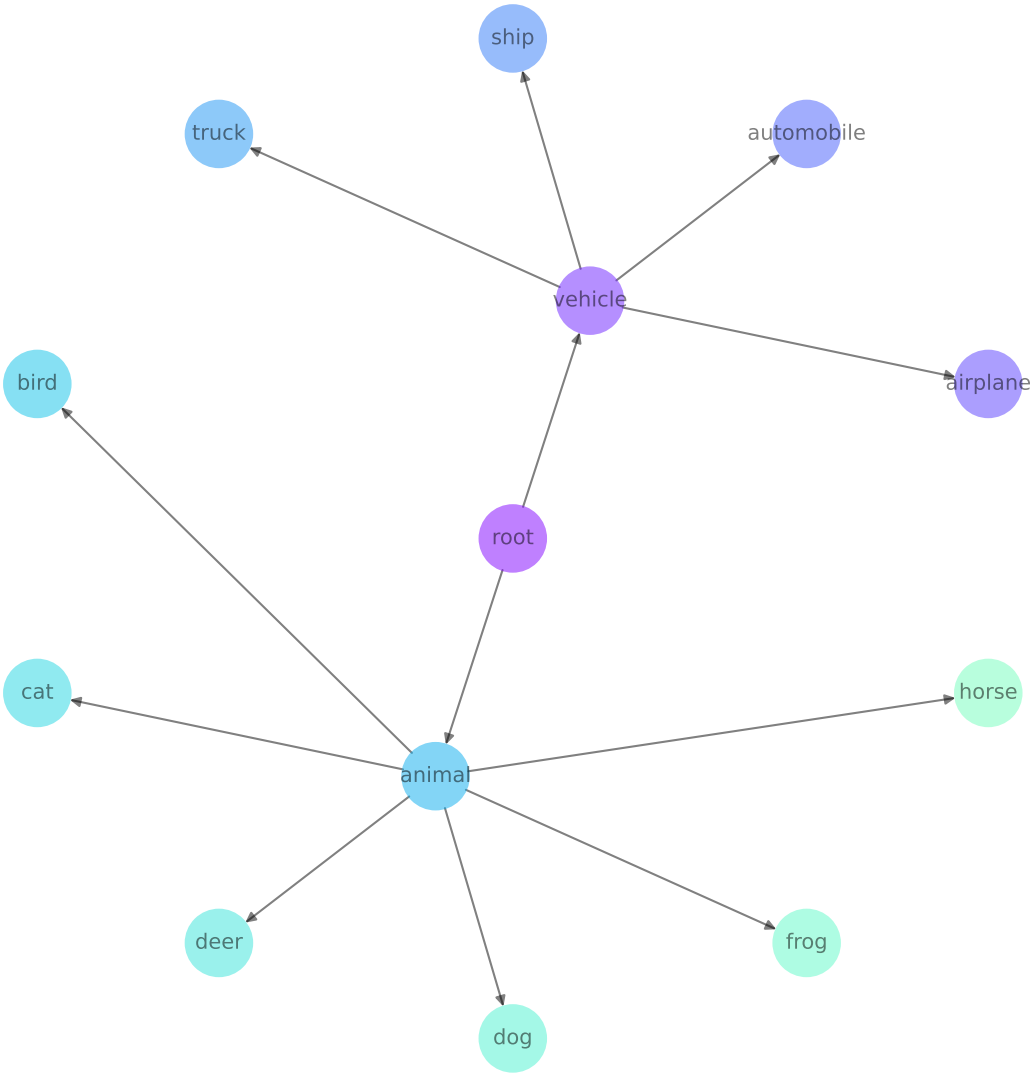


Figure A.3: CIFAR10 Hierarchy.

Figure A.4: NuScenes Hierarchy.

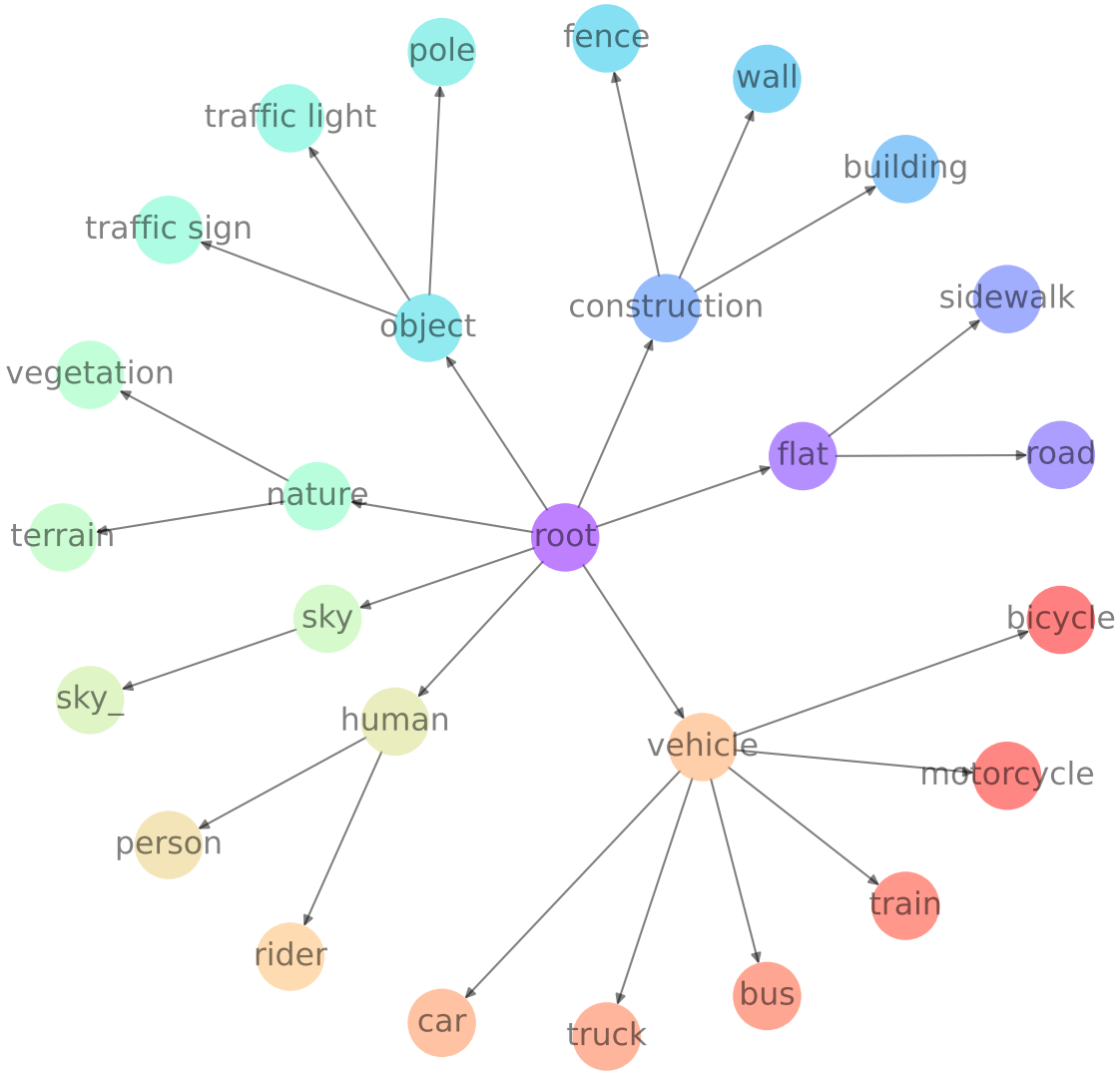


Figure A.5: Cityscapes Hierarchy.

