ABSTRACT

JIANG, BO. From Neuronal to Artificial Neural Network: Discovering Non-linearity in the Inference Problems. (Under the direction of Hamid Krim.)

In recent years, the machine learning field has made significant progress in developing complex systems that can help computers learn and understand the chaotic world in different tasks, such as image classification, domain adaptation, and natural language processing. One of the most widely-used machine learning approaches, the deep neural network, follows the working principles of neurons in brains. Although some deep connections and information transfer details among neurons are still unknown, the artificial neural network achieves an outstanding performance in different tasks by encoding the real-world data into a latent space. In addition, a big assumption for most existing approaches is that latent representations of a dataset lie on a flat space, where linear measurement or Euclidean distance is widely applied to help decrease the computational complexity in practice. A representation space should be a number of connected opening sets, where the representations of each object are not independent but related. The representation space, therefore, is presented by a low-dimensional manifold embedded in a higher-dimensional space. This linear approximation to the geodesic distance ignores much non-linear information.

In Chapter 2, we first explore the dynamics and high-order information contained in neuronal signals, that pairwise connection and high-order interactions (i.e., three or more neurons) among neurons should reflect received external messages. This chapter aims to infer a dynamic graph learning model of time-varying measurements by formulating it as an optimization problem, which is a quadratic objective function and tensor information of observed node signals over short time intervals. The proper regularization constraints reflect the graph smoothness and other dynamics involving the underlying graph's Laplacian, as well as the time evolution smoothness of the underlying graph. The resulting joint optimization is solved by a continuous relaxation of the weight parameters and an introduced novel gradient-projection scheme. We apply our algorithm to a real-world dataset comprising recorded activities of individual brain cells. The resulting model is not only viable but also efficiently computable.

In Chapter 3, we introduce a new statistical perspective, exploiting the Jaccard similarity metric as a measure-based metric to effectively invoke non-linear features in the loss of self-supervised contrastive learning. Specifically, our proposed metric may be interpreted as a dependence measure between two adapted projections learned from the so-called latent representations. This is in contrast to the cosine similarity measure in the conventional contrastive learning model, which accounts for correlation information. To the best of our knowledge, this effectively non-linearly fused information embedded in the Jaccard similarity, is novel to self-supervision learning with promising results. The proposed approach is compared to two state-of-the-art self-supervised contrastive learning methods on three image datasets. We demonstrate not only its amenable applicability in current ML problems but also its improved performance and training efficiency.
In Chapter 4, we take advantage of Optimal Transport (OT) ’s power and flexibility, which have pervaded a wide spectrum of problems, including recent Machine Learning challenges such as unsupervised domain adaptation. Its essence of quantitatively relating two probability distributions by some optimal metric has been creatively exploited and shown to hold promise for many real-world data challenges. In a related theme in the present work, we posit that domain adaptation robustness is rooted in the respective data's intrinsic (latent) representations, which are inherently lying in a non-linear submanifold embedded in a higher dimensional Euclidean space. We account for the associated geometric properties by refining the $L^2$ Euclidean metric better to reflect the geodesic distance between two distinct representations. We integrate a metric correction term and a prior cluster structure in the source data in the OT-driven adaptation. We show that this is tantamount to an implicit Bayesian framework, which we demonstrate to be viable for a more robust and better-performing approach to domain adaptation.
DEDICATION

This thesis work is dedicated to my parents, who always love and support me unconditionally, and they are good examples of teaching me to work hard and be patient in everything I chase. I am genuinely thankful for being your child. This thesis work is dedicated to my girlfriend, Patricia, who has been a constant source of support and encouragement during this long tough pandemic. Your happiness helps me steadfastly walk on this journey far away from home without hesitation.
**BIOGRAPHY**

Bo Jiang received his B.S. degree in Biomedical Engineering (BME) from Tsinghua University, Beijing, China, in 2016. He, collaborating with a local hospital, developed a visual typing system based on motion event-related potential (ERP) in the V1 area of the brain. In August 2017, he started his Ph.D. in the Department of Electrical and Computer Engineering (ECE) at North Carolina State University and joined Vision, Information, and Statistical Signal Theories and Applications (VISSTA) group under the direction of Dr. Hamid Krim. His current research interests include interdisciplinary research of learning from the essence of brains and applying it to an artificial model, the theoretical basis of the machine learning area, and their applications for computer vision. More specifically, he is working on developing and improving deep learning methods for different kinds of image problems, such as conventional image classification and domain adaptation problems. Moreover, recovering the non-linearity measure (e.g., distance on the manifold) is one of his main directions, such as defining simplex (higher-order information) in the networks, using Jaccard similarity between latent representations to include all-moment information, geodesic recovery, and optimal transport theory.
ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Hamid Krim, for your invaluable advice, continuous support, and patient assistance at every stage of the research project. Your immense knowledge and plentiful experience have encouraged and inspired me all the time in my academic research.

I also thank the rest of my advisory committee: Dr. Derya Cansever, Dr. Edgar Lobaton, Dr. Radmila Sazdanovic, Dr. Tianfu Wu, and Dr. Chau-Wai Wong. I am truly thankful for your questions and advice. You guided me into this beautiful area from different aspects, such as neural networks, image processing, and topology. I also thank the general support of the ARO grant. I am also very grateful to Dr. Spencer Smith and Dr. Yiyi Yu for our collaborations and research on neuronal data.

I appreciate working with Dr. Ashkan Panahi, the postdoctoral researchers in the VISSTA Laboratory, and for his guidance and careful teaching at the start of my Ph.D. career. Last but not least, I greatly valued all the discussions and lengthy debates within the VISSTA Laboratory. Shahin Mahdizadehaghdam, Yuming Huang, Wen Tang, Kenneth Tran, Siddharth Roheda, Tanmay Asthana, and Sally Ghanem have been great officemates and friends.

Finally, I am deeply grateful to my parents, my girlfriend, Patricia Peng, and my furry children, Cooper and Purple. There are also many others who could not be mentioned here, and I would like to say thanks to all of you.
# TABLE OF CONTENTS

## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td></td>
</tr>
</tbody>
</table>

## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>vii</td>
<td></td>
</tr>
</tbody>
</table>

## Chapter 1 INTRODUCTION

- Introduction
- Problem Setup and Background
  - Definition of Graph and Graph Laplacian
  - Topological Basics
- Dynamic Graph Learning
  - Static Graph Learning
  - Dynamic Graph Learning
  - Dynamic Graph Learning from Sparse Signals
  - Dynamic Graph Learning with Higher-Order Information
- Algorithm Numerical Solution
  - Proximal Operator
  - Projection Method
  - Algorithm
- Experiments and Results
  - Computational Complexity
  - Synthetic Data Generation
  - Neuronal-Activity Data
- Discussion
- Future Work
- Conclusions

## Chapter 2 Dynamic Graph Learning: A Structure-Driven Approach

- Introduction
- Problem Setup and Background
  - Definition of Graph and Graph Laplacian
  - Topological Basics
- Dynamic Graph Learning
  - Static Graph Learning
  - Dynamic Graph Learning
  - Dynamic Graph Learning from Sparse Signals
  - Dynamic Graph Learning with Higher-Order Information
- Algorithm Numerical Solution
  - Proximal Operator
  - Projection Method
  - Algorithm
- Experiments and Results
  - Computational Complexity
  - Synthetic Data Generation
  - Neuronal-Activity Data
- Discussion
- Future Work
- Conclusions

## Chapter 3 Refining Self-Supervised Learning: Beyond Linear Metric

- Introduction
- Related Works
- Method
  - Notations
  - Contrastive Loss
  - A Variation on a Theme: A Jaccard-index Assessment
- Experimental Results
  - Experiment Setup
  - k-NN and Linear Evaluation
  - Evaluation Results
  - Experiment Setup
  - k-NN and Linear Evaluation
  - Evaluation Results
  - Learning performance of different projectors

---

v
Chapter 4  Implicit Bayes Adaptation: A Collaborative Transport Approach  
4.1  Introduction  
4.2  Background  
4.2.1  Notation  
4.2.2  Unsupervised Domain Adaptation  
4.2.3  Optimal Transport  
4.2.4  Geodesic Distance  
4.3  Equalizing Domain Variation: An Optimal Transport Approach  
4.3.1  Cluster-to-Cluster Optimal Transport  
4.3.2  Relaxation of CCOT  
4.3.3  Collaborative Optimal Transport  
4.3.4  Algorithmic Solution  
4.4  Experiments and Results  
4.4.1  Datasets  
4.4.2  Experiments and Results  
4.5  Conclusions  

Chapter 5  CONCLUSION AND FUTURE WORK  
5.1  Conclusion  
5.2  Future Work  

BIBLIOGRAPHY  
APPENDIX  
Appendix A  Additional experiment details and results in Chapter 3  
A.1  Training Setup Details  
A.2  Experiment Results  
A.2.1  MoCo vs. Our model  
A.2.2  Different initialization of Projectors  
A.2.3  Non-linearity based on Different Samples  

## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Synthetic data results</td>
<td>19</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>Comparison between related stimuli with and without simplex information</td>
<td>25</td>
</tr>
<tr>
<td>Table 2.3</td>
<td>Correlation between pairwise connections without 2-simplex connectivity.</td>
<td>26</td>
</tr>
<tr>
<td>Table 2.4</td>
<td>Correlation between pairwise connections with 2-simplex connectivity.</td>
<td>27</td>
</tr>
<tr>
<td>Table 2.5</td>
<td>Correlation between 2-Simplex connectivity.</td>
<td>27</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Linear evaluation results on MoCo with different pre-defined $\tau$s.</td>
<td>43</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>Our model takes slightly more time compared with SimCLR, which is reason-</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>able based on its learning performance.</td>
<td></td>
</tr>
<tr>
<td>Table 3.3</td>
<td>Linear evaluation results on different datasets.</td>
<td>45</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Digit number results</td>
<td>61</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Office-31 results</td>
<td>62</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Office-Home results</td>
<td>62</td>
</tr>
<tr>
<td>Table A.1</td>
<td>Our modified model takes slightly more time compared with MoCo.</td>
<td>75</td>
</tr>
</tbody>
</table>
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>Vietoris-Rips Complex and Cech Complex.</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Computational Complexity Test.</td>
<td>15</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Distribution comparison: (a) The distribution of data intensity of neuronal activity data. (b) The distribution of data intensity of synthetic data. (c) Distribution of trial-to-trial variability of neuronal data. (d) Distribution of trial-to-trial variability of synthetic data. (e) Distribution of correlations between neurons of neuronal data. (f) Distribution of correlations between neurons of synthetic data.</td>
<td>17</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>ROC curve: (a–c): ROC curve for our model (DGL) in different states. (d–f): ROC curve for using the Pearson correlation (PC) in different states.</td>
<td>19</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>Visual Stimuli: The visual stimuli for the mouse in a single trial.</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Neurons’ Signals and Associated Positions.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 2.7</td>
<td>Correlation matrix between graphs.</td>
<td>23</td>
</tr>
<tr>
<td>Figure 2.8</td>
<td>Comparison between related stimuli with and without simplex information:</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Second and sixth stimuli are related stimuli. The yellow triangle stands for 2-simplex connections. (a) The graph representation for the second stimulus with simplex information. (b) The graph representation for the sixth stimulus with simplex information. (c) The graph representation for the second stimulus without simplex information. (d) The graph representation for the sixth stimulus without simplex information.</td>
<td></td>
</tr>
<tr>
<td>Figure 2.9</td>
<td>Pairwise connectivity comparison.</td>
<td>28</td>
</tr>
<tr>
<td>Figure 2.10</td>
<td>2-Simplex connectivity comparison.</td>
<td>29</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>An overview of the self-supervised learning model. Contrastive loss is calculated in the hidden space. Encoders’ performance are evaluated in the latent space.</td>
<td>32</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>An intuitive explanation of our model. Latent representations on the manifold are projected to two different hidden spaces, which may capture different information. Contrasting positive with negative keys is applied in each hidden space. Jaccard similarity provides non-linear fusion between two hidden spaces.</td>
<td>37</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>An overview of our Jaccard-based self-supervised learning model. One additional projector is added to give a different aspect of extracting features. Two hidden spaces are fused based on Jaccard similarity.</td>
<td>38</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>An overview of our Jaccard-based self-supervised learning model. One additional projector is added to give a different aspect of extracting features. Two hidden spaces are fused based on Jaccard similarity.</td>
<td>44</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>To see how the encoder learns, two shark images are selected to be original object and positive key in the training set, while two whale images, a similar class to shark, are chosen to be negative keys, where one is similar to original object, and the other is less similar to that.</td>
<td>46</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Cosine similarity is calculated between positive keys, similar negative keys, and less similar negative keys in the latent space. Our model shows a better distance measuring between positive pairs and similar negative pairs.</td>
<td>47</td>
</tr>
</tbody>
</table>
Figure 3.7 Cosine similarity is calculated between positive keys, similar negative keys, and less similar negative keys in the hidden spaces. In our model, one projector can be seen as a complement to the other, so overall they will give better distance measuring between positive pairs and similar negative pairs. 47

Figure 3.8 Removing the linear feature between latent representations of the original object, positive key, and similar negative key. Conventional SSL model doesn't retain additional information, while our model preserves a higher-moment similarity. 48

Figure 4.1 A comparison between Euclidean and Geodesic distance. 53
Figure 4.2 Illustration of multiple classifiers. 57
Figure 4.3 Examples of datasets. 61

Figure A.1 In this additional example, our result shows that our model also preserve the non-linearity compared to conventional SSL. 76
Figure A.2 Similarity between the positive pair and negative pairs in the latent space. 77
Figure A.3 Similarity between the positive pair and negative pairs in the hidden space. 77
Figure A.4 In this additional example, our result shows that our model also preserve the non-linearity compared to conventional SSL. 78
Nowadays, one natural question to think about is how humans learn about the complex and chaotic world. One general answer is that humans develop the ability to abstract some basic descriptions from different aspects of observations and phenomena. One then extracts common traits and attributes from data, gleans, and combines distinct concepts. The purpose of such an abstraction and extraction is to purify, separate, and transform received information into specific definitions, and humans can then summarize and generalize the connections among things, events, and phenomena.

To learn more about how the human brain learns, neurons or nerve cells, as a single unit in the brain, transmit and process the information received from our sense organs, such as vision, hearing, and touching. Brains have a network of nerves made up of great amounts of neurons, which transfer excitation/electrical impulses from one neuron to the next [Hin01]. Specifically, in the process of neuronal information transmission, after a stimulus from a sense organ activates a neuron, dendrites carry the information, formed by a set of spikes, to the nerve cell’s nucleus, soma. The electrical action potential is then processed and transmitted to the axon, which carries the electrical information from the soma to the synapse. The synapse finally passes the impulse to the dendrites of the second neuron as an activation signal. As a result, brains build a complex network of neurons to pass, encode, and understand external stimuli. Algorithms for machine learning employ the same notion of a neural network of neurons. In this instance, an artificial neural network is composed of connections among a huge set of artificial neurons, which are defined on a computer instead of biological brains. The information transmission functions of artificial neurons are similar to the working principles of neurons found in brains, e.g., activation functions.

In the recent Machine Learning area, one of the most common tasks, classification, aims to assign a class label to an unseen object based on a set of labeled samples [Tan16; Tan18; Tan19a].
Most experiments [Yua21] are based on the labeled dataset, which is time-consuming and labor-intensive to build. In addition, the trained models can hardly be generalized to other tasks. Therefore, an intelligent model [Tan21; Tan22] is required to extract more general and clean representations, and the newly designed tasks are similar to human learning procedures. Advanced classification methods [Tan19b] thus involve adapting knowledge of already-known objects to recognition of new ones via contrasting or minimizing the gap. Based on getting the label information, whether from the dataset itself or a source dataset that shares the same label space but quite different latent spaces, the two categories of classification tasks in Machine Learning are thus called self-supervised contrastive learning and unsupervised domain adaptation, respectively.

In the second chapter, we [Jia21b; Jia21a] first focus on exploring the working principles of the brain and its single unit, neurons. We take advantage of the increasing and widespread use of graphs, which offer an excellent method for measuring the interaction between various parts in a wide range of network-based applications. Analyzing and discovering the underlying structure of a graph for a given dataset has become central to various signal processing research problems with such a potential interpretation. The rather recent connectome paradigm [Spo12] in neuroscience is based on the hypothesis that the brain is a massively connected network among tremendous amounts of neurons. The behavioral variation of the network connectivity, as a result of an external stimulus, can be utilized to study the anatomy and functionality of the brain [Pol11; Ock17].

Recovering pairwise connectivity based on observed neurons’ signals currently follows two main approaches: (1) Noise correlation analysis [Yu18; Som01] is commonly used in neuroscience to uncover the connectivity among neurons. The idea is to build the connection by constantly observing unexpected spikes (noise) simultaneously of two neurons. (2) Static graph learning [Mar17; Che17] is used to extract a fixed graph over time. These two methods are likely to return the true neuronal connectivity but ignore the time-varying. Additionally, the functional connectivity would give more details of the neuronal working principles following an experiment with specific stimuli in different time periods. Moreover, topological data analysis (TDA) [Was16], capturing the metric-insensitive characteristics of the data has been demonstrated to be a meaningful measure of a graph's shape or connectome [Bal05]. Higher-order information [Gua16], using simplices and homology [Wag89], was proposed to determine the dominant homology in addition to investigating pairwise connections between every two nodes. This allowed for an interpretation of a topological 2-simplex (a tighter connection among three neurons) as the efficiency of information diffusion [Lee14]. The TDA method is, however, more practical and widely applied in analyzing the shape of the data, but it is challenging to build up a differential optimization model to consider the shape/characteristics as an underlying constraint associated with distributed data.

In this work, we start from the basis of prior research work in neuroscience [VF14] and graph learning [Che17], and propose an optimization model with a $L_1$-norm dynamics constraint to track neurons’ temporal connectivity, as well as higher-order information (three neurons’ connection), referred to as 2-simplex information. In the end, our proposed dynamic graph learning will reflect the neurons’ connectivity by graph edges and simplexes. An edge/2-simplex attribute of the learning
graph is relaxed from a matrix of booleans to be continuous on \([0, 1]\), which is defined as the probability of connection between a pair of/three neurons.

In the later work \([\text{Jia22}]\), the idea of learning latent representations of data has been a major motivation in the computer vision area, where a visual representation in the latent space serves as a meaningful base for supervised learning and inference. Scaling learning-based inference to large datasets (e.g., ImageNet), however, is a labor-intensive and time-consuming task and calls for a more viable alternative. In addition, supervised learning is a bottleneck for training a more generalized model that can work on multiple tasks, where labels are mostly corresponding to a specific task. There are also some areas that are hard to obtain enough labeled data, such as medical images \([\text{Chu19}]\). To address the fore-mentioned issues, a more useful artificial intelligence (AI) system should extract and capture a purer and deeper representation beyond what is labeled in the training set. Inspired by the fact that infants learn the new concept/object largely only by observation \([\text{Bai94}]\), a hypothesis of the process of biological intelligence is to involve observing the world from different aspects, comparing it with related things, and building hypotheses to categorize the observations based on the error (similarity and dissimilarity). As an approximation of biological intelligence in the machine, Self-Supervised Learning (SSL) seeks to acquire labels from different modalities of a fraction of the totality of a given data sample, which enables to recognize and learn more precise and common representations of the world.

In the third chapter, we started from the existing contrastive learning model \([\text{Cho05}]\), which minimized a semantic distance between images from the same class while maximizing that from different classes. A \(L_2\)-norm distance is applied to measure the similarity between two representations \([\text{Ham03}]\). The features extracted by non-linear processing across the learning layers should be elements lying on some manifold rather than on a linear space. Similarity measurement based on a linear metric is an approximation to the true geodesic distance on the manifold between two points. To that end, we propose using the Jaccard similarity coefficient, developed in the early 1900’s \([\text{Jac12}]\), to account for the intrinsic non-linearity. It can capture non-linear characteristics in the set similarity by considering the representations of an object as a finite set of general features. On the basis of interpreting SSL as a hypothesis problem, with a quantitative decision in contrastive learning, we propose a bi-projector Jaccard-based fusion model which effectively reflects the non-linearities in the hidden spaces. Our proposed adaptive approach effectively aims at addressing the inherent modal structure of the latent/pseudo-invariant characteristics of the data, and we provide a simple framework in practice. The modified model is compared to two state-of-the-art self-supervised learning models SimCLR \([\text{Che20b}]\) and MoCo \([\text{He20}]\), based on both the k-nearest neighbors (k-NN) and a standard linear classification methods (single-layer neural network) on selected datasets, CIFAR-10, CIFAR-100, and Tiny-ImageNet-200. We demonstrate that the model can improve training efficiency and achieve competitive performance in most experiments.

Similar to the reason for developing self-supervised learning, unsupervised domain adaptation is another field in computer vision to reduce time-consuming labeling tasks, where the idea is to train a meaningful embedding function and a classifier on a labeled source data, adapt the model to an
unlabeled target data and achieve a good performance as well. The key to solving this unsupervised domain adaptation problem is to minimize the distance between the source and target domains using different distance measurements, such as Maximum Mean Discrepancy (MMD) [Lon15], Correlation Alignment [Sun16], and Kullback-Leibler (KL) divergence [Zhu15]. Given its flexible amenability to assessing imbalance between distributions, optimal transport, as a distance metric, has also been chosen to consider the distance between complex distributions [Dam18; Xu20; Fat21; Red17; Pat15]. In the existing approaches, the distance between source and target domains may be affected by the uncertainty of one training classifier and its predicted labels. Moreover, the computation of transport cost is often considered as the distance between samples in a high-dimensional Euclidean space. These data points, as we discussed in the early paragraphs, should be distributed over a manifold rather than from a distribution with support over all of $\mathbb{R}^n$ [Li19; Mal18; SOL16]. Then, a natural question to ask is how we can reconstruct or estimate the underlying manifold through some measured characteristics based on some concepts of the Riemannian manifold.

In the fourth chapter, we propose an OT-based cost formulation and a dual-map model for unsupervised domain adaptation to address the aforementioned issues in the existing approaches. The modified cost first helps preserve the underlying multi-class structure in each domain, where the true transport matrix, upon classification, should have a block-diagonal structure. Our modifications to a Euclidean distance are intended to provide a more accurate estimation of the geodesic distance on the manifold due to the intrinsicality of the latent representations to a submanifold embedded in a higher dimensional Euclidean space. Then, highlighting the importance of reducing the dependence on the classifier performance and presenting the feasibility and convergence of multiple classifiers, a dual collaborative OT framework seeks to define a differential measure of goodness between an "agnostic" OT and a cluster-driven and implicitly Bayesian OT to optimize the domain adaptation and shrink the searching space.

The balance of the thesis first proceeds with an optimization model of dynamic graph learning in Section 2. Section 3 discusses a proposed refined similarity in self-supervised learning. In section 4, we propose a dual OT-map learning model with geometric properties in the unsupervised domain adaptation problem, and we have some concluding remarks in Section 5.
CHAPTER 2

DYNAMIC GRAPH LEARNING: A STRUCTURE-DRIVEN APPROACH

2.1 Introduction

The increased and ubiquitous emergence of graphs provides a great approach for quantifying interaction between different elements in a great variety of network-based applications. Analyzing and discovering the underlying structure of a graph for a given dataset has become central to various signal processing research problems with such a potential interpretation. For example, in social media [Jag14] (e.g., Facebook and LinkedIn), the basic interaction/relation between two individuals being represented by a link, yields the notion of a graph known as the social network, which is used for inferring further characteristics among all involved individuals [Hua19]. The power of the graph representation paradigm is in fact so fundamental that it captures the very structure of life [Aud03; Can03], namely the interactions among atoms/molecules to reflect the behavior of different materials or bacteria. The rather recent connectome paradigm [Spo12] in neuroscience is based on the hypothesis that the brain is a massively connected network due to its activity and connectivity. The network connectivity, hence behavioral variation, is in response to an external stimulus, may be used to investigate a brain's structure and its functionality [Pol11; Ock17]. In our validation of this framework, we consider sparse images of the brain's visual cortex where non-zero components vary in time. Our proposed approach is just as applicable to dense images as well as medium-dense images such as Functional Magnetic Resonance Imaging data. Our ready access to and experience in the data at hand thus influenced the choice.

Establishing connectivity among neurons, currently follows two main approaches, (i) Noise
correlation analysis [Yu18; Som01], often used in neuroscience to uncover the connectivity among neurons, (ii) Static graph learning [Mar17; Che17] used to extract a fixed graph over time. When noise correlation is significant, it is commonly used in neuroscience as a short-time connectivity metric between every pair of neurons. Requiring abundant observations, this method cannot, however, yield an acceptable and reliable connectivity estimate over an observation interval. Additionally, the acquired connectivity does not yield simple rationalization following an experiment with a specific stimulus. To that end, and for additional interpretational potential, graph structures that are learned from data have been of great research interest. Research in this direction has pervasively been based on graphs’ topology and signals’ smoothness, as well as the application of the graph Laplacian. Other recent work includes deep neural network modeling, for which training/testing was performed on graph datasets to generate a graph, representing patterns under given signals ultimately. These studies have primarily focused on static graphs with non-sparse signals, as well as on the consistency assumption of the graphs over time. These models require sufficiently adequate samples for training and testing, once again making it difficult to use neuronal signals whose sample size is typically small for adequately assessing variations over rather short time intervals. Graphs’ dynamics with a potential impact on temporal data characterization have also been of much interest to [Goy19; Goy18]. In this theme, the models are used to predict the links given previous graphs and associated signals. All these approaches require much prior information on known structures and sufficient data for training and predicting future graphs. Moreover, game-theoretic techniques, which have been proposed to address multi-objective optimization problems [Zha11], may be seen as potentially useful in our use case by interpreting our interacting neurons as a multi-player game. It is, however, difficult to achieve a single comprehensive model with a capacity to capture the neurons’ large variability and time-varying characteristics.

To analyze the characteristics and representation of a graph, topological data analysis has been shown to measure the shape or connectome of a graph [Bal05]. In addition to investigating pairwise connections between every pair of nodes, higher-order information, using simplices and homology, was proposed to determine the prevailing homology, thus affording an interpretation of a topological hole/cycle as the inefficiency of (or slow down) information diffusion [Lee14]. While this approach is practical when analyzing a given graph, it is more challenging to account for differential constraints to optimize the construction of a graph associated with distributed data.

In this work, we build on the wealth of prior research work in neuroscience as well as in graph learning [Che17], to propose a new model, with a target dynamic structure to track neurons’ temporal connectivity, as well as higher-order connectivity (three neurons connected sets), referred to as 2-simplex information. To that end, our proposed dynamic graph learning will include vertices/nodes with their connectivity reflected by graph edges. An edge attribute is defined as the probability of connection between a pair of neurons, and that of a 2-simplex to reflect the connectivity degree of pairwise connection for all three neurons in a 2-simplex.

To proceed, the outline of our paper follows, in order, our contributions in the sequel. Firstly, exploiting the insight from prior research on graph learning with graph Laplacian [Che17; Don16],
we propose a formulation to estimate an optimal graph short time intervals, which in turn reflects the evolving transformation of the connectivity. Secondly, we modify our model to fit sparse signals to verify our optimized solution on a neuronal signal dataset. Thirdly, we call on a tensor to model connectivity tracking among three neurons to account for higher-order information. Fourthly, we propose three alternative methods to simplify the optimization procedures so that the optimization problems’ solution procedures are simplified and help reach the optimal points. We finally proceed to test our proposed model using a neuronal dataset, to improve our understanding of the neuronal interaction and their process of signal/information propagation.