Appendix B

Introduction (Français)

Contents

B.1 Motivations 116

B.2 Aperçu des contributions 122

B.1 Motivations

Tandis que la vision apparaît comme un processus naturel pour les humains, les ordinateurs n'ont pas de notion inhérente pour la compréhension d'images. Dans les ordinateurs, les images sont généralement représentées comme des matrices de pixels avec des dimensions supplémentaires si plusieurs canaux sont présents. Cette représentation peut être utile pour certaines analyses d'images but est largement motivée par la nature des capteurs d'images qui sont positionnés en forme de grille. C'est pourquoi les programmes informatiques n'ont pas de façon direct de raisonner à propos de la nature visuel des objets et éléments présent dans une image. Par conséquent, la communauté de recherche en vision par ordinateur a développée des techniques pour aider les ordinateurs à réaliser de meilleurs analyses et classification d'images en apprenant aux ordinateurs à apprendre de meilleurs représentations que la grille de pixels.

L'importance d'apprendre de meilleurs représentations en vision par ordinateur ne peut pas être sous-estimée. Les représentations traditionnelles conçues manuellement, qui était il fut un temps la pierre angulaire de la vision artificielle, ont largement été surpassées par les représentations apprises qui offre une plus grande flexibilité, adaptabilité et performance. La montée en popularité des techniques d'apprentissage profond a encore plus solidifié l'importance des représentations apprises, permettant aux modèles d'apprendre des paternes complexes et des hiérarchies de représentations à partir de gros jeux de données annotés.

De nos jours, les méthodes d'apprentissage profond sont devenus omniprésentes en

vision par ordinateur. Des réseaux profonds sont nourris avec plein d'exemples annotés afin de résoudre des tâches de compréhension d'images, atteignant parfois des performances comparables ou supérieures à celles des humains. Cependant, l'annotation de large volume de données reste une tâche complexe et méticuleuse. En fonction du type d'annotation, des experts peuvent être requis pour annoter les données pour la tâche visuel qu'ils veulent résoudre. Comme exemple, le jeu de données ImageNet (Russakovsky et al., 2015) a requis un effort considérable à annoter complètement ses plus de 14 millions d'images. Entraîner des modèles d'apprentissage profond sur de tels jeux de données est un processus couteux en ressources et en calcul mais les modèles résultants sont souvent non seulement capable d'être performant sur le jeu de données de choix mais ils servent également de bonne initialisation pour l'entrainement de modèles sur des jeux de données ayant des propriétés visuelles similaires. En effet, dans le cadre de sa tâche d'apprentissage, le modèle est capable d'extraire des représentations qui sont efficace pour décrire le contenu visuel d'une image, et en particulier dans les premières couches du modèle. De tel façon que ces poids peuvent être utilisés tels quel avec de bon résultats pour des tâches initialement sans rapport tel que la détection d'anomalies dans les images (Defard et al., 2021) ou pour mesurer une distance de perception entre images (Zhang et al., 2018).

Malheureusement, de tels jeux de données et poids pré entrainés ne sont pas toujours disponibles quand le jeu de données n'est pas composé d'images naturelles en couleur RGB ou que l'architecture du modèle est peu commune. Aussi, ces jeux de données peuvent contenir des biais qui les rendent non adéquat à l'usage hors du monde académique en fonction de comment ils ont été créés. Par exemple, dans les 1000 classes qui compose le jeu de données ImageNet, plus de 100 classes sont des races de chiens, ce qui pose la question de si les modèles pré entrainé sur celui-ci peuvent être utilisés tels quel sans adaptation pour d'autre tâches. Par conséquent, des méthodes d'apprentissage de représentations non basées sur l'apprentissage supervisé ont commencés à émerger pour remplacer la soit disant initialisation ImageNet. Un ensemble de méthodes de plus en plus populaire connues sous le nom de méthodes auto-supervisées a commencé à gagner de l'ampleur au sein de la communauté de recherche en vision par ordinateur. Elles sont basées sur l'idée qu'au lieu d'utiliser la classification comme tâche pour générer des représentations utiles, des tâches prétextes peuvent être conçues uniquement à partir d'un jeu de données non annoté. Par conséquent, un modèle peut être entrainé à résoudre de tel tâches prétextes sans dépendre d'une supervision extérieure pour apprendre des représentations sensées qui pourra ensuite être utilisé dans une série de tâches en aval avec plus de succès que l'initialisation par

poids supervisés. La rapide croissance en popularité de ces méthodes peut être visualisé sur la Figure B.1. Cette tendance pour l'auto supervision a commencé à accélérer en parallèle de la montée de l'apprentissage profond en vision par ordinateur avec notamment les ouvrages de références tels que ceux de Gidaris et al. (2018), Noroozi et Favaro (2016), Wu et al. (2018) et Zhang et al. (2016). Plus récemment, une branche populaire de l'apprentissage auto supervisé est d'utiliser des méthodes dites à représentations jointes ou un modèle est entraîné à produire des représentations similaires pour des vues augmentées artificiellement d'une image de référence. Ces méthodes inclut également une technique pour prévenir de l'effondrement des représentations dans lequel un modèle entraîné produit uniquement une représentation constante indépendante de l'image d'entrée mais qui a un alignement parfait entre les vues du même échantillon. Parmi ces méthodes à représentations jointes, nous pouvons distinguer les méthodes dites contrastives ou pour chaque échantillon, un ensemble d'échantillons négatifs est repoussés dans l'espace latent. Les méthodes non-contrastives, d'un autre côté, sont basées sur d'autres techniques pour prévenir cet effondrement tel que l'utilisation de l'opérateur coupe gradient sur une des branches (Chen & He, 2021 ; Grill et al., 2020) où sur la maximisation d'information (Bardes et al., 2022 ; Ermolov et al., 2021 ; Zbontar et al., 2021). Finalement, avec la popularité croissance des transformeurs pour la vision (Dosovitskiy et al., 2020), les auto-encodeurs masquées (He et al., 2021) offrent une direction de recherche prometteuse grâce à la simplicité de l'objectif de reconstruction.

En outre, dans des domaines spécifiques tel que celui de la télédétection, les jeux de données possèdent souvent des spécificités comparés aux jeux de données d'images naturelles. Les méthodes de d'apprentissage de représentations devrait pouvoir utiliser ces particularités afin de produire un processus d'entraînement plus efficace en données annotés. Par exemple, dans des taches denses telles que la segmentation sémantique, les objets et régions sont souvent petit en échelle comparé à l'image entière. Il y a également une large variété de résolutions spatiales en fonction des capteurs choisis. Cela peut résulter en des images de télédétection étant très larges et donc couteuse en calcul à analyser pour des méthodes sur étagère sans être pré-traiter dans un premier temps. La régularité à laquelle les satellites observent la terre est un autre facteur qui doit être pris en compte en commençant un projet d'observation de la terre. Il y a souvent un équilibre à trouver entre avoir des images à haute résolution spatiale mais avec une faible régularité par rapport à avoir des images à faible résolution spatiale mais prises avec une haute régularité. Certaines missions capturent uniquement les bandes RGB et proche de l'infrarouge

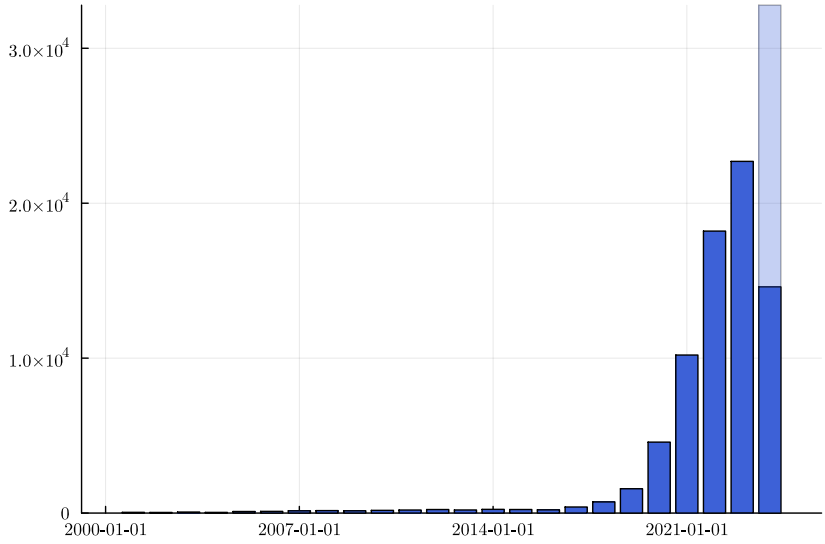


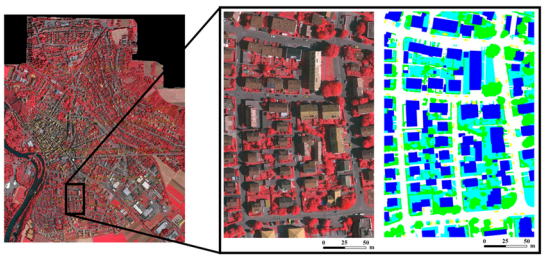
FIGURE B.1 : Nombre de publications mentionnant le terme "Self-Supervised Learning" publié chaque année depuis 2000 (Source : Google Scholar, le 11/06/2024). Les années récentes ont vu une augmentation dans le nombre de publications liées à l'apprentissage auto supervisé. Il faut noter que l'année 2024 n'est pas encore complétée et a donc été extrapolée avec les données disponibles.

tandis que d'autres capturent de larges bandes résultant dans des images hyperspectrales avec des centaines de canaux. C'est pourquoi les utilisateurs de données de télédétection sont souvent motiver de générer des jeux de données multimodaux afin d'accumuler les bénéfices de plusieurs fournisseurs d'images. Des exemples de plusieurs jeux de données de télédétection peuvent être vus sur la Figure B.2.

Dans un jeu de données multimodale, un échantillon est composé de plusieurs captures de différents capteurs. Ces capteurs peuvent varier largement et offrir des jeux de données complètement hétérogène qui requiert des méthodes spécifiques car les méthodes de vision par ordinateur ne sont pas applicables directement. Dans les jeux de données multimodaux, les échantillons peuvent être co-registré. C'est à dire, la capture de différents capteurs a été alignée en utilisant les coordonnées géographiques disponibles avec chaque échantillon. En effet, les images issues de la télédétection ont des caractéristiques différentes des images naturelles.

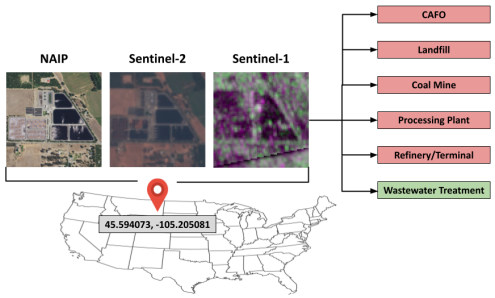
Un autre caractère que les jeux de données peuvent posséder est celui d'être hiérarchique. Un jeu de données hiérarchique est composé de labels qui peuvent être organisés en hiérarchie. Les taxonomies des jeux de données peuvent être exploitées pour produire des représentations plus informées et pour aider dans le processus d'entraînement. Dans

(a) Le jeu de détection de véhicules VEDAI (Razakarivony & Jurie, 2016).



(b) Le jeu de données de segmentation sémantique Vaihingen (Rottensteiner et al., 2012).

(c) Le jeu de données de segmentation de séries temporelles d'images satellites EuroCrops (Schneider et al., 2021) with hierarchical labels.



(d) Le jeu de données multi-modal de classification de scènes Meter-ML (Zhu et al., 2022).

Figure B.2 : Examples of remote sensing datasets created for diverse tasks. Images are taken from the respective papers/dataset.

les taches tels que la segmentation sémantique, l'organisation hiérarchique des labels vient naturellement. En télédétection, les labels peuvent souvent être groupés en différentes super classes sémantiques qui peut amener à la création d'une hiérarchie utile. La similarité dans de telles espaces hiérarchique pour deux échantillons devrait idéalement produire une plus grande similarité entre caractéristiques visuelles pour que la hiérarchie puisse améliorer la performance dans des taches de vision par ordinateur. La classification sur des jeux de données hiérarchiques reste un domaine sous exploré dans la littérature de vision par ordinateur mais elle a un grand potentiel. On peut imaginer que même si les erreurs de classification ne peuvent pas être complètement évitées, il est moins grave pour un modèle de classification de prédire une classe qui reste proche hiérarchiquement à la place de la classe attendue au lieu d'une classe qui est très distante hiérarchiquement et qui n'a donc aucune similarité sémantique avec la classe attendue. Afin d'évaluer de

tel systèmes, des métriques hiérarchiques doivent être utilisées à la place des mesures de performance classique tel que la précision. Une mesure de performance hiérarchique devrait être capable d'évaluer si des prédictions faites par un modèle sont proche dans la hiérarchie de la vérité terrain au lieu de juste mesurer si la bonne classe a été prédite où non. La recherche en classification hiérarchique est aussi lié à la résolution de la tache dite de *few shot learning* car l'introduction de nouvelles classes peut être préparées en utilisant la relation hiérarchique entre les nouvelles classes et les classes existantes. La nature hiérarchique des jeux de données est aussi une bonne motivation d'explorer des espaces de projection différent pour les représentations d'images.