2.2 Problem Setup and Background

Throughout the paper, we will adopt an upper and lower case bold letter to respectively denote a matrix and a vector, and the superscripts $T$, $-1$ to respectively denote a matrix transpose and inverse. The operator $tr(\cdot)$ denotes a matrix trace. The identity, zero and “1” matrices are respectively denoted by $I$, $0$ and $1$, while $x_{ij}$ represents the $i$-th row, $j$-th column element of a matrix $X$.

2.2.1 Definition of Graph and Graph Laplacian

Our neuronal-activity dataset will be $N$-dimensional, and will be characterized by a connectivity graph $G = \{V, E, W\}$, where $V$ denotes the vertex set $V = \{v_1, v_2, ..., v_N\}$, $E$ is the edge set with attributes reflecting the connectivity between each pair of vertices quantified by $0 \leq w \leq 1$, $\forall w \in W$ as a connectivity strength. A time series $y_n(t)$ of observations with $t = 1, 2, ..., T_{all}$, is associated with each node $v_n$. For simplicity, we will aggregate the nodes’ finite-length time series into a $N \times T_{all}$ matrix $Y$, where $(Y)_{t,n} = y_n(t)$. Our problem formulation will seek for each observed $Y$, either a static graph $G$ or a time dependent graph series of graphs $G_1, G_2,...$

The well-known graph Laplacian of an undirected graph can help describe its topology, and can serve as the second derivative operator of the underlying graph. The corresponding Laplacian matrix $\mathcal{L}$ is commonly defined as [Chu97], with $\mathcal{L}(i, j) = -w_{ij}$, for $i, j$ adjacent nodes, and $\mathcal{L}(i, i) = 0$ otherwise, and $\mathcal{L}(i, i) = d_i$, where $d_i = \sum_j w_{ij}$ denotes the degree of node $i$. Its simple matrix expression is $\mathcal{L} = D - W$, where $D$ is a diagonal matrix of degrees.

The Laplacian matrix may also usefully adopt, in some context, a second derivative interpretation of graphs: Given an assignment $x = (x_1, x_2, ..., x_N)^T$ of real numbers to the nodes, the Laplacian matrix may be found as the second derivative of $x$ as $\mathcal{L}(W) = \sum_i \sum_{j>i} w_{ij} a_{ij} a_{ij}^T$, where $a_{ij}$ denotes a $N$-dimensional vector whose elements are all $0$s, except the $i$-th element being respectively $1$ and $j$-th element $-1$. As may be seen, $a_{ij}$ represents the first derivative of the edge between the $i$-th and $j$-th node. The notion of a Laplacian will be exploited in this sequel for a given data set as a structural regularizer when an optimal graph is sought.
2.2.2 Topological Basics

Euler characteristic originally proposed for polyhedron is also invoked in the homology theory, with a motivation of comparing two shapes by taking into account any dimensional holes proper to their topology. For instance, the difference between a disk and a circle in homology is that a one-dimensional hole exists in a circle. In this sense, homology theory provides an alternative definition of shapes in topological spaces by decomposing a shape $S$ into a family of abelian groups $H_0(S), \ldots, H_n(S)$ [Mac12].

In geometry, a simplex is a general concept of a fully connected shape in any dimensions, where a $k$-simplex is a $k$-dimensional polyhedron of $k+1$ vertices. As an example, a disk can be simplified and assimilated to a 0-simplex, while a circle is distinguished as a chain complex with a one-dimensional hole. Two different strategies are used to construct simplex-based structures for computational purposes in homological data analysis. In a so-called Vietoris-Rips complex, two nodes are said to be connected to each other if their neighborhoods (balls around the node) overlap, thus making the polytope a simplex. In a Čech complex, a simplex is, on the other hand, defined if and only if any given node is connected to any other nodes with overlapping of all neighbors. Therefore, a simplex in a Čech complex is a stricter connection in any dimension. These simplices are illustrated in Figure 2.1.

In our neuronal-activity dataset, we translate the connectivity among three neurons into 2-simplex with the criteria in Čech complex with a mathematical equation, and the result gives higher order connection information beyond pairwise connectivity.

![Vietoris-Rips Complex and Čech Complex](image_url)
2.3 Dynamic Graph Learning

2.3.1 Static Graph Learning

Prior to proposing the dynamic structure learning of a graph, we briefly revisit the basic notions of static graph learning [Che17]. Using the Laplacian quadratic form \( \text{tr}(X^T \mathcal{L}(W)X) \) as a smoothness regularizer of the data \( x_n \), and the degree of connectivity \( K \) as a tuning parameter, ref. [Che17] discovers a \( K \)-sparse graph from noisy signals \( Y \). This is the result of solving the following,

\[
\begin{align*}
\arg\min_{X,W} & \ |Y-X|^2 + \text{tr}(\gamma X^T \mathcal{L}(W)X) \\
\text{s.t.} & \ 0 \leq w_{ij} \leq 1, \ \forall i,j, \ \sum_{i,j>i} w_{ij} = K,
\end{align*}
\]

where \( \gamma \) and \( K \) are tuning parameters, \( X \) is the noiseless signals, and \( Y \) its noisy observation. \( W \) is the adjacency weight matrix for the undirected graph, with the additional relaxation of the individual weights to the interval \([0, 1]\).

2.3.2 Dynamic Graph Learning

Please note that in [Che17], a single connectivity graph is inferred over the entire observation time interval, thus overlooking the practically varying connections between every pair of nodes over time. To account for these events evolving over short term intervals and hence capture the true underlying structure of the graph, we propose to learn the graph’s dynamics. Learning these dynamics is consistent with our goal of modeling the brain signals to elicit timely information. These would particularly aim to pick up the responses to corresponding stimuli in that time interval and account for their dependence on previous graph instances and time intervals. This also introduces a practical constraint which needs to be accounted for by way of the similarity of temporally adjacent graphs in the overall functionality of the sequence of graphs in congruence with the observed data by selecting a 1-norm distance of connectivity weight matrices between consecutive time intervals, we can proceed with the graph sequence discovery, and hence the dynamics by seeking the solution to the following,

\[
\begin{align*}
\arg\min_{X_t,W_t} & \ |Y_t-X_t|^2 + \text{tr}(\gamma X_t^T \mathcal{L}(W_t)X_t) \\
+ & \alpha \sum_{t=1}^{T-1} |W_t - W_{t+1}|_1 \\
\text{s.t.} & \ 0 \leq w_{t,ij} \leq 1, \ \forall i,j, \ \sum_{i,j>i} w_{t,ij} = K,
\end{align*}
\]

where \( \alpha \) is a penalty coefficient, \( Y = [Y_1, \ldots, Y_T] \), \( Y_t \) is the observed data in the \( t \)-th time interval, \( X_t \) is the corresponding noise-free data, \( W_t \) is the weight matrix in the \( t \)-th time interval, and \( K \) is a...
tuning parameter.

2.3.3 Dynamic Graph Learning from Sparse Signals

The solution given by [Che17] provides the static graph learning, but the observed signals $Y$ may often be sparse, which poses a problem: noting that $tr\{X^T \mathcal{L}(W)X\} = tr\{\sum_{i} \sum_{j>i} w_{ij} x_i^T a_j a_j^T X\}$ is the Laplacian quadratic form, we use $w_{ij} \|x_i - x_j\|^2$ to reflect the distance between two signals, whose minimization will unveil some problematic nodes with very similar signals over a small time interval.

Technical and Practical Constraints

While the above formulation (2) captures the principles of the dynamics of interest, additional practical difficulties arise in the case when $x_i$ and $x_j$ are close to 0, making their distance close to 0. This, as a result, introduces unexpected false edges when sparse signals are present. Such an instance may arise for, say, given sparse signals written as $Y = [\tilde{Y}, 0]^T$, where the dimension of $Y$ is $N \times t$, $\tilde{Y}$ is an $n \times t$ matrix and $0$ is $(n - N) \times t$. Given that 2-norm is non-negative and the Laplacian matrix is positive semi-definite, we can find a trivial optimal solution of $(X, W)$, where $W$ is sparse, such that $X = Y$, and the weight matrix can be represented by some block matrix, $W = \begin{bmatrix} 0 & 0 \\ 0 & W \end{bmatrix}$.

Given that our given graph learning problem is a convex and non-negative problem, one can show that the optimal loss value is 0 by inserting the solution $Y = X$ and $W$. This indicates that the presence of sparse signals (which happens to be the case for brain firing patterns) results in a solution of the formulated optimization that may not be unique. Additionally, this leads to optimal points that establish connectivity between zeros-signal nodes (i.e., erroneous and perhaps meaningless connections).

Proposition 1.

\[
L_{DGL}(X_t, W_t) = \sum_{t=1}^{T} \left[ \|Y_t - X_t\|^2 + t r\{\gamma X_t^T \mathcal{L}(W_t)X_t\} \right] - 2\eta \sum_{i,j>i} w_{ij}(\|x_{t,i}\|^2 + \|x_{t,j}\|^2) + \alpha \sum_{t=1}^{T-1} \|W_t - W_{t+1}\|_1 \tag{2.3}
\]

\[s.t. \quad 0 \leq w_{ij} \leq 1, \quad \forall i, j, \quad \sum_{i,j>i} w_{ij} = K,
\]

where $\eta$ is a penalty coefficient, $x_{t,i} = (x_{1,i}, ..., x_{t,i})^T$ is the $i$-th node's signal in the $t$-th time interval, and $X_t = [x_{t,1}, x_{t,2}, ..., x_{t,N}]^T$.

Proving this proposition is tantamount to lifting the fore-noted difficulty of sparse signals. To that end, we introduce a constraint term to help focus on the nodes with significant values, specifically we constrain edge nodes energy to be of relevance.
Since the weight matrix for an undirected graph is symmetric, therefore this additional part of the new optimization can be simplified as follows:

\[
P = -2\eta \sum_{i,j>i} w_{t,i,j} (\|x_{t,i}\|^2 + \|x_{t,j}\|^2)
\]

(2.4)

\[
P = -t r(X_t^T \eta D(W_t) X_t)
\]

where \(D(W_t)\) is a diagonal matrix defined above from weight matrix \(W_t\). Combining the two \(t r(\cdot)\) expressions from Equations (3) and (4) will yield the simpler form. With a little more attention, one could note that this procedure naturally prefers nodes of higher energy by associating a higher weight.

### 2.3.4 Dynamic Graph Learning with Higher-Order Information

The quadratic form of \(X_t\) over \(W_t\) in optimization model (3) accounts for pairwise connections between nodes. To exploit higher order than pairwise interaction among nodes requires a topology-driven information structure across nodes in a graph. To that end, and to bypass the computational complexity of homology assessment of the topological complex resulting from a set of nodes, we propose a practically computable alternative in the optimization model which quantifiably reflects the connectivity among three neurons, thus capturing the inherent information within a 2-simplex.

**Proposition 2.**

\[
L_{DG\text{LH}}(X_t, W_t, W_t^{(3d)}) = \sum_{t=1}^{T} \left( \|Y_t - X_t\|^2 + \gamma X_t^T L(W_t) X_t \right) - 2\eta_1 \sum_{i,j>i} w_{t,i,j} (\|x_{t,i}\|^2 + \|x_{t,j}\|^2) + \alpha \sum_{t=1}^{T-1} \|W_t - W_{t+1}\|_1 + \tau \sum_{i,j,k} w_{t,i,j,k} (\|x_{t,i} - x_{t,j}\|^2 + \|x_{t,j} - x_{t,k}\|^2)
\]

(2.5)

\[
-\eta_2 \sum_{i,j,k} w_{t,i,j,k} (\|x_{t,i}\|^2 + \|x_{t,j}\|^2 + \|x_{t,k}\|^2),
\]

s.t. \(0 \leq w_{t,i,j} \leq 1, \quad \forall i, j, \quad \sum_{i,j>i} w_{t,i,j} = K_1, \quad 0 \leq w_{t,i,j,k} \leq 1, \quad \forall i, j, k, \quad \sum_{i,j>i} w_{t,i,j,k} = K_2.
\]

In the basic Vietoris-Rips complex, if three given neurons are connected to each other, we will consider these three neurons as a 2-simplex. We hence propose the following strategy for considering all connections among three neurons in the \(t\)-th time interval: \(w_{t,i,j,k} (\|x_{t,i} - x_{t,j}\|^2 + \|x_{t,i} - x_{t,k}\|^2 + \|x_{t,j} - x_{t,k}\|^2).\) The formulation is in fact a direct result of the simplex definition in a VR-complex,
making the proximity between every pair of signals in a triplet of nodes, a 2-simplex is generated. In other words, we regard this 2-simplex information as a tight connection among these three neurons.

As with the development for pair-wise node interaction, we proceed to tackle the 3-way node interaction by introducing a weight attribute tensor \( W^{(3d)}_t = [w^{(3d)}_{t, ij k}] \), when recalling the non-negative characteristic of \( \|x_{t,i} - x_{t,j}\|^2 + \|x_{t,i} - x_{t,k}\|^2 + \|x_{t,j} - x_{t,k}\|^2 \). Our inclusion of a constraint \( \sum_{i,j,k} w^{(3d)}_{t, ij k} = K \) to force to construct the tightest higher order connections may hence be achieved. As with the afore-discussed issue about sparse signals yielding non-realistic connections among three weak/zero node signals, we proceed to constrain such a solution out by adding a energy penalty term at each node: \( -w^{(3d)}_{t, ij k} (\|x_{t,i}\|^2 + \|x_{t,j}\|^2 + \|x_{t,k}\|^2) \). By accounting for all these factors, we obtain Proposition 2 which considers both the pairwise connection and a 2-simplex connection/information.