Ces particularités poussent la communauté de la télédétection a développer des méthodes pour mettre à profit ou manipuler les spécificités des jeux de données de télédétection. Ces contributions peuvent varier en taille. Elles peuvent parfois requérirent des changements architecturaux comparés aux méthodes plus générales de vision par ordinateur ou alors de concevoir des objectifs d'entrainement sur mesure qui embrassent la nature des données de la tache.

Une zone d'amélioration possible pour certaines taches est de questionner l'espace de plongement par défaut des représentations latentes qui est l'espace Euclidien. En effet, les méthodes d'apprentissage profond n'opèrent que sur des représentations dans l'espace Euclidien en partie grâce à sa distance bien définie et sa représentation numérique naturel. Récemment, ce statu quo a été remis en question avec l'introduction d'alternatives aux espaces Euclidiens pour l'apprentissage profond de représentations tels que les espaces hypersphériques ou hyperboliques. Ces espaces de projection alternatifs sont une direction de recherche prometteuse pour améliorer la performance des modèles dans des tâches spécifiques où ils sont naturellement adaptés de par leur structure inhérente. Par exemple, puisque les espaces hyperboliques sont connues pour leur capacité à projeter des structures d'arbres avec une distorsion minimal (Sarkar, 2011), ils deviennent une alternative séduisante aux espaces Euclidiens pour résoudre des problèmes de classification hiérarchique.

Finalement, en apprentissage de représentations, plusieurs travaux ont proposés avec succès de considérer les échantillons comme faisant partie d'une distribution (Caron et al., 2020 ; Robinson et al., 2020 ; Zbontar et al., 2021). Ce point de vue basé sur les distributions considère que l'encodeur transforme les échantillons du jeu de données d'une distribution vers une autre distribution qui doit idéalement montrer des propriétés discriminatoires par rapport à l'objectif final. Par exemple, en apprentissage contrastif,

nous pouvons considérer que le jeu d'échantillons positifs sont tirés dans le voisinage d'un échantillon parmi la distribution inconnue du jeu de données. Tandis que les échantillons négatifs sont eux tirés parmi l'ensemble de la distribution et représentent donc par définition le jeu de données dans sa globalité. C'est pourquoi nous pouvons considérer l'utilisation d'outils pour mesurer des distances entre distributions empiriques pour concevoir de nouveaux objectifs d'entraînement contrastifs. Le transport optimal est un outil populaire pour mesurer des distances entre distributions. Sa formulation et ses nombreuses variantes font du transport optimal une solution versatile que les pratiquants du machine learning utilisent de plus en plus.

Cette thèse et ces contributions sont placées dans ce contexte avec le but de concevoir des méthodes plus informées pour l'apprentissage de représentations. C'est pourquoi cette thèse a les objectives suivants :

- Étudier l'utilisation de l'apprentissage auto supervisé pour la télédétection.
- Contribuer à réduire le besoin pour les annotations dans le processus d'apprentissage de représentations d'images.
- Proposer des méthodes d'apprentissage de représentations adaptées aux spécificités de la télédétection afin d'entrainer des encodeurs plus efficacement.

Nous contribuons à ces aspects de plusieurs façons. Nos contributions vont de la proposition d'améliorations à l'apprentissage auto supervisé en utilisant des outils tels que le transport optimal à la proposition d'une méthodologie pour l'apprentissage de représentations multimodales avec des applications sur les jeux de données en télédétection de la communauté. Dans la Section B.2, nous décrivons le contenu de ce manuscrit, chapitre par chapitre.

B.2 Aperçu des contributions

Dans cette thèse, nous étudions l'apprentissage de représentations dans le contexte des taches que l'on retrouve en télédétection, allant de la classification de scènes à la détection d'objets où segmentation sémantique. À cette fin, nous tirons parti des outils de l'apprentissage machine tel que le transport optimal. Ce manuscrit est organisé de la façon suivante :

- Dans le chapitre 2, les notions requises pour la compréhension du document sont présentées. Ainsi, les méthodes auto-supervisées de l'état de l'art sont présentées en tant qu'introduction au domaine de l'apprentissage auto supervisé. Nous introduisons également la théorie du transport optimal et ses aspects numériques. En plus du transport optimal général, des variantes du transport optimal sont également présentés tels que le transport optimal entropique, le transport optimal non balancé et les méthodes pour résoudre le problème du transport optimal entre espaces incomparables.

- Le chapitre 3 introduit les travaux que nous avons réalisés afin de tirer parti du transport optimal pour l'apprentissage auto supervisé. Nos contributions s'étalent de la proposition d'une variante de l'apprentissage contrastif basé sur le plan de transport entre échantillons dans une pipeline auto-supervisées à une méthode non contrastive pour obtenir une distribution uniforme sur l'hypersphère qui peut être utilisé comme l'un des deux objectifs nécessaire à l'apprentissage auto supervisé. Nous présentons également une modification à l'apprentissage contrastif dense qui fonctionne en modélisant chaque image comme une distribution de patches spécifiquement pour les tâches denses tels que la détection d'objets où la segmentation sémantique. Tandis que certains des travaux présentés sont actuellement non publiés, la Section 3.2 est basé sur la recherche qui est publié dans l'article de conférence suivant :

  > "*Spherical Sliced-Wasserstein*", C. Bonet, **P. Berg**, N. Courty, F. Septier, L. Drumetz, and M.-T. Pham, in International Conference on Learning Representations, 2023.

- Le chapitre 4 est dédié au problème d'apprentissage de représentations multimodales qui est souvent rencontrées dans le domaine de la télédétection. À ce titre, nous proposons un protocole de pré entraînement contrastif de modèles sur des jeux de données multimodaux. De plus, nous présentons une généralisation supervisée de ce protocole qui nous permet d'améliorer le finetuning spécifique à la tâche par rapport au finetuning par entropie croisée catégorique utilisée traditionnellement. Ce chapitre est basé sur les publications suivantes :

> "*Self-Supervised Learning for Scene Classication in Remote Sensing : Current State of the Art and Perspectives*", **P. Berg**, M.-T. Pham, and N. Courty, Remote Sensing, vol. 14, 16, p. 3995, 2022.

> "*Joint Multi-Modal Self-Supervised Pretraining in Remote Sensing : Application to Methane Source Classification*", **P. Berg**, M.-T. Pham, and N. Courty, in IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2023, pp. 6624 - 6627.

> "*Multimodal Supervised Contrastive Learning in Remote Sensing Downstream Tasks*", **P. Berg**, B. Uzun, M.-T. Pham, and N. Courty, IEEE Geoscience and Remote Sensing Letters, 2024.