### 2.4 Algorithm Numerical Solution

Solving Propositions 1 and 2 by the conventional Lagrangian duality is computationally complex, and an alternative is in order. To that end, we propose a coupled pair of computational steps to solve Equation's optimization (5). The first facilitates solving the optimization by the periodic updates of the proximal operator as detailed, and the other seeks to relax graph adjacency weight terms to be continuous on \([0, 1]\) below.

#### 2.4.1 Proximal Operator

\( W_t \) are updated with gradient descent method. On account of the non-smoothness of the \( l_1 \) norm in Equation (5), we call on the proximal operator to first rewrite model (3) and proceed to solve it [Com08; Par14]. Firstly, the \( l_1 \) term \( \|W_t - W_{t+1}\|_1 \) in optimization (3) may be affected by the order of updating \( W_t \)'s. To reduce the impact of the order of updating variables, we introduce an auxiliary variable \( Z_t \), with the constraint that of the order of updating variables, we introduce \( Z_t \) to replace the this term, and add a new constraint that \( Z_t = W_t - W_{t+1} \). This substitution allows an update of \( W_t \) with no influence by the others weight matrices, and \( Z_t \) provides the relaxation between each pair of adjacency weight matrices. This results in an equivalence to the previous optimization problem, with the advantage a reduced impact on the optimization caused by the variables updating order. We introduce \( \beta_t \) as the Lagrange multiplier of the equality constraints \( Z_t = W_t - W_{t+1} = 0 \).

**Claim:** As a result, the Lagrangian duality form of the optimization can hence be re-expressed as,

\[
L(W_t, X_t, Z_t, \gamma, \eta, \alpha, \beta_t) = \sum_{t=1}^{T} \|Y_t - X_t\|^2 + tr(X_t^T (\gamma \mathcal{L}(W_t) - \eta \mathcal{D}(W_t))X_t) + \alpha \sum_{t=1}^{T-1} \|Z_t\|_1 + \langle \beta_t, Z_t - W_t + W_{t+1} \rangle.
\]
We now have a function of \( Z_t \), denoted as \( f(Z_t) = \alpha \|Z_t\|_1 + \langle \beta_t, Z_t \rangle \), which is a convex and non-smooth function over \( Z_t \). To alleviate the numerical difficulty of the non-differentiable contribution to Equation (6), we adopt the proximal operator approach to search for an optimal \( Z_t \). The function is defined as \( \text{prox}_{\lambda f}(V_t) = \arg\min_{Z_t} f(Z_t) + \frac{1}{2\lambda} \|Z_t - V_t\|_2^2 \), where \( \lambda \) is a tuning parameter, which can be interpreted as the gradient step in the proximal algorithm. It is clear that we achieve the optimal point \( Z_t^* \), if and only if we have \( Z_t^* = \text{prox}_{\lambda f}(Z_t^*) \), therefore for the \( k \)-th iteration, we update the variable \( Z_t \) as \( Z_{k+1} = \text{prox}_{\lambda f}(Z_k) \).

### 2.4.2 Projection Method

To address this issue, we associate a subspace structure \( W \) to the set of constraints, where \( W \) is the whole weight space for graphs, with \( w_{ij} \geq 0 \), and \( W \subset W' \), such that \( 0 \leq w_{ij} \leq 1 \). We next construct a projection procedure of \( W_t \in W' \) onto \( W \) to ensure a subspace membership of the next iterate. Considering an updated weight matrix as a point in a high dimensional space, we minimize its distance to the subspace within the whole space by enforcing \( \min_{\tilde{W}_t} \frac{1}{2} \sum_{i,j>\ell} (\tilde{w}_{t,ij} - w_{t,ij})^2 \), \( s.t. \sum_{i,j>\ell} \tilde{w}_{t,ij} = K \) and \( \tilde{W}_t \in W \). A similar membership-set projection can equally be applied to \( W_t^{(3d)} \) by vectorizing the associated tensor. Applying the Lagrangian Duality on this minimization problem yields,

\[
\begin{align*}
L_{\text{Proj}}(W_t, \kappa) &= \frac{1}{2} \sum_{i,j>\ell} (\tilde{w}_{t,ij} - w_{t,ij})^2 \\
&\quad + \kappa \left( \sum_{i,j>\ell} \tilde{w}_{t,ij} - K \right), \\
&\quad \text{s.t. } W_t \subset W. \\
\end{align*}
\]

(2.7)

\[
\begin{align*}
L_{\text{Proj}}(\tilde{W}_t^{(3d)}, \kappa') &= \frac{1}{2} \sum_{i,j,k} (\tilde{w}_{t,ijk} - w_{t,ijk})^2 \\
&\quad + \kappa' \left( \sum_{i,j,k} \tilde{w}_{t,ijk} - K' \right), \\
&\quad \text{s.t. } \tilde{W}_t^{(3d)} \subset W'^{(3d)}. \\
\end{align*}
\]

### 2.4.3 Algorithm

With the \( Z_t \) update in hand, we proceed to unfold the various steps of the algorithmic solution of model (5). The functional dependent on \( X_t \) being convex smooth allows the calculation of the differential of the optimization formulation over \( X_t \) and and yielding the following iteration.

\[
X_t^{(k)} = (1 + \gamma \mathcal{L}(W_t^{(k-1)}) - \eta D(W_t^{(k-1)}))^{-1} Y_t. 
\]

(2.8)

Since the functional depending on \( W_t \) are smooth, we use gradient descent to update each \( W_t \). The whole algorithm is presented in Algorithm 1.
Algorithm 1 algorithm for dynamic graph learning

1: **Input:** \( Y_t \)
2: **Output:** \( X_t, W_t \)
3: \( \alpha, \gamma, \eta, \lambda \) and learning rate \( \tau \) are pre-defined.
4: **while** not converged **do**
5: Update \( X_t \) by (8)
6: \( W_t^{(k)} = W_t^{(k-1)} - \tau \frac{\partial}{\partial W_t} L_{DGLH}(W_t, X_t, Z_t, \gamma, \eta, \alpha, \beta_t) |_{W_t^{(k-1)}} \)
7: Project \( W_t^{(k)} \) to the defined domain by Projection method.
8: Update \( W_t^{(3d)} \).
9: Project \( W_t^{(3d)} \) to the defined domain by Projection method.
10: Update \( L(W_t^{(k-1)}) \) and \( D(W_t^{(k-1)}) \) by definition.
11: Update \( Z_t \) by Proximal operator.
12: \( \beta_t^{(k)} = \beta_t^{(k-1)} - \tau_2 \frac{\partial}{\partial \beta_t} L_{DGLH}(W_t, X_t, Z_t, \gamma, \eta, \alpha, \beta_t) |_{\beta_t^{(k-1)}} \)
13: **end while**

2.5 Experiments and Results

The following experiments on neural activity data also provide, as a result, a new analysis and a potential new exploratory tool in neuroscience and machine learning. The neuronal activity data was provided by Dr. S. L. Smith’s Laboratory at UCSB.

2.5.1 Computational Complexity

The computational complexity is defined as the execution time of our program which is, in turn, dependent on the number of nodes \( N \) as well as on the number of time intervals \( T \). We fix all other parameters and vary the choice of \( N \) and \( T \), respectively, to discuss our model’s complexity. The results are shown in Figure 2.2.

2.5.2 Synthetic Data Generation

Modeling and analyzing experimental/real data of neural activity with inference objectives were the primary motivation of this work. Our goal of thoroughly evaluating our model of neural activity is unfortunately met with limited amounts of real data available for validation, as noted, which led us to generate a set of synthetic signals from pre-defined connections based on biological criteria. This affords us a characterization and retrieval of all associated structural features to compare our modeled-based results with graph representations obtained using Pearson correlation-based edge connection. This is widely used in neuroscience to evaluate neuronal connectivity of observed signals.
Figure 2.2 Computational Complexity Test.

(a) Executive Time vs. Number of Neurons

(b) Executive Time vs. Number of Intervals
2.5.2.1 Synthetic Data Model

Following computational neuroscience guidelines \cite{Kas18}, we postulate the following model to generate signals mirroring the relevant dynamics,

\[ y_i(t) = p(x_i(t)I(x_i)) + n_i(t), \]  

(2.9)

where \( y_i(t) \) is the measured spiking signal of \( i \)-th neuron at time \( t \), \( x_i(t) \) represents the number of supposed arrival spikes generated from Poisson Distribution, \( I(x_i) \) is an indicator function, which indicates the \( i \)-th neuron firing/not, \( p \) function is a probabilistic process for determining each spike firing/not, and \( n_i(t) \) represents a measurement noise.

The number of arrival spikes is described by a probabilistic model, where the probability of a firing spike in an interval \( [t, t + \Delta t] \) is in direct proportion to \( \Delta t \) \cite{Kas18},

\[ P(s = 1 | (x, x + \Delta t)) = \lambda \Delta t, \]  

(2.10)

\( \lambda \) is the firing rate for generating the number of arriving spikes, hence, parameterizing a Poisson distribution \( \lambda \).

The indicator function \( I(\cdot) \) supports the non-uniform firing of neurons due to the same stimulus. The fact that only a subset of the neurons in a given region are active at each trial, indicates that the proportion of event-related neurons should be a tuning parameter.

\( p \) function imitates the idea of the perceptron model in the visual cortex, making the output the result of a weighted summation of neurons’ contributions in the previous layer, i.e., \( y = \sigma(w_1 x_1 + \cdots + w_n x_n + w_0) \), with \( w_i \) describing the degree of influence of each neuron, \( w_0 \) as some bias, and \( \sigma(\cdot) \) representing an activation function. In the proposed model, the probability of each neuron \( i \) firing/not is dependent on its neighborhood’s neurons signals/contributions and computed as follows: \( P(s_i = 1 | graph) = P(s_i = 1 | N(i)) = P(s_i = 1 | x_1, x_2, \ldots) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + \ldots) \), where \( s_i \) denotes \( i^{th} \) neuron spike, \( N(i) \) its \( i^{th} \) neighborhood. The combined model accounts for the probability of spike firing and the Poisson nature of the generated spikes, together with the Bernoulli process of a turn on, yields under a connectivity schedule, some generated information.

The measurement noise \( n_i(t) \), assumed white and Gaussian \((\mu, \sigma^2)\), was experimentally determined to best fit the generative process.

2.5.2.2 Synthetic Data

To first evaluate the viability of the proposed model, we qualitatively show the behavior of the model’s respective densities and of the real data in Figure 2.3. We also show the degree of correlation between the neuronal data for various trials and the degree of the correlation between different neurons in the same trial. The original intensities, X-axis, are all normalized between 0 and 1 using \( \frac{\text{sig-min}}{\text{max-min}} \), Y-axis presents the number of times that the values occurred within the intervals set by the X-axis. The results are shown in Figure 2.3.
Figure 2.3 Distribution comparison: (a) The distribution of data intensity of neuronal activity data. (b) The distribution of data intensity of synthetic data. (c) Distribution of trial-to-trial variability of neuronal data. (d) Distribution of trial-to-trial variability of synthetic data. (e) Distribution of correlations between neurons of neuronal data. (f) Distribution of correlations between neurons of synthetic data.
2.5.2.3 Simulated Connectivity Graph

To evaluate the effectiveness of generating neuronal networks by merely using the Pearson correlation, we manually subdivide the signals into 3 intervals. We use the same constraint parameter $K$, where $\sum_{i,j} w_{t,ij} = K$ for our model. At the same time, we select $K$ largest correlation values after calculating the Pearson correlation between every two nodes’ signals. Upon applying the same process on $N$ synthetic sets of signals, we obtain the graph representation sets $W_{t,i}$ for our model, and $\tilde{W}_{t,i}$ using Pearson correlation, $t$ representing $t$-th time interval, and $i$ denoting $i$-th trial. By calculating $W_{t,i} = \sum_i W^i_{t,i}$ and $\tilde{W}_{t,i} = \sum_i \tilde{W}^i_{t,i}$, we determine the consistency of edge connectivity over trials by defining a threshold $t h_w$ to count those edges repeating more than $t h_w$ times.

To get an additional assessment of our activity network generation model against that obtained by Pearson correlation, we use the receiver operating characteristic (ROC) curves and further illustrate the two methods’ capabilities. We use 100 synthetic trials and record connections in each trial over each time interval. By choosing different threshold $t h_w$, we select $N$ consistent edges and then count the number of correct edges $n$ based on the ground truth, which is set to 30 in our synthetic data experiment. The total number of edges of an undirected graph among 50 nodes is 1225. Therefore, the ROC curve is true positive rate, $\frac{n}{30}$, vs. false negative rate, $\frac{N-n}{1225-30}$.

From Figure 2.4, the ROC curve for graphs using the Pearson correlation shows that it is close to a random-select model. In other words, the Pearson correlation can hardly recover the correct connectivity among all nodes. Compared with the Pearson correlation method, our model shows a significantly better performance in every state. Even with limited synthetic data trials, our model can also restore the essential connectivity (ground truth) accurately. Our model recovers the essential graphs for each state with 10 generated trials, by choosing those edges repeating more than 5 times. As shown in Table 2.1, the correct number of edges number in each state is 30, the number of selected edges from our model for each state is shown in the third column, and the number of correct edges for each state is shown in the last column. Those edges with a high repetition rate infer the essential structure of the underlying connectivity.
Table 2.1 Synthetic data results

<table>
<thead>
<tr>
<th>State</th>
<th>Ground Truth</th>
<th># Connectivity Selection in Our Model</th>
<th># Correct Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start state</td>
<td>30</td>
<td>66</td>
<td>20</td>
</tr>
<tr>
<td>Mid state</td>
<td>30</td>
<td>59</td>
<td>18</td>
</tr>
<tr>
<td>End state</td>
<td>30</td>
<td>57</td>
<td>21</td>
</tr>
</tbody>
</table>

2.5.3 Neuronal-Activity Data

Neurons, as the brain’s basic active elements, receive and transmit information to a center and one another (typically the pre-frontal cortex in the primates). Information collected by sensory organs, such as visual information, olfactory, etc., is encoded into neural signals, which are membrane potentials, which are, in turn, propagated in the brain for decision making. The widely accepted model distinguishes the brain’s visual sensorial part as the visual cortex. The visual cortex [Cri95; Nie10], a widely studied part of the brain for its importance and relevance here, is often used to explore the cause-effect of visual stimuli. The visual cortex study may comprise several layers, the first being the so-called V1 area, known for its directional sensitivity. The two cardinal features of neurons in the V1 area are driven by 2 types, orientation selectivity and direction selectivity.
[Hub62], where orientation stands for the stimulus angle outline, and direction stands for the motion tendency. Most neurons in the V1 area are orientation-selective, which means that they are highly active under a specific angle of the stimulus for both directions. Some neurons are direction-selective, where they are sensitive to a specific angle and moving direction of a stimulus. As a result, models are constructed to explain motion detection in the primary visual cortex [Ade85; Bra08], which suggest that a few simple cells with quadrature phases and directions jointly define and transfer the information of motion direction to the higher-level visual cortex.

2.5.3.1 Real Experimental Data

The data were collected in S. L. Smith's Laboratory at UCSB [Yu18]. A two-photon electro-microscope [Ji16] was used to collect fluorescent signals. The whole experiment consists of 3 specific scenarios with a 20 trial measurement, with the same stimuli in each trial. The stimuli for each of the scenarios are shown in Figure 2.5, consisting of a “gray” movie, an artificial movie, and natural movie 1 and movie 2 (which we will not discuss here and defer to future work). The dataset includes 590 neurons in the V1 area, and the sample rate is approximately 13.3 Hz. In the artificial movie data we have used in our experiments, each stimulus is characterized by an orientation and pattern motion direction. This is adapted to be detectable and trackable in the brain V1 area. We qualify two stimuli as similar if they share both orientation and direction of pattern motion. The stimuli are deemed related if they share the same orientation but differ in the pattern motion direction.

2.5.3.2 Pre-Processing

The presumed and accepted neuronal redundancy in the brain is equivalent to observing a diverse neuronal activity pattern for the exact same stimulus. As is also known, additive noise causes unexpected firings of neurons. Moreover, much noise, such as unexpected firing, is contained in signals. To better cope with such variability and improve our analysis quality, we select the 50 most active and consistent neurons from the 590 neurons in the V1 area. Specifically, the selection is...
based on the highest correlation of each neuron activity across trials. We consider these neurons as
the most consistently functioning neurons due to the same stimulus over trials. Neuronal signals
stimulated by the artificial movie in two different trials are presented in Figure 2.6a, where X-axis
reflects time, Y-axis indexes the 50 neurons, while the brightness reflects the normalized data
intensities. Figure 2.6b depicts the 50 selected neurons’ relative positions, where numbering reflects
the consistency (i.e., #1 is the most consistently active neuron).
Figure 2.6 Neurons’ Signals and Associated Positions.
2.5.3.3 Interval Partitioning of Data

The brain’s reaction time for stimuli is approximately 100ms, and the delay of the device is around 50 to 100 ms; the time difference between 2-time points is 75 ms; therefore, we choose $T = 213$ in the optimization model to capture the change within 150 ms, and we have 25 to 26 graphs for each stimulus. We choose $K = 30$ (5 percent of the total edge number of a complete graph) to enforce a sparse graph. Uniformly adopting these parameters on the data across 20 trials, we obtain as a result, 213 graphs for each trial $W_1^t, \ldots, W_{20}^t$, where $t = 1, \ldots, 213$. Great variations can be observed between graphs across the different trials. We obtain a neuronal connectivity graph/adjacency matrix, by setting the weights (probabilities) less than 0.5, to zero. The sum of the adjacency matrices from different trials in the same time interval $W_t = W_1^t + \cdots + W_{20}^t$, are also used to determine the connectivity consistency. All consistencies greater than 5 are preserved.

A graph correlation is obtained by way of cross-correlating the duly vectorized weight matrices, e.g., the element value of $i$-th row and $j$-th column of the matrix stands for the correlation between $i$-th and $j$-th graphs’ weight vectors, and the matrix is symmetric. The red dash lines divide the plot of Figure 2.7 into small blocks, representing the exact time interval corresponding to each specific stimulus shown on the left of the plot, and Figure 2.7 provides an intuitive view of the memories (time delay) between consecutive stimuli and similarities of graphs activated by similar stimuli.

![Figure 2.7](image)

**Figure 2.7** Correlation matrix between graphs.
2.5.3.4 Improvement with 2-Simplex Information

To reduce the computational complexity and explore information gain by considering 3-way interaction relative to pairwise connection, we repartition the artificial movie into 8-time intervals corresponding to 8 different stimuli. These are also grouped into 4 groups of related stimuli based on the orientation information. According to the result we have in Figure 2.7, we can observe that the fifth stimulus’s neuronal response is highly correlated with that of the fourth stimulus, so two related stimuli, first and fifth stimuli, are not analyzed in this subsection. Further analysis focuses on the comparison of graph representations under other related stimuli ($2^{nd}-4^{th}$, $6^{th}-8^{th}$ stimuli).

We run the optimization model in Equations (3) and (5), respectively, with the same hyperparameter settings on our neuronal-activity dataset. We apply the same post-processing after optimizing the model as in the previous subsection. From Equation (3), we reach the optimal solution $W_1^t, \ldots, W_{20}^t$ and $W_{1}^{(3d)},1^t, \ldots, W_{20}^{(3d)},20^t$ for 20 trials, where $t = 1, \ldots, 8$. Then, we transform each weight matrix into an adjacency matrix and each simplex tensor into a binary tensor by setting the weights to 1 if they are greater than 0.5 and 0 otherwise. Afterward, we aggregate the adjacency matrices and the binary tensor, respectively, within the same time intervals from different trials and set the threshold of valid weight matrices to 8 and the simplex tensor threshold to 4. In other words, we only plot the graphs with edges repeating more than 8 times and 2-simplex connections repeating more than 4 times in the same time interval across the 20 trials, thereby capturing response consistency of neurons over trials. The results we get from Equation (3) are called graph representation without simplex information, and those we get from Equation (5) are called graph representation with simplex information. We use a red triangle to fill the space to represent the tight (simplex) connection among three neurons. As shown in Figure 2.6, plot (a) shows the graph representation calculated from Equation (5) under the second stimulus with red simplex information, while plot (c) shows the graph representation calculated from Equation (3) under the same stimulus without simplex information.

From the graph representations in Figure 2.8, we can observe that the pairwise connection representations with and without 2-simplex connections are similar to each other under the same or related stimuli. In contrast, the 2-simplex connections under related stimuli are intuitively different. Therefore, we vectorize the weight matrices and calculate the Pearson correlation coefficient between weight vectors under related stimuli. The result is shown in Table 2.2, and Graph $i$ stands for the graph representation under $i$-th stimulus.
Table 2.2 Comparison between related stimuli with and without simplex information.

<table>
<thead>
<tr>
<th>Related Group</th>
<th>Correlation with 2-Simplex Connection</th>
<th>Correlation without 2-Simplex Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>graph 2–graph 6</td>
<td>0.49</td>
<td>0.47</td>
</tr>
<tr>
<td>graph 3–graph 7</td>
<td>0.55</td>
<td>0.51</td>
</tr>
<tr>
<td>graph 4–graph 8</td>
<td>0.39</td>
<td>0.30</td>
</tr>
</tbody>
</table>

Based on the result, $corr(w_2, w_6) < corr(w'_2, w'_6)$, shown in Table 2.2, and the intuitive observation of the difference between 2-simplex connections under different stimuli, we come up with a hypothesis that the pairwise connection space neglecting higher-order information ($W_{2d}$), can be decomposed into a new pairwise connection space ($W'_{2d}$) and a 2-simplex connection space ($W_{3d}$). The new pairwise connection space primarily includes orientation information, and the 2-simplex
connection space contains more direction information. The hypothesis can be concluded per the following formula:

\[ W_{2d} = W'_{2d} \bigoplus W_{3d} \]  

To verify the hypothesis that 2-simplex connections may contain more directional information, and due to the limited neuronal-activity dataset for comparative test for the same stimulus, we randomly divide 20 trials into 2 groups each of 10 trials. By setting the thresholds for pairwise connection and 2-simplex connection matrix to 5 and 2, respectively, we apply the same post-process to both results with or without simplex information. In a similar way described earlier, we calculate the Pearson correlation coefficient between pairwise connectivity results with or without 2-simplex connections under the same and related stimuli. Our analysis is based on the results under related stimuli in 2 groups.

Figures 2.9 and 2.10 show all the correlations calculated in the experiments for pairwise connectivity and 2-simplex connectivity. In Figure 2.9, the 2 subplots in the first row are graph representations with and without 2-simplex connections for one stimulus in the first group; the 2 subplots in the second row are graph representations with and without 2-simplex connections for the same stimulus in the second group; the 2 subplots in the third row are graph representations with and without 2-simplex connections for the related stimulus in the first group, and the 2 subplots in the fourth row are graph representations with and without 2-simplex connections for the related stimulus in the second group. The marked numbers in Figure 2.9 represent 6 correlations for pairwise connectivity, which are: (1, 5): correlations between pairwise connection under related stimuli in the same group; (2, 6): correlations between pairwise connection under the same stimulus in different groups; (3, 4): correlations between pairwise connection under related stimuli in different groups. The marked numbers in Figure 2.10 represent 6 correlations for 2-simplex connectivity, which are: (1', 2'): correlations between 2-simplex connection under the same stimulus in different groups; (3', 4', 5', 6'): correlations between 2-simplex connection under related stimuli within each group or in the different groups. Table 2.3 shows all the correlations between graph representations without simplex information. Table 2.4 shows all the correlations between graph representations with simplex information, and Table 2.5 shows all the correlations between 2-simplex connections. We run the program 20 times and calculate the mean value of correlations.

**Table 2.3** Correlation between pairwise connections without 2-simplex connectivity.

<table>
<thead>
<tr>
<th>Related Stimuli</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>graph 2–graph 6</td>
<td>0.486</td>
<td>0.623</td>
<td>0.250</td>
<td>0.443</td>
<td>0.266</td>
<td>0.266</td>
</tr>
<tr>
<td>graph 3–graph 7</td>
<td>0.475</td>
<td>0.531</td>
<td>0.337</td>
<td>0.419</td>
<td>0.361</td>
<td>0.361</td>
</tr>
<tr>
<td>graph 4–graph 8</td>
<td>0.219</td>
<td>0.640</td>
<td>0.150</td>
<td>0.142</td>
<td>0.201</td>
<td>0.201</td>
</tr>
</tbody>
</table>
We observe that by decomposing the original pairwise connectivity information into new pairwise connectivity information and 2-simplex connectivity information, the correlations of pairwise connectivity representations between related stimuli or same stimulus in different groups (in Table 2.3) mostly increases by 10 percent compared with those between the original pairwise connectivity representations (in Table 2.2). Moreover, the first and second columns of Table 2.4 show the correlations of 2-simplex connectivity representations between the same stimulus in different groups, which is around 15 percent higher than those between related stimuli in the third to the sixth columns.
Figure 2.9 Pairwise connectivity comparison.
2.6 Discussion

Learning and analyzing structures from signals, as captured by a graph, is an interesting topic in not only neural-science but also in the computer science realm. Graph structures are instrumental in conveying the potentially complex interacting structure of signals and efficiently preserving all associated information.

**Model:** The challenge of this neuronal dataset is the low sample size as tens of data points can hardly support learning an adequate graph model, thus making static graph embedding unattainable by partitioning the data into smaller batches. With no prior knowledge about data and no interpolation, our proposed model acquires the data’s dynamics and inherent structure. We have also introduced a functional and practical means of tracking the homology of the underlying complex with the advantage of gleaning motion information as explained.

**Dataset:** From this neuron signal dataset, by way of the mathematical model, we observe variations of neuron connectivity over trials, while preserving similar patterns for similar stimuli in the V1 area, and by looking at different time scales, we also see hysteresis from one stimulus to another. These observations can be seen as an essential step for studying a brain’s functional connectivity in response to specific stimuli.

**Results:** Based on the results of the given neuronal-activity dataset, we were able to conclude that orientation information in the V1 area is adequately reflected by pairwise connectivity. In contrast,
direction information may be gleaned from a structure obtained by the connections among neurons beyond pairwise connections. This is an empirical finding based on the model awaiting experimental biological validation.

2.7 Future Work

Natural movie: While our model was tested on artificial movie data, much work remains in investigating natural movie stimuli. While we expect higher-order information structures will have a stronger presence, variability will also be a great challenge.

2.8 Conclusions

This paper introduces an optimization driven approach to learning the dynamics of graphs. Three alternative perspectives were presented to capturing the evolution of a graph in time while accounting for the dynamics generated by the nodal activities. In particular, we addressed the difficulty of the low sample rate for detecting graphs and discovered the functional connectivity due to specific stimuli instead of revealing the physical connections of neurons. Future work will include the collaborative activity of neurons and graph transformation to characterize the evolution better.
CHAPTER

3

REFINING SELF-SUPERVISED LEARNING: BEYOND LINEAR METRIC

3.1 Introduction

The notion of latent representation of data has, over the last few years, emerged as a significant catalyst, particularly in computer vision, where a visual representation in the latent space provided a powerful perspective for supervised learning and inference, such as ImageNet classification. In addition to achieving a gain in accuracy performance, such learned representations may be more richly interpreted as an embedding map to project data in the original space onto an intrinsic and distinct latent space whose model (learned weights) may, in turn, be useful for initialization of other modeling tasks with more limited data. However, scaling this approach to large datasets (e.g., ImageNet) is labor-intensive and time-consuming and calls for a more acceptable alternative.