- Le chapitre 5 est consacré au problème de classification sur des jeux de données hiérarchiques. Dans de tels jeux de données, les labels de la vérité terrain peuvent être organisés en une hiérarchie qui fournit une connaissance à priori intéressante sur de tels jeux de données. En effet, en fonction de comment la hiérarchie est construite, nous trouvons que les classes proches dans la hiérarchie exposent souvent des caractéristiques visuels similaires qui peuvent aider dans la tâche de classification d'image. À cette fin, nous tirons parti des espaces hyperboliques qui sont des manifolds non Riemannien avec des propriétés intéressantes pour projeter des arbres et hiérarchies. Notre contribution est double. Dans un premier temps, nous introduisons un classifieur hyperbolique basé sur les prototypes idéaux. Ensuite, nous proposons une technique permettant d'utiliser l'information hiérarchique lors de l'initialisation de notre classifieur. Le contenu de chapitre est le sujet de l'article de conférence suivant :

  > "*Horospherical Learning with Smart Prototypes*", **P. Berg**, B. Michele, M.-T. Pham, L. Chapel, and N. Courty, British Machine Vision Conference (BMVC) 2024.

- Dans le chapitre 6, nous résumons nos contributions et concluons le manuscrit en discutant des potentielles perspectives futures dans la lignée de nos contributions. Il s'agit du chapitre final de cette thèse.

Bibliography

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1988). Network flows (cited on p. 37).

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., ... Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. https://doi.org/10.1145/3620665.3640366 (cited on p. 79).

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. *International conference on machine learning*, 214–223 (cited on p. 35).

Atigh, M. G., Schoep, J., Acar, E., Van Noord, N., & Mettes, P. (2022). Hyperbolic image segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4453–4462 (cited on pp. 30, 102, 103).

Bardes, A., Ponce, J., & LeCun, Y. (2022). Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *ICLR* (cited on pp. 11, 27, 118).

Bécigneul, G., & Ganea, O.-E. (2018). Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760* (cited on pp. 35, 92, 100).

Berg, P., Michele, B., Pham, M.-T., Chapel, L., & Courty, N. (2024a). Horospherical learning with smart prototypes. *Proceedings of the British Machine Vision Conference (BMVC)* (cited on p. 109).

Berg, P., Pham, M.-T., & Courty, N. (2022). Self-supervised learning for scene classification in remote sensing: Current state of the art and perspectives. *Remote Sensing*, *14*(16), 3995 (cited on pp. 71, 77, 108).

Berg, P., Pham, M.-T., & Courty, N. (2023). Joint multi-modal self-supervised pretraining in remote sensing: Application to methane source classification. *IGARSS*

2023-2023 IEEE International Geoscience and Remote Sensing Symposium, 6624–6627 (cited on pp. 83, 85, 108).

Berg, P., Pham, M.-T., & Courty, N. (2024b). Apprentissage contrastif multi-modal : Du pré-entrainement auto-supervisé à la classification supervisée. In *Reconnaissance des formes, image, apprentissage et perception (rfiap)*. (Cited on p. 109).

Berg, P., Uzun, B., Pham, M.-T., & Courty, N. (2024c). Multimodal supervised contrastive learning in remote sensing downstream tasks. *IEEE Geoscience and Remote Sensing Letters* (cited on p. 108).

Bonet, C., Berg, P., Courty, N., Septier, F., Drumetz, L., & Pham, M.-T. (2023a). Sliced-wasserstein spherique. In *Conférence sur l'apprentissage automatique (cap)*. (Cited on p. 109).

Bonet, C., Berg, P., Courty, N., Septier, F., Drumetz, L., & Pham, M.-T. (2023b). Spherical sliced-wasserstein. *International Conference on Learning Representations* (cited on pp. 41, 57, 94, 104, 108).

Bonet, C., Chapel, L., Drumetz, L., & Courty, N. (2023c). Hyperbolic sliced-wasserstein via geodesic and horospherical projections. *Topological, Algebraic and Geometric Learning Workshops 2023*, 334–370 (cited on p. 41).

Bonet, C., Drumetz, L., & Courty, N. (2024). Sliced-wasserstein distances and flows on cartan-hadamard manifolds. *arXiv preprint arXiv:2403.06560* (cited on p. 41).

Bonnabel, S. (2013). Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, *58*(9), 2217–2229 (cited on pp. 34, 92).

Brannan, D. A., Esplen, M. F., & Gray, J. J. (2011). *Geometry*. Cambridge University Press. (Cited on p. 30).

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, *34*(4), 18–42 (cited on p. 29).

Busemann, H. (1955). The geometry of geodesics (cited on p. 33).

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). nuScenes: A multimodal dataset for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on pp. 102, 103).

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, *33*, 9912–9924 (cited on pp. 14, 24, 46, 55, 68, 106, 121).

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9650–9660 (cited on pp. 26, 28).

Chang, D., Pang, K., Zheng, Y., Ma, Z., Song, Y.-Z., & Guo, J. (2021). Your "flamingo" is my" bird": Fine-grained, or not. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11476–11485 (cited on p. 30).

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European conference on computer vision (ECCV)*, 801–818 (cited on p. 102).

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. *International conference on machine learning*, 1597–1607 (cited on pp. 11, 23, 28, 49, 54–56, 58–61, 72–74, 77, 78).

Chen, X., Fan, H., Girshick, R., & He, K. (2020b). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (cited on pp. 72–74).

Chen, X., & He, K. (2021). Exploring simple siamese representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15750–15758 (cited on pp. 11, 26, 118).

Cheng, G., Han, J., & Lu, X. (2017). Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, *105*(10), 1865–1883. https://doi.org/10.1109/jproc.2017.2675998 (cited on p. 71).

Chizat, L., Peyré, G., Schmitzer, B., & Vialard, F.-X. (2018). Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, *87*(314), 2563–2609 (cited on p. 42).

Cho, H., DeMeo, B., Peng, J., & Berger, B. (2019). Large-margin classification in hyperbolic space. *The 22nd international conference on artificial intelligence and statistics*, 1832–1840 (cited on p. 30).

Choy, C., Gwak, J., & Savarese, S. (2019). 4d spatio-temporal convnets: Minkowski convolutional neural networks. *CVPR* (cited on p. 102).

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., & Jegelka, S. (2020). Debiased contrastive learning. *Advances in neural information processing systems*, *33*, 8765–8775 (cited on p. 81).

Coates, A., Ng, A., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 215–223 (cited on p. 53).

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37–46 (cited on pp. 72, 109).

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223 (cited on pp. 102, 103).

Courty, N., Flamary, R., Tuia, D., & Rakotomamonjy, A. (2016). Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(9), 1853–1865 (cited on p. 35).

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, *26* (cited on pp. 38, 48).