As a proposed solution, Self-Supervised Learning (SSL) seeks to acquire labels from different modalities of a fraction of the totality of a given data sample. SSL has been quite successfully adopted in natural language processing (NLP) [Mik13], where for a given sentence, the model predicted the missing word. The capacity of this model to learn and effectively exploit the representation of the original data by merely observing and predicting interactions of subsets of the complete data is remarkable. In addition to the markedly reduced requirement on the size of labeled/explained words for training, this led to the clever enrichment of each data sample (a sentence) to a multi-modal potential by merely varying the word excision. It is fair to impute the inspiration to SSL (used in several unsupervised learning approaches [Che20b; He20] of visual representations) and the related recent research growth in narrowing the gap between classification accuracy of unsupervised
Figure 3.1 An overview of the self-supervised learning model. Contrastive loss is calculated in the hidden space. Encoders’ performance are evaluated in the latent space.

and supervised learning. A key feature of SSL is the adopted contrastive loss, which is aimed at discriminating positive keys (i.e., samples) from negative ones in the latent space of images. Note that the positive keys for a given sample constitute different modalities (i.e., duly selected transformation and augmentation methods as afore-noted) of the original images. In contrast, the negative keys coincide with those that do not. The contrastive loss is thus aimed at respectively minimizing the distance between positive pairs and maximizing that between negative pairs in the latent space. Cleverly selecting a metric is in order as the invoked transformation and augmentation procedures may significantly impact the training process. The larger number of negative keys than the number of positive keys may be of consequence. With the overall model flow shown in Fig.3.1, prior research [Che20b] has shown that evaluating the similarity measure in a projected hidden space can significantly improve the performance of an encoder sought to capture meaningful features in the latent space.

The most commonly adopted approach to quantifying a similarity between object pairs is to project latent features onto a flat hidden space followed by a cosine similarity between them. Underlying this, which is a strong assumption, these features may be linearly distinguished in the latent space. These are, in fact, the result of non-linear processing across the learning layers and are hence elements point elements on a manifold rather than a linear space. The discrepancy between the computed linear distance and the actual geodesic between the features points to a need for its accounting to potentially improve SSL. To that end, we propose a way of accounting for the intrinsic non-linearity by using the Jaccard similarity coefficient, developed in the early 1900's [Jac12], to include non-linear characteristics in the similarity between finite sets. Upon interpreting SSL as an inference problem and using a hypothesis test for a quantitative decision in contrastive learning, we propose a bi-projector Jaccard-based loss which effectively captures the non-linearities in the hidden spaces.

Our proposed adaptive approach effectively aims at addressing the inherent modal structure of the latent/pseudo-invariant characteristics of the data (using Jaccard-metric) with a validation using two well-known self-supervised learning models SimCLR [Che20b] and MoCo [He20], using both the k-nearest neighbors (k-NN) classification and the standard linear classification method. The selected datasets are CIFAR-10, CIFAR-100, and Tiny-ImageNet-200 to capture the class variability bandwidth used in evaluating the performance of SSL. Our contributions in this paper include (1) Our providing a theoretically sound and tractable hypothesis testing formulation of the contrastive
loss and its original triplet loss [Sch15], as well as a clear understanding of the temperature parameter τ and of the ensuing projection for stabilization; (2) Our introduction of a bi-projector model along with a novel Jaccard-similarity-based contrastive loss evaluation; (3) Our derivation of similarity maximization/minimization of positive/negative pairs in each of the hidden spaces separately accounting for the non-linear features by a Jaccard metric-based fusion.

The balance of the paper first proceeds with an overview of related works. Section 3 discusses the proposed method followed by the experimental substantiation in Section 4 and concluding remarks in Section 5.

### 3.2 Related Works

**Self-Supervised Learning:** A contrastive learning model ultimately yielding a semantic space in a Computer Vision (CV) setting was proposed in [Cho05]. In so doing, they proposed a $L_2$-norm distance to estimate similarity among two images. This procedure entailed minimizing a semantic distance between images from the same class while maximizing that from different classes. Self-supervised learning studies were also widely applied to natural language processing [Mik13], where the model was to predict a missing word from a given input sentence. The use of contrastive learning has recently gained popularity in CV in conjunction with learned visual representations [Che20b; He20] as well as pretext tasks [Doe15; Jen18; Kim18].

Comparing all pairs of images from one or different classes being impractical and time-consuming has led to more recent approaches carrying out the positive-negative comparison among a small random subset of the whole dataset. As a result, a transformation/augmentation of some original images yields positive keys, while the others are treated as negative pairs. Empirically, choosing the transformation and augmentation methods [Che20c] and handling negative samples [Chu20] may also induce biases and hence significantly affect the learning encoders' performance.

In addition, the recent focus in research has gradually shifted to only minimizing the distance between the positive pairs without simultaneously maximizing the distance between negative pairs [Che21; Gri20; Tia21]. Given different views of the original images, the encoder is trained by minimizing the distance between different views in the hidden space. An additional predictor [Gri20] was introduced in an online network to ensure an asymmetric system, which helped prevent collapse during training. Bootstrap Your Own Latent (BYOL) model was theoretically analyzed in [Tia21] together with the dynamics of the model without contrastive pairs, highlighting the essence of the asymmetric system unfolding the appropriate transformation for an invariant representation, and an exponential moving average (EMA) for updating the two encoders (the intuitive geometric interpretation is that an inverse map is learned to ensure an invariant representation of the data). This resulted in showing that this approach achieved a similar performance level than that without contrasting negative pairs, thus yielding significant savings by the reduction of negative keys.

**Current works on τ, loss, and similarity measurement:** As shown in [Che20b; Che20a], the temperature is a scaling parameter that has a significant influence on the performance of learning latent...
representations and is key to the discrepancy arising between the conventional contrastive loss and standard cross-entropy loss. More specifically, [Wan20] pointed out the temperature-controlled penalties’ strength of hard negative keys, causing an imbalance. Decreasing $\tau$ causes the hard negative samples, which are similar to the positive key, to increase the impact on the loss, while extreme small $\tau$ may lead to poor performance due to the concentration on the extreme narrowed negative samples and false-negative samples (different images from the same class). Recent research [Wan20; Che20a] unveiled two components to the original contrastive loss function: the first one quantifying the alignment loss, and the second reflecting the uniformity of the representation in the latent space. The similarity measurement between keys being captured by the cosine similarity[Sin01] and $L_2$-norm, thus accounts for the former.

3.3 Method

3.3.1 Notations

Throughout the paper, we will adopt a mathematical upper case letter to denote a space and an upper case letter to denote the data representations in different spaces, for instance, $X$ represents the original data in the pixel space $\mathcal{X}$, $X^K$ the transformation and augmentation of the original image data in the pixel space, and $Q$ and $K$ respectively stand for the features in the latent representation spaces of the original and augmented images. We use the lower case letter to denote the elements in different spaces, i.e., $x_i$ indicates the $i^{th}$ object element of $X$, the original dataset in the pixel space. We consider $f_\theta(\cdot)$ to be a function parameterized by $\theta$.

3.3.2 Contrastive Loss

Given an unlabeled dataset $X = \{x_1, x_2, \ldots, x_N\}$, a similarity calculation method $s(\cdot, \cdot)$, an embedding functions $f_\theta(\cdot)$, and a projector $g_\theta_p(\cdot)$, which collaboratively projects the data from the original space $\mathcal{X}$ into a flat hidden space $\mathcal{H}$. Define $x_i$ a data sample in $X$, and $x^+$ and $x^-$ stands for a positive key and negative keys of $x_i$, respectively, $h_{q_i} = g_\theta_p \circ f_\theta(x_i)$, $h_{k^+} = f_\theta \circ f_\theta(x^+)$, and $h_{k^-} = g_\theta_p \circ f_\theta(x^-)$. To carry through the original idea of contrasting positive with negative keys, a triplet loss, was first proposed in [Had06], and is aimed at which learning an invariant mapping to project high dimensional features onto a low dimensional space, such that intra-class features are close to each other while keeping inter-class features far apart,

$$
\mathcal{L}_{triplet}(x, x^+, x^-|\theta, \theta_p) = \max(0, d(h_q, h_{k^+}) - d(h_q, h_{k^-}) + m),
$$

(3.1)

with $d(\cdot, \cdot)$ denoting a metric, and $m$ is some margin that restricts the largest meaningful distance difference between intra-class and inter-class latent representations. A probabilistic interpretation
has recently emerged "softmax" expression to highlight the frequential aspect of sampling,

\[ \mathcal{L}_{co}(x_i, x^+, x^-) = -\log \left( \frac{\exp \left( \frac{\text{sim}(h_q, k^+)/\tau}{\text{max}(Y)} \right)}{\exp \left( \frac{\text{sim}(h_q, h_{k^+})/\tau}{\text{max}(Y)} + \sum_j \exp \left( \frac{\text{sim}(h_q, h_{k^-})/\tau}{\text{max}(Y)} \right) \right)} \right), \]  

(3.2)

where \( \tau \) is the temperature scalar in the contrastive loss. One may note the presence of the cosine similarity expression \( \text{Cos-Sim}(x, y) = \langle x, y \rangle / \|x\| \|y\| \), as a similarity measure of choice between latent representations. The encoder of queries and keys can be identical [Che20b] or updated with different rules based on momentum rules [He20]. While seemingly distinct, these two losses bear a relation as follows,

**Theorem 3.** The contrastive loss is a smooth approximation, and an upper bound to the mean of triplet loss in the condition of zero margin, \( m = 0 \).

**Proof:** The LogSumExp term in the denominator can be interpreted as the smooth approximation to the maximum function. Consider a finite set \( Y = \{y_1, ..., y_n\} \),

\[ \text{max}(Y) = \log \left( \sum_i e^{y_i} \right) \leq \log \left( n e^{\text{max}(Y)} \right) = \log n + \text{max}(Y). \]  

(3.3)

While the so-called soft-max is well known in ML for avoiding non-smoothness in calculating the gradient, it often is implicit in other settings. The previous equation yields the conventional contrastive loss without the margin \( m \). By defining \( s^+ = \text{Cos-Sim}(h_q, h_{k^+})/\tau \) and \( s^- = \text{Cos-Sim}(h_q, h_{k^-})/\tau \), we can rewrite,

\[ \mathcal{L}_{co}(h_q, h_{k^+}, h_{k^-}) = -\log \left( \frac{e^{s^+}}{e^{s^+} + \sum_i e^{s^-}} \right) = -\log \left( e^{s^+} \right) + \log \left( e^{s^+} + \sum_i e^{s^-} \right) = -s^+ + \log \left( e^{s^+} (e^0 + \sum_i e^{s^-}) \right) = \log \left( e^{s^+} (e^0 + \sum_i e^{s^+ - s^-}) \right) \]

\[ \geq \text{max} \left( 0, s_1 - s^+, ..., s_n - s^+ \right) \geq \frac{1}{n} \sum_i \mathcal{L}_{\text{triplet}}(h_q, h_{k^+}, h_{k^-}). \]  

(3.4)

The minimization of the contrastive loss function implies a smooth enhancement of proximity of positive pairs and of separation of negative pairs. One can observe that when \( \tau = 1 \), the loss function coincides with the cross-entropy loss of one positive pair and \( n \) negative pairs, thus enabling \( \tau \) to modulate the sharpening/flattening of the similarity contrast between positive and negative pairs.

### 3.3.3 A Variation on a Theme: A Jaccard-index Assessment

#### 3.3.3.1 Jaccard Similarity

The Jaccard index [Jac12], commonly called the Jaccard similarity coefficient, is a set-theoretic measure to generically evaluate similarity between two finite sample sets. In general, the Jaccard
similarity coefficient is evaluated as a ratio of the intersection of two finite sets $A$ and $B$ and their union,

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A \setminus B| + |B \setminus A| + |A \cap B|}.$$  \hspace{1cm} (3.5)

Note that the cardinality of a finite set is non-negative; if $A$ and $B$ are both empty sets, the coefficient is 1, otherwise $0 \leq J(A, B) \leq 1$. An alternative interpretation of the Jaccard index is based on the measure. An information theoretic/probabilistic interpretation unveils a measure-theoretic hence deeper comparison using a probability measure $\mu$ on the measurable sample space $S$, yielding a similar coefficient as,

$$J_\mu(A, B) = \frac{\mu(A \cap B)}{\mu(A \cup B)}. \hspace{1cm} (3.6)$$

In light of the use of measures (i.e., all moments), this new measure particularly points to the significance of the information scoped out and beyond the linear (i.e., second-order moment) typically invoked by a Cosine similarity. This is hence consistent with our goal of gleaning additional secondary information beyond that normally sought in the latent space of the classes.

**Proposition 4.** The Jaccard index defines a sharper similarity measure than that defined by the Cosine-Similarity.

**Proof:** If $F$ is a measurable finite information space, which includes all extracted information from the finite dataset $X = \{x_i\}_{i=1,...,N}$ by the encoder $f_\theta(\cdot)$, $F = \bigcup_{i=1}^{N} f_\theta(x_i)$ with a measure $\mu$, where $\mu(F) = 1$. The cosine similarity is to only measure the intersection between the finite informational representations of two samples, where $\text{Cos-Sim}(x_i, x_j | \theta, \mu) = \mu(f_\theta(x_i) \cap f_\theta(x_j))$. By comparison, Jaccard index is a measurement considering the intersection as well as the measure of two finite sets, where $\text{Jac-Sim}(x_i, x_j | \theta, \mu) = \frac{\mu(f_\theta(x_i) \cap f_\theta(x_j))}{\mu(f_\theta(x_i) \cap f_\theta(x_j)) + \mu(f_\theta(x_i) \setminus f_\theta(x_j)) + \mu(f_\theta(x_j) \setminus f_\theta(x_i))}$. The cosine similarity measures the non-shared informational elements between two sets as $1 - \mu(f_\theta(x_i) \cap f_\theta(x_j))$, which does not precisely describe how dissimilar two sets are, since $f_\theta(x_i) \cap f_\theta(x_j) \neq F$. In other words, the complete informational space is not properly defined, and the similarity is not normalized by the measure of the complete space. \hspace{1cm} $\blacksquare$

### 3.3.3.2 Inferential Perspective of Contrastive Learning

A hypothesis is a proposed explanation of an observed phenomenon. As a statistical inference strategy, statistical hypothesis testing requires a probabilistic model to describe the underlying distribution of each hypothesis reflected by the observed data of the problem at hand[Leh06].

Self-supervised contrastive learning, viewed as a guessing phenomenon, is the result of random data sample assessment and assignment to either the positive or negative class, each with an under-
lying distribution, hence justifying the proposed approach. As a result, a null hypothesis $H_0$ stating that two observations (i.e., images) are from the same class, with a corresponding distributional score $f_{\theta^+}({\cdot},{\cdot})$. Similarly, an alternative hypothesis $H_1$ is defined for two observations from different classes, with a corresponding score $f_{\theta^-}({\cdot},{\cdot})$.

### 3.3.3.3 Proposed Contrastive Learning Model

With the inferential perspective in mind and the Jaccard similarity measure in hand, we propose to seek hidden representations as finite information sets/distributions of original objects. In congruence with the probabilistic structures of probability measures discussed earlier, intersections will account for similarity, while set differences reflects dissimilarities (or disparities) among the data samples. As the explicit distributions are not readily available, we proceed to learn the information so long as the metrics we use account for the prevailing embedding space. To that, we introduce a bi-projector system model $g_{\theta^i}({\cdot})$, $i = 1, 2$, with the first following $H_0$ (i.e., that two objects are from the same class by measuring the intersection of extracted features), while the other following $H_1$. We can hence define two hypotheses scoring functions,

$$
\begin{align*}
  f(x_1, x_2|H_0, \theta) &:= f_{\theta^+}({\cdot},{\cdot}) = s \circ g_{\theta^1} \circ f_{\theta}, \text{ and} \\
  f(x_1, x_2|H_1, \theta) &:= f_{\theta^-}({\cdot},{\cdot}) = d s \circ g_{\theta^2} \circ f_{\theta}, 
\end{align*}
$$

(3.7)

respectively, where $s({\cdot},{\cdot}), d s({\cdot},{\cdot})$ is each respectively a similarity and dissimilarity measure.

Fig. 3.2 provides a high-level and intuitive description of the proposed process model. The latent representations of the original data lying on an unknown curved manifold, make it unreasonable
Figure 3.3 An overview of our Jaccard-based self-supervised learning model. One additional projector is added to give a different aspect of extracting features. Two hidden spaces are fused based on Jaccard similarity.

and possibly detrimental to the training/learning, to be using Euclidean-based similarity metrics. The key idea is to then learn a homeomorphism $g_\theta(\cdot)$ that projects the latent space information onto a set of Euclidean spaces, where similarity can be measured by $L^2$-norm. In our case, we hence seek, as illustrated in Fig. 3.3, two distinct projectors which directly contribute to the computation of the Jaccard similarity, which is in turn used to adapt the encoder. A carefully designed combination of the information extracted by the two projectors (i.e., similarity and dissimilarity) as described next, will yield a Jaccard similarity.

**Algorithmic Development:** This Jaccard-based similarity/loss provides a non-linear combination (fusion) between features and involves asymmetry in the whole system, which prevents the two projectors from collapsing in one single point. To proceed with the detailed development of the Jaccard index-based model, we first note that the similarity/dissimilarity alluded to in Eq. 3.7, are Euclidean-based (i.e., $ds(x, y) = \|x - y\|^2$), where $x, y$ are normalized in the hidden space. Specifically, for given latent representations of two images $q_1$ and $q_2$, we only consider the two finite information set extracted by the first projector $g_\theta(\cdot)$ for measuring the intersection, $(g_\theta(q_1), g_\theta(q_2))$, while the second projector is for measuring the dissimilarity, $\|g_\theta(q_1) - g_\theta(q_2)\|_2^2$, when ultimately computing the Jaccard similarity.

**Claim 1.** The application of Jaccard index to measure the similarity as the fusion between two hidden spaces, is equivalent to respectively maximizing and minimizing Jaccard similarity between positive and negative pairs:

$$L_{J-Tri}(x, x^+, x^-, \theta, \theta_1, \theta_2) = - \frac{|x \cap x^+|}{|x \cup x^+|} + \frac{|x \cap x^-|}{|x \cup x^-|}_{\theta, \theta_1, \theta_2}$$

$$= - \frac{\langle g_\theta(\cdot) \circ f_\theta(x), g_\theta(\cdot) \circ f_\theta(x^+) \rangle}{\| g_\theta(\cdot) \circ f_\theta(x), g_\theta(\cdot) \circ f_\theta(x^+) \|_2} + \frac{\| g_\theta(\cdot) \circ f_\theta(x) - g_\theta(\cdot) \circ f_\theta(x^-) \|_2}{\langle g_\theta(\cdot) \circ f_\theta(x), g_\theta(\cdot) \circ f_\theta(x^-) \rangle}$$

$$+ \frac{\| g_\theta(\cdot) \circ f_\theta(x), g_\theta(\cdot) \circ f_\theta(x^-) \|_2}{\langle g_\theta(\cdot) \circ f_\theta(x), g_\theta(\cdot) \circ f_\theta(x^-) \rangle}.$$

(3.8)
we follow conventional self-supervised learning and calculate the contrastive loss among positive and negative keys based on hidden representations in each hidden space, respectively. Subsequently, applying Jaccard similarity to achieve feature fusion between two hidden spaces (as noted with sensitivity to similarity and dissimilarity, respectively) yields the proposed model detailed next.

**Proposition 5.** The overall loss includes conventional contrastive loss of each projectors and a Jaccard-based fusion across the two projectors.

\[
\mathcal{L}_J(x, x^+, x^-; \theta, \theta_1, \theta_2) = -\alpha_1 \log \left( \frac{e^{(h_{q_1}^{(1)}, h_{k_1}^{(1)})/\tau}}{e^{(h_{q_1}^{(1)}, h_{k_1}^{(1)})/\tau} + \sum_i e^{(h_{q_1}^{(1)}, h_{k_1}^{(1)})/\tau}} \right) - \alpha_2 \log \left( \frac{e^{(h_{q_2}^{(2)}, h_{k_2}^{(2)})/\tau}}{e^{(h_{q_2}^{(2)}, h_{k_2}^{(2)})/\tau} + \sum_i e^{(h_{q_2}^{(2)}, h_{k_2}^{(2)})/\tau}} \right) \\
- (1 - \alpha_1 - \alpha_2) \log \left( \frac{e^{(h_{q_1}^{(1)}, h_{k_1}^{(1)})/\tau}}{e^{(h_{q_1}^{(1)}, h_{k_1}^{(1)})/\tau} + \sum_i e^{(h_{q_1}^{(1)}, h_{k_1}^{(1)})/\tau}} \right),
\]

where \( h_{q_i}^{(j)} = g_{\theta_j} \circ f_\theta(x), h_{k_i}^{(j)} = g_{\theta_j} \circ f_\theta(x^+), \) and \( h_{k_i}^{(j)} = g_{\theta_j} \circ f_\theta(x^-), \) \( j = 1, 2. \)

In summary, the algorithm of the proposed model in Fig. 3.3 is (1) the original image, its augmentation view, and the negative keys are encoded into latent space; (2) Projector 1 and 2 respectively extract hidden representations and calculate the loss presented in Eq.(10); (3) The encoder and two projectors are directly updated based on stochastic gradient descent (SGD).

### 3.4 Experimental Results

In this section, we demonstrate the applicability, efficiency, and accuracy of the Jaccard-similarity-based model on contrastive representation learning. In all experiments, we first train the encoders...
and projectors with unsupervised learning, and then we evaluate the encoders’ performance (classification accuracy) with k-NN classification and a supervised linear classifier. We consider SimCLR[Che20b] and MoCo[He20] as the benchmarks in our experiments. Our model is built based on the same setup as conventional models, and we compare the performance via the same evaluation method. We test the performance of our model on CIFAR-10/100[Kri09], which is composed of 50,000 training and 10,000 testing images, and Tiny-imagenet-200[Le15], which consists of 100,000 training and 10,000 testing images for 200 classes.

3.4.1 Experiment Setup

In the SSL training stage, we use ResNet-50[He16] as the backbone network for all models. All neural networks are trained with a batch size of 512 for 500 epochs on both CIFAR-10 and CIFAR-100 datasets and a batch size of 256 on the Tiny-Imagenet dataset. All models apply the same set of augmentation methods as SimCLR, and a constant learning rate $10^{-3}$. We fix the encoder after each training epoch and apply the k-NN classification method to verify the encoder’s performance during the training. Different temperature parameters ($\tau$) are chosen for training, and we choose the best encoder’s performance as the model’s final result. We have reproduced the results of the two benchmarks. Different temperature hyperparameters are chosen for training, and we choose the best encoder’s performance as the model’s final result. In the Jaccard index, the measure of a finite set is non-negative, but the range of cosine similarity is $[-1, 1]$, so $J(h_1, h_2) = 1 + \frac{\langle h_1, h_2 \rangle}{\|h_1\|_2 \|h_2\|_2}$. All the initialization of the encoder is the same by setting the same seed for generating random numbers. More details of experiment setups are shown in Appendix A.

3.4.2 k-NN and Linear Evaluation

We validate the unsupervised pre-trained models with k-NN and the supervised linear evaluation methods. Specifically, we firstly train the encoders and two projectors simultaneously in the self-supervised training procedure. In the testing, the trained encoders extract latent features of both training and testing sets to classify the representations with (1) the k-NN method: the labels of testing images are assigned based on k nearest latent representations of training images; (2) a single-layer neural network: the labeled training dataset is applied for training the linear classifier. Note that the final goal is to evaluate the encoders’ performance, we do not change or involve two projectors in the linear evaluation step.

3.4.3 Evaluation Results

In this section, we demonstrate the applicability, efficiency, and accuracy of the Jaccard-similarity-based model on contrastive representation learning. In all experiments, we first train the encoders and projectors with unsupervised learning, and then we evaluate the encoders’ performance (classification accuracy) with k-NN classification and a supervised linear classifier. We consider SimCLR[Che20b] and MoCo[He20] as the benchmarks in our experiments. Our model is built based on
the same setup as conventional models, and we compare the performance via the same evaluation method. We test the performance of our model on CIFAR-10/100[Kri09], which is composed of 50,000 training and 10,000 testing images, and Tiny-imagenet-200[Le15], which consists of 100,000 training and 10,000 testing images for 200 classes.

3.4.4 Experiment Setup

In the SSL training stage, we use ResNet-50[He16] as the backbone network for all models. All neural networks are trained with a batch size of 512 for 500 epochs on both CIFAR-10 and CIFAR-100 datasets and a batch size of 256 on the Tiny-Imagenet dataset. All models apply the same set of augmentation methods as SimCLR, and a constant learning rate $10^{-3}$. We fix the encoder after each training epoch and apply the k-NN classification method to verify the encoder’s performance during the training. Different temperature parameters ($\tau$) are chosen for training, and we choose the best encoder’s performance as the model’s final result. We have reproduced the results of the two benchmarks. Different temperature hyperparameters are chosen for training, and we choose the best encoder’s performance as the model’s final result. In the Jaccard index, the measure of a finite set is non-negative, but the range of cosine similarity is $[-1, 1]$, so $J(h_1, h_2) = 1 + \frac{\langle h_1, h_2 \rangle}{\|h_1\|_2 \|h_2\|_2}$. All the initialization of the encoder is the same by setting the same seed for generating random numbers. More details of experiment setups are shown in Appendix A.

3.4.5 k-NN and Linear Evaluation

We validate the unsupervised pre-trained models with k-NN and the supervised linear evaluation methods. Specifically, we firstly train the encoders and two projectors simultaneously in the self-supervised training procedure. In the testing, the trained encoders extract latent features of both training and testing sets to classify the representations with (1) the k-NN method: the labels of testing images are assigned based on k nearest latent representations of training images; (2) a single-layer neural network: the labeled training dataset is applied for training the linear classifier. Note that the final goal is to evaluate the encoders’ performance, we do not change or involve two projectors in the linear evaluation step.

3.4.6 Evaluation Results

In this section, we demonstrate the applicability, efficiency, and accuracy of the Jaccard-similarity-based model on contrastive representation learning. In all experiments, we first train the encoders and projectors with unsupervised learning, and then we evaluate the encoders’ performance (classification accuracy) with k-NN classification and a supervised linear classifier. We consider SimCLR[Che20b] and MoCo[He20] as the benchmarks in our experiments. Our model is built based on the same setup as conventional models, and we compare the performance via the same evaluation method. We test the performance of our model on CIFAR-10/100[Kri09], which is composed of
50,000 training and 10,000 testing images, and Tiny-imagenet-200[Le15], which consists of 100,000 training and 10,000 testing images for 200 classes.

3.4.7 Experiment Setup

In the SSL training stage, we use ResNet-50[He16] as the backbone network for all models. All neural networks are trained with a batch size of 512 for 500 epochs on both CIFAR-10 and CIFAR-100 datasets and a batch size of 256 on the Tiny-Imagenet dataset. All models apply the same set of augmentation methods as SimCLR, and a constant learning rate $10^{-3}$. We fix the encoder after each training epoch and apply the k-NN classification method to verify the encoder’s performance during the training. Different temperature parameters ($\tau$) are chosen for training, and we choose the best encoder’s performance as the model’s final result. We have reproduced the results of the two benchmarks. Different temperature hyperparameters are chosen for training, and we choose the best encoder’s performance as the model’s final result. In the Jaccard index, the measure of a finite set is non-negative, but the range of cosine similarity is $[-1, 1]$, so $J(h_1, h_2) = 1 + \frac{\langle h_1, h_2 \rangle}{\|h_1\|_2 \|h_2\|_2}$. All the initialization of the encoder is the same by setting the same seed for generating random numbers. More details of experiment setups are shown in Appendix A.