Defard, T., Setkov, A., Loesch, A., & Audigier, R. (2021). Padim: A patch distribution modeling framework for anomaly detection and localization. *International Conference on Pattern Recognition*, 475–489 (cited on pp. 10, 117).

Delon, J., Salomon, J., & Sobolevski, A. (2010). Fast transport optimization for monge costs on the circle. *SIAM Journal on Applied Mathematics*, *70*(7), 2239–2258 (cited on p. 57).

Dhall, A., Makarova, A., Ganea, O., Pavllo, D., Greeff, M., & Krause, A. (2020). Hierarchical image classification using entailment cone embeddings. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 836–837 (cited on p. 30).

Dhingra, B., Shallue, C. J., Norouzi, M., Dai, A. M., & Dahl, G. E. (2018). Embedding text in hyperbolic spaces. *arXiv preprint arXiv:1806.04313* (cited on p. 30).

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. *International Conference on Computer Vision (ICCV)* (cited on p. 21).

Dong, X., & Shen, J. (2018). Triplet loss in siamese network for object tracking. *Proceedings of the European Conference on Computer Vision (ECCV)* (cited on pp. 22, 23).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (cited on pp. 11, 20, 62, 68, 118).

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, *27* (cited on p. 22).

Ermolov, A., Mirvakhabova, L., Khrulkov, V., Sebe, N., & Oseledets, I. (2022). Hyperbolic vision transformers: Combining improvements in metric learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7409–7419 (cited on p. 30).

Ermolov, A., Siarohin, A., Sangineto, E., & Sebe, N. (2021). Whitening for self-supervised representation learning. *International conference on machine learning*, 3015–3024 (cited on pp. 11, 118).

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, *111*(1), 98–136 (cited on p. 67).

Fan, X., Yang, C.-H., & Vemuri, B. (2024). Horospherical decision boundaries for large margin classification in hyperbolic space. *Advances in Neural Information Processing Systems*, *36* (cited on p. 30).

Fatras, K., Séjourné, T., Flamary, R., & Courty, N. (2021). Unbalanced minibatch optimal transport; applications to domain adaptation. *International Conference on Machine Learning*, 3186–3197 (cited on p. 42).

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., et al. (2021). Pot: Python

optimal transport. *The Journal of Machine Learning Research*, *22*(1), 3571–3578 (cited on p. 100).

Franco, L., Mandica, P., Munjal, B., & Galasso, F. (2023). Hyperbolic self-paced learning for self-supervised skeleton-based action representations. *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=3Bh6sRPKS3J (cited on pp. 29, 30).

Ganea, O., Bécigneul, G., & Hofmann, T. (2018a). Hyperbolic entailment cones for learning hierarchical embeddings. *International Conference on Machine Learning*, 1646–1655 (cited on p. 30).

Ganea, O., Bécigneul, G., & Hofmann, T. (2018b). Hyperbolic neural networks. *Advances in neural information processing systems*, *31* (cited on pp. 30, 33, 99, 100).

Gao, Z., Shi, X., Wang, H., Zhu, Y., Wang, Y. B., Li, M., & Yeung, D.-Y. (2022). Earthformer: Exploring space-time transformers for earth system forecasting. *Advances in Neural Information Processing Systems*, *35*, 25390–25403 (cited on p. 107).

Garnot, V. S. F., & Landrieu, L. (2021). Leveraging class hierarchies with metric-guided prototype learning. *BMVC* (cited on pp. 95, 98, 100–103, 107).

Garrido, Q., Chen, Y., Bardes, A., Najman, L., & Lecun, Y. (2023). On the duality between contrastive and non-contrastive self-supervised learning. *International Conference on Learning Representations* (cited on pp. 28, 48).

Ge, S., Mishra, S., Kornblith, S., Li, C.-L., & Jacobs, D. (2023). Hyperbolic contrastive learning for visual representations beyond objects. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6840–6849 (cited on pp. 29, 30, 106).

Ghadimi Atigh, M., Keller-Ressel, M., & Mettes, P. (2021). Hyperbolic busemann learning with ideal prototypes. *Advances in Neural Information Processing Systems*, *34*, 103–115 (cited on pp. 30, 90, 91, 94, 99–103, 105).

Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (cited on pp. 10, 21, 22, 118).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling,

C. Cortes, N. Lawrence, & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 27). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf (cited on p. 20).

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, *33*, 21271–21284 (cited on pp. 11, 26, 55, 72–74, 118).

Guo, Y., Wang, X., Chen, Y., & Yu, S. X. (2022). Clipped hyperbolic classifiers are super-hyperbolic classifiers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11–20 (cited on p. 91).

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021). Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377* (cited on pp. 11, 20, 28, 118).

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9726–9735. https://doi.org/10.1109/CVPR42600.2020.00975 (cited on pp. 11, 24, 25, 49).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (cited on pp. 59, 67, 79, 97, 100).

Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(7), 2217–2226 (cited on pp. 24, 71).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network (2015). *arXiv preprint arXiv:1503.02531*, *2* (cited on p. 26).

Huh, M., Agrawal, P., & Efros, A. A. (2016). What makes imagenet good for transfer learning? *ArXiv*, *abs/1608.08614* (cited on pp. 10, 19).

Jain, P., Schoen-Phelan, B., & Ross, R. (2022). Self-supervised learning for invariant representations from multi-spectral and SAR images. https://doi.org/10.48550/ARXIV.2205.02049 (cited on pp. 73, 76).

Jakubik, J., Roy, S., Phillips, C., Fraccaro, P., Godwin, D., Zadrozny, B., Szwarcman, D., Gomes, C., Nyirjesy, G., Edwards, B., et al. (2023). Foundation models for generalist geospatial artificial intelligence, 2023. *arXiv preprint arXiv:2310.18660* (cited on p. 107).

Jing, L., Vincent, P., LeCun, Y., & Tian, Y. (2021). Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348* (cited on p. 22).

Jing, L., & Tian, Y. (2020). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *43*(11), 4037–4058 (cited on pp. 20, 71).

Jung, H., Oh, Y., Jeong, S., Lee, C., & Jeon, T. (2021). Contrastive self-supervised learning with smoothed representation for remote sensing. *IEEE Geoscience and Remote Sensing Letters*, 1–5. https://doi.org/10.1109/LGRS.2021.3069799 (cited on p. 74).

Kantorovich, L. V. (1942). On the translocation of masses. *Dokl. Akad. Nauk. USSR (NS)*, *37*, 199–201 (cited on p. 36).

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised contrastive learning. *Advances in neural information processing systems*, *33*, 18661–18673 (cited on pp. 81, 83, 88, 105).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (cited on pp. 60, 92, 100).

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *CoRR*, *abs/1312.6114* (cited on p. 20).

Kochurov, M., Karimov, R., & Kozlukov, S. (2020). Geoopt: Riemannian optimization in pytorch. (Cited on p. 100).

Kosmopoulos, A., Partalas, I., Gaussier, E., Paliouras, G., & Androutsopoulos, I. (2015). Evaluation measures for hierarchical classification: A unified view and novel approaches. *Data Mining and Knowledge Discovery*, *29*, 820–865 (cited on p. 101).

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. (Cited on p. 53).

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images (cited on pp. 59, 61, 94, 99, 100).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90 (cited on pp. 79, 97).

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, *2*(1-2), 83–97 (cited on p. 51).

Lê, H.-A., Berg, P., & Pham, M.-T. (2024). Box for mask and mask for box: Weak losses for multi-task partially supervised learning. *Proceedings of the British Machine Vision Conference (BMVC)* (cited on p. 109).

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, 2980–2988 (cited on p. 86).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755 (cited on p. 67).

Liu, S., Chen, J., Pan, L., Ngo, C.-W., Chua, T.-S., & Jiang, Y.-G. (2020). Hyperbolic visual embedding learning for zero-shot recognition. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9273–9281 (cited on p. 93).

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., & Tang, J. (2021). Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* (cited on p. 20).

Long, T., Mettes, P., Shen, H. T., & Snoek, C. G. (2020). Searching for actions on the hyperbole. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1141–1150 (cited on p. 30).

López, F., Pozzetti, B., Trettel, S., Strube, M., & Wienhard, A. (2021). Symmetric spaces for graph embeddings: A finsler-riemannian approach. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning, ICML 2021, 18-24 july 2021, virtual event* (pp. 7090–7101, Vol. 139). (Cited on p. 30).

Loustau, B. (2020). Hyperbolic geometry. *arXiv e-prints*, arXiv–2003 (cited on p. 30).

Mémoli, F. (2011). Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, *11*, 417–487 (cited on pp. 42, 95).

Mettes, P., Ghadimi Atigh, M., Keller-Ressel, M., Gu, J., & Yeung, S. (2024). Hyperbolic deep learning in computer vision: A survey. *International Journal of Computer Vision*, 1–25 (cited on p. 30).

Mettes, P., Van der Pol, E., & Snoek, C. (2019). Hyperspherical prototype networks. *Advances in neural information processing systems*, *32* (cited on p. 93).

Miller, G. A. (1998). *Wordnet: An electronic lexical database*. MIT press. (Cited on pp. 89, 93).

Mizrahi, D., Bachmann, R., Kar, O., Yeo, T., Gao, M., Dehghan, A., & Zamir, A. (2024). 4m: Massively multimodal masked modeling. *Advances in Neural Information Processing Systems*, *36* (cited on pp. 88, 107).

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, 666–704 (cited on p. 35).

Moreira, G., Marques, M., Costeira, J. P., & Hauptmann, A. (2023). Hyperbolic vs euclidean embeddings in few-shot learning: Two sides of the same coin. *arXiv preprint arXiv:2309.10013* (cited on p. 91).

Neumann, M., Pinto, A. S., Zhai, X., & Houlsby, N. (2019). In-domain representation learning for remote sensing. (Cited on p. 72).

Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, *30* (cited on p. 30).

Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. *ECCV* (cited on pp. 10, 22, 118).

Ohri, K., & Kumar, M. (2021). Review on self-supervised image recognition using deep neural networks. *Knowledge-Based Systems*, *224*, 107090 (cited on p. 20).

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. (2023). Dinov2: Learning robust

visual features without supervision. *arXiv preprint arXiv:2304.07193* (cited on pp. 29, 46, 55, 69, 105, 106).

Peng, W., Varanka, T., Mostafa, A., Shi, H., & Zhao, G. (2021). Hyperbolic deep neural networks: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, *44*(12), 10023–10044 (cited on p. 30).

Pennec, X. (2020). 3 - manifold-valued image processing with spd matrices. In X. Pennec, S. Sommer, & T. Fletcher (Eds.), *Riemannian geometric statistics in medical image analysis* (pp. 75–134). Academic Press. (Cited on p. 30).

Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trendső in Machine Learning*, *11*(5-6), 355–607 (cited on pp. 25, 35).

Piran, Z., Klein, M., Thornton, J., & Cuturi, M. (2024). Contrasting multiple representations with the multi-marginal matching gap. *arXiv preprint arXiv:2405.19532* (cited on pp. 48, 54).

Rabin, J., Peyré, G., Delon, J., & Bernot, M. (2012). Wasserstein barycenter and its application to texture mixing. *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*, 435–446 (cited on p. 39).

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. *International conference on machine learning*, 8748–8763 (cited on pp. 48, 88, 107).

Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. (Cited on p. 21).

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, *66*(336), 846–850 (cited on p. 28).

Razakarivony, S., & Jurie, F. (2016). Vehicle detection in aerial imagery: A small target detection benchmark. *Journal of Visual Communication and Image Representation*, *34*, 187–203 (cited on pp. 13, 120).

Redko, I., Vayer, T., Flamary, R., & Courty, N. (2020). Co-optimal transport. *Advances in Neural Information Processing Systems*, *33*(17559-17570), 2 (cited on pp. 44, 53).

Robinson, J., Chuang, C.-Y., Sra, S., & Jegelka, S. (2020). Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (cited on pp. 14, 49, 81, 121).

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., & Breitkopf, U. (2012). The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences; I-3*, *1*(1), 293–298 (cited on pp. 13, 120).

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65 (cited on p. 28).

Rubner, Y., Tomasi, C., & Guibas, L. (1998). A metric for distributions with applications to image databases. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 59–66. https://doi.org/10.1109/ICCV.1998.710701 (cited on p. 35).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y (cited on pp. 10, 19, 62, 89, 93, 117).

Sander, M. E., Ablin, P., Blondel, M., & Peyré, G. (2022). Sinkformers: Transformers with doubly stochastic attention. *International Conference on Artificial Intelligence and Statistics*, 3515–3530 (cited on pp. 50, 68).

Sarkar, R. (2011). Low distortion delaunay embedding of trees in hyperbolic plane. *International symposium on graph drawing*, 355–366 (cited on pp. 14, 30, 121).

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., & Cox, D. D. (2019). On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, *2019*(12), 124020 (cited on p. 27).

Scheibenreif, L., Hanna, J., Mommert, M., & Borth, D. (2022a). Self-supervised vision transformers for land-cover segmentation and classification. *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1422–1431 (cited on pp. 83–85).

Scheibenreif, L., Mommert, M., & Borth, D. (2022b). Contrastive self-supervised data fusion for satellite imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *3*, 705–711 (cited on p. 76).

Schmitt, M., Hughes, L. H., Qiu, C., & Zhu, X. X. (2019). Sen12ms–a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion. *arXiv preprint arXiv:1906.07789* (cited on p. 83).

Schneider, M., Broszeit, A., & Körner, M. (2021). Eurocrops: A pan-european dataset for time series crop type classification. *arXiv preprint arXiv:2106.08151* (cited on pp. 13, 120).

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823 (cited on p. 54).

Shi, L., Fan, J., & Yan, J. (2024). Ot-clip: Understanding and generalizing clip via optimal transport. *Forty-first International Conference on Machine Learning* (cited on p. 48).

Shi, L., Zhang, G., Zhen, H., Fan, J., & Yan, J. (2023). Understanding and generalizing contrastive learning from the inverse optimal transport perspective. *International conference on machine learning*, 31408–31421 (cited on pp. 48, 54).

Shimizu, R., Mukuta, Y., & Harada, T. (2020). Hyperbolic neural networks++. *arXiv preprint arXiv:2006.08210* (cited on p. 30).

Surís, D., Liu, R., & Vondrick, C. (2021). Learning the predictability of the future. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12607–12617 (cited on p. 30).

Tifrea, A., Bécigneul, G., & Ganea, O.-E. (2018). Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546* (cited on p. 30).

Tran, Q. H., Janati, H., Courty, N., Flamary, R., Redko, I., Demetci, P., & Singh, R. (2023). Unbalanced co-optimal transport. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(8), 10006–10016 (cited on p. 44).

Uscidda, T., & Cuturi, M. (2023). The monge gap: A regularizer to learn all transport maps. *International Conference on Machine Learning*, 34709–34733 (cited on p. 48).

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., & Belongie, S. (2018). The inaturalist species classification and detection dataset. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8769–8778 (cited on p. 89).

van Spengler, M., Berkhout, E., & Mettes, P. (2023). Poincar\'e resnet. *arXiv preprint arXiv:2303.14027* (cited on pp. 30, 32).

Vayer, T., Chapel, L., Flamary, R., Tavenard, R., & Courty, N. (2020). Fused gromov-wasserstein distance for structured objects. *Algorithms*, *13*(9), 212 (cited on p. 48).

Villani, C., et al. (2009). *Optimal transport: Old and new* (Vol. 338). Springer. (Cited on p. 35).

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th international conference on Machine learning*, 1096–1103 (cited on p. 20).

Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset (cited on pp. 89, 92–94, 99, 100).

Wang, T., & Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *International Conference on Machine Learning*, 9929–9939 (cited on pp. 47, 48, 56, 58–61, 94).

Wang, X., Zhang, R., Shen, C., Kong, T., & Li, L. (2021). Dense contrastive learning for self-supervised visual pre-training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3024–3033 (cited on pp. 25, 47, 62, 64, 67).

Wang, Y., Albrecht, C. M., Braham, N. A. A., Mou, L., & Zhu, X. X. (2022a). Self-supervised learning in remote sensing: A review. *arXiv preprint arXiv:2206.13188* (cited on p. 71).

Wang, Y., Albrecht, C. M., & Zhu, X. X. (2022b). Self-supervised vision transformers for joint SAR-optical representation learning. *IGARSS 2022* (cited on p. 76).

Wills, G. J. (1999). Nicheworksinteractive visualization of very large graphs. *Journal of computational and Graphical Statistics*, *8*(2), 190–212 (cited on p. 96).

Wu, Z., Xiong, Y., Yu, S. X., & Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3733–3742 (cited on pp. 10, 23, 118).

Yerxa, T., Kuang, Y., Simoncelli, E., & Chung, S. (2023). Learning efficient coding of natural images with maximum manifold capacity representations. *Advances in Neural Information Processing Systems*, *36*, 24103–24128 (cited on pp. 27, 54, 55).

Yokoya, N., Ghamisi, P., Hansch, R., & Schmitt, M. (2020). Report on the 2020 ieee grss data fusion contest-global land cover mapping with weak supervision [technical committees]. *IEEE Geoscience and Remote Sensing Magazine*, *8*(4), 134–137 (cited on p. 83).

Zbontar, J., Jing, L., Misra, I., LeCun, Y., & Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. *International Conference on Machine Learning*, 12310–12320 (cited on pp. 11, 14, 27, 47, 49, 51, 54, 55, 72–74, 105, 118, 121).

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. *ECCV* (cited on pp. 10, 21, 118).

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595 (cited on pp. 10, 97, 117).

Zhu, B., Lui, N., Irvin, J., Le, J., Tadwalkar, S., Wang, C., Ouyang, Z., Liu, F. Y., Ng, A. Y., & Jackson, R. B. (2022). Meter-ml: A multi-sensor earth observation benchmark for automated methane source mapping. *arXiv preprint arXiv:2207.11166* (cited on pp. 13, 71, 78, 79, 83, 84, 87, 120).

COLLEGE MATHSTIC
DOCTORAL BRETAGNE
BRETAGNE OCEANE

Université Bretagne Sud

Titre : Contributions à l'apprentissage de représentations en vision par ordinateur et télédétection

Mot clés : Apprentissage de représentation, Télédétection.

Résumé : L'apprentissage profond est devenu un outil incontournable pour la résolution de tâches d'analyse d'images, notamment dans le domaine de la télédétection. En conséquence, les besoins en données annotées ont considérablement augmenté. Cependant, l'annotation de données peut être coûteuse en temps et en moyens. Ainsi, tout un champ de la littérature s'intéresse à l'apprentissage de représentations d'images en réduisant la dépendance aux annotations par des méthodes dites autosupervisées. Les représentations apprises sont ensuite exploitables pour des tâches de vision grâce à leur nature discriminante par rapport aux labels de la tâche finale. Dans ce contexte, nous évaluons dans cette thèse comment ces méthodes peuvent être exploitées dans le domaine de la télédétection en s'intéressant à des tâches telles que la classification de scène multimodale pour laquelle nous proposons une méthode d'apprentissage autosupervisé. Nous mettons à profit le problème du transport optimal pour modéliser certains problèmes et proposer des contributions méthodologiques à l'apprentissage contrastif. Finalement, nous proposons d'aller au-delà des espaces Euclidiens pour l'apprentissage de représentation en proposant une méthode de classification dans les espaces hyperboliques. Notre méthode qui est informée par la hiérarchie du jeu de données permet d'améliorer les performances en classification hiérarchique.

Title: Contributions to Representation Learning in Computer Vision and Remote Sensing

Keywords: Representation Learning, Remote Sensing.

Abstract: Deep Learning has become an ubiquitous tool for the resolution of image analysis tasks, notably in the remote sensing domain. As such, the need for annotated data have largely increased. But, annotating data can be costly and time consuming. Therefore, a whole field of the literature is dedicated to image representation learning by decreasing the dependence to annotations using so called self-supervised methods. The learnt representations are then usable in downstream tasks because of their discriminative nature with respect to the labels. In this context, we evaluate in this thesis how these methods can be exploited in the remote sensing domain by investigating tasks such as multi-modal scene classification for which we propose a self-supervised framework. We leverage the optimal transport problem to model several problems et propose new methodological contributions to contrastive learning. Finally, we propose to go beyond Euclidean spaces for representation learning by proposing a new classification method in hyperbolic spaces. Our method which is hierarchically-informed improves the performance in hierarchical classification.