3.4.8 k-NN and Linear Evaluation

We validate the unsupervised pre-trained models with k-NN and the supervised linear evaluation methods. Specifically, we firstly train the encoders and two projectors simultaneously in the self-supervised training procedure. In the testing, the trained encoders extract latent features of both training and testing sets to classify the representations with (1) the k-NN method: the labels of testing images are assigned based on k nearest latent representations of training images; (2) a single-layer neural network: the labeled training dataset is applied for training the linear classifier. Note that the final goal is to evaluate the encoders’ performance, we do not change or involve two projectors in the linear evaluation step.

3.4.9 Evaluation Results

3.4.9.1 Experiments on $\tau$

The temperature parameter $\tau$ as a key factor of measuring similarity, it is highly related to the balancing between intra-class and inter-class distances. The higher the temperature is, the tighter structure is required in the same class. By varying the $\tau$ as a constant number in the training process, table 3.1 shows the linear evaluation results of SimCLR and our model on CIFAR-10 and CIFAR-100, respectively. We can observe that the trained encoders’ performance is highly affected by the dataset and the selected temperature parameter. In CIFAR-10 dataset, the encoder reaches the best performance with 0.3, while in CIFAR-100 dataset, 0.1 is the optimal value in table 3.1 with 68%, and 0.5 leads to 57%.
<table>
<thead>
<tr>
<th>Temperature (τ)</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>88.4</td>
<td>57.4</td>
<td>90.2</td>
<td>55.3</td>
</tr>
<tr>
<td>0.3</td>
<td>91.2</td>
<td>67.2</td>
<td>91.3</td>
<td>68.4</td>
</tr>
<tr>
<td>0.2</td>
<td>89.1</td>
<td>67.4</td>
<td>91.8</td>
<td>69.3</td>
</tr>
<tr>
<td>0.1</td>
<td>86.4</td>
<td>68.3</td>
<td>84.8</td>
<td>69.5</td>
</tr>
<tr>
<td>0.05</td>
<td>83.2</td>
<td>62.6</td>
<td>78.2</td>
<td>60.3</td>
</tr>
</tbody>
</table>

### 3.4.9.2 Experiments on our model

We show in Table 3.2 the complexity, i.e., the running time, of our model compared with SimCLR. The training time of one epoch is recorded on one NVIDIA Tesla V100 GPU for all models. Only one additional projector, a 2-layer fully connected neural network, is added to our model, and the loss is computed in matrix format. The recording time shows our model takes 1.1 times of SimCLR running time in CIFAR datasets and 1.01 times of it in the Tiny-Imagenet-200 dataset. Fig.3.4 shows the encoders’ performance curves of SimCLR and our model based on the k-NN method w.r.t the training epoch. We can observe that our model's encoder demonstrates quite a convincing learning speed compared with the conventional model with the same learning rate setup.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CIFAR</th>
<th>Tiny-Imagenet-200</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimCLR</td>
<td>1m36s/epoch</td>
<td>13m51s/epoch</td>
</tr>
<tr>
<td>Our model</td>
<td>1m47s/epoch</td>
<td>14m01s/epoch</td>
</tr>
</tbody>
</table>

Table 3.3 shows the linear evaluation results’ comparison on different datasets between conventional self-supervised learning models and our modified models. Note that compared with the conventional contrastive learning model, the complexity of the whole system is rarely increased, and the main difference is based on the similarity metric. The results show that our proposal of considering a non-linear metric helps improve the accuracy in most comparable experiments. The more complex the data is, the more improvement is achieved, confirming that the non-linear impart is more pronounced. These observations reflect the fact that latent representations of a complex dataset lie on non-linear manifold, where the linear metric may lead to imprecise estimation. Overall, the encoder based on SimCLR setup with our proposed model obtains the best performance in all datasets. We also adapt our model to MoCo's setup. The k-NN classification accuracy curves in the training process are shown in Appendix B.1.
Figure 3.4 An overview of our Jaccard-based self-supervised learning model. One additional projector is added to give a different aspect of extracting features. Two hidden spaces are fused based on Jaccard similarity.
Table 3.3 Linear evaluation results on different datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Top-1 Accuracy (%)</th>
<th>Top-1 Accuracy (%)</th>
<th>Top-1 Accuracy (%)</th>
<th>Top-1 Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SimCLR</td>
<td>SimCLR +</td>
<td>MoCo</td>
<td>MoCo +</td>
</tr>
<tr>
<td></td>
<td>k-NN   linear</td>
<td>our model</td>
<td>k-NN linear</td>
<td>our model</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>88.6  91.5</td>
<td>88.6  91.8</td>
<td>85.46  90.5</td>
<td>85.1  90.2</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>62.1  68.6</td>
<td>63.9  69.5</td>
<td>56.8   66.8</td>
<td>59.2  67.5</td>
</tr>
<tr>
<td>Tiny-imagenet</td>
<td>35.9  50.7</td>
<td>39.5  53.8</td>
<td>33.5   49.6</td>
<td>35.4  51.2</td>
</tr>
</tbody>
</table>

3.4.10 Learning performance of different projectors

To check how the bi-projector system helps improve the encoders’ performance, we select specific samples, shown in Fig.4.3, from the CIFAR-100 training set, and record how their latent representations’ similarities change during training. Shark and Whale are two similar classes, where pictures in these two categories are easier to be confused with each other. Two shark images are chosen to be positive pairs. In contrast, one whale image, which has a similar background and shape, is treated as a similar negative key, and the other is a dissimilar negative key.

Fig.3.6 and 3.7 show the change curves of cosine similarities in the latent spaces and hidden spaces in terms of training time, respectively. The blue curve represents the similarity between the positive pair, the orange curve reflects the similarity between the similar negative pair, and the green curve shows the similarity between the dissimilar negative pair. The figures demonstrate that the dissimilar negative pairs can be easily distinguished in all latent and hidden spaces. Fig.3.6 shows that our model can better distinguish the positive key from the similar negative key. Moreover, the trend of the blue curve is not decreasing, and it remains at a higher value. Fig.3.7 shows the difference between the single-projector and bi-projector systems. In the conventional SSL model, the extracted features between the similar negative pair become indistinguishable. However, in our model, one projector can be a complement to the other, where even Projector 2 does not perform well, Projector 1 can prevent the model from becoming confused with a similar negative pair. More examples of projectors’ learning process based on different initialization are presented in Appendix B.2.

3.4.11 Non-linearity preservation

Conventional contrastive loss only considers the linear similarity between hidden representations, while our model contains cosine similarity for each hidden space and introduces a Jaccard-based fusion, which provides a non-linear combination in the system. We use the Gram-Schmidt process[Leo13] to remove linear features and check the similarity between second-order features. To be specific, given two latent features $q_1$ and $q_2$, $\hat{q}_1 = q_1 - \frac{(q_1 \cdot q_2)}{(q_2 \cdot q_2)} q_2$ and $\hat{q}_2 = q_2 \odot q_2$, where $\odot$ is the element-wise multiplication operator. Fig.3.8 shows the cosine similarity curve of the positive pair
Figure 3.5 To see how the encoder learns, two shark images are selected to be original object and positive key in the training set, while two whale images, a similar class to shark, are chosen to be negative keys, where one is similar to original object, and the other is less similar to that.
Figure 3.6 Cosine similarity is calculated between positive keys, similar negative keys, and less similar negative keys in the latent space. Our model shows a better distance measuring between positive pairs and similar negative pairs.

Figure 3.7 Cosine similarity is calculated between positive keys, similar negative keys, and less similar negative keys in the hidden spaces. In our model, one projector can be seen as a complement to the other, so overall they will give better distance measuring between positive pairs and similar negative pairs.
Figure 3.8 Removing the linear feature between latent representations of the original object, positive key, and similar negative key. Conventional SSL model doesn't retain additional information, while our model preserves a higher-moment similarity.

and similar negative pair after removing the linearity. The right plot shows that non-linearity is also preserved in the extracted features, while curves are overlapped in the left figure. We also show the non-linearity between samples of whales in Appendix B.3.

3.5 Conclusions

This paper explains the contrastive loss from a hypothesis-testing point of view by distinguishing whether a pair of images is from the same category. We subsequently introduce a bi-projector model and a Jaccard-similarity-based loss to fuse the information extracted by the two projectors. In the experiments, we demonstrate the compatibility of our model by applying it as a plug-in unit to the state-of-art approaches. We also show that our model outperforms state-of-art contrastive SSL methods. The accuracy improvement is more pronounced for more complex datasets proving that the similarity measure should account for non-linearity on the latent space. Moreover, we use some images as samples to present more training details in our model, which can capture non-linearity compared with conventional models. More future research should focus on defining each projector specifically, which will guide our further understanding of the feature representations’ distribution in both hidden spaces and latent space and help explain what the machine learns in the training procedure.
CHAPTER 4

IMPLICIT BAYES ADAPTATION: A COLLABORATIVE TRANSPORT APPROACH

4.1 Introduction

Much of the great success of Deep Learning is owed to the massive amounts of labeled data whose considerable effort of collection and labeling is often overlooked. Its promise and viability largely ride on the flexibility of safeguarding the learning on a labeled data set to adapt to unlabeled data with possibly a different distribution. With a possibly significant semantic gap between the first referred to as "source data", and the second, the "target data" (due to different sensing or other), an equivalence mapping is generally sought to not only exploit the extensive initial learning, but to also associate effective labeling whose space is assumed coincident with that of the source space.

Unsupervised domain adaptation essentially entails a minimization of the distance between the source and target domains using different criteria to reduce the distance, such as Maximum Mean Discrepancy (MMD)[Lon15], Correlation Alignment[Sun16], and Kullback-Leibler (KL) divergence[Zhu15]. An adversarial learning approach was proposed in[Tze17] to synthesize an adapted equalizing mapping of the target domain to the source data, thereby facilitating the sought domain adaptation. Given its flexible amenability to assessing imbalance between distributions, OT has also been central to reducing the distance between complex distributions[Dam18; Xu20; Fat21; Red17; Pat15]. In [Red20]a search for two simultaneous maps is based on initiating a collaborative OT learning using a feature-sample based tensor between distributions, while an exploitation of different clustering parameters of the two distributions [EBB21] serves to carry out a collaborative OT. Accounting for the complete statistical characteristics of both domains has remained absent,
and using coarser domain knowledge together with a pair-wise association has fallen short in OT-based performance in domain adaptation. Similar samples, e.g., digit 1 and digit 7, will invariably result in gross mismatching in the optimal transport plan.

In computing the transport cost in practice, samples are often considered as a finite set of vectors in a high-dimensional Euclidean space. In many applications, these data points, also subjected to a likely non-linear representation, will be distributed over a manifold, hence from a distribution with support over a manifold, rather than from a distribution with support over all of $\mathbb{R}^n$ [Li19; Mal18; SOL16]. In many applications, the shape of interest is a low dimensional submanifold embedded in a higher dimensional space. A natural question to ask is how we can reconstruct or estimate the underlying manifold through some measured characteristics, e.g., curvatures. If the observations on a manifold (sampled point cloud) are sufficiently dense, a triangulation of the data points can approximately reconstruct the manifold for tracking associated geometric properties were it not for possible noise. The challenge of adequately reconstructing the manifold to obtain a good geodesic estimate may have motivated the use of a Euclidean metric even if the source data space structure had been recognized. Other works using Euclidean metric with additional information have yielded inadequate accuracy or high complexity with impediments to practical implementation.

To address the aforementioned issues in the existing approaches, we propose an OT-based cost formulation for domain adaptation. The proposed cost firstly helps preserve the underlying multi-class structure in each domain, where the true transport matrix, upon classification, should have a block-diagonal structure. In light of the intrinsicality of the latent representations to a submanifold embedded in a higher dimensional Euclidean space, our proposed alternative to a Euclidean distance is aimed at providing a refined estimate of the geodesic distance on the manifold. In so doing, we seek a metric so that the latent representations reflect the class labels and adequately safeguard the deep invariant features on the manifold. Following a discussion of some preliminary background on OT and domain adaptation in the next section, we develop our proposed approach in Section 3, where we first discuss a cluster-based OT and its associated adapted metric. In so preserving the underlying structure of the domain as well as the associated ambient curved space, the proposed metric is shown to reflect the latter’s curvature. Moreover, upon highlighting the importance of reducing the dependence of the captured latent embedding function on the classifier performance, we propose a dual collaborative OT framework which seeks to define a differential measure of goodness between an "agnostic" OT and a cluster-driven and implicitly Bayesian OT to optimize the domain adaptation. In Section 4, we provide a detailed experimental assessment and evaluation of the proposed work, and some concluding remarks in Section 5.

4.2 Background

For self-containment, we first review the concept of unsupervised domain adaptation, and its relation to the Monge-Kantorovich optimal transport optimization problem [San15]. The latter, as noted in the last section, depending on estimating the transport cost between samples on the submanifold
embedded in \( \mathbb{R}^n \), entails discussion of a geodesic distance estimation using the inherent geometry of the data.

### 4.2.1 Notation

We adopt throughout the paper, \( D \), \( X \), and \( Y \) to respectively denote the pixel space, the latent representation domain, and the label space. We use the lower case letter superscripts \( s \) and \( t \) to distinguish spaces or elements coming from the source or target, and the subscripted lower case letter to distinguish different elements in different spaces, i.e., \( x_i^s \) indicates the \( i^{th} \) object element of source domain \( X^s \). We will also denote the cost matrix with the upper letter \( C \), and the transport matrix (coupling) as \( \gamma \). We let \( f_\theta(\cdot) \) be a function parameterized by \( W_\theta \).

### 4.2.2 Unsupervised Domain Adaptation

In an unsupervised domain adaptation setting, one may consider a set of source latent features \( X^s = \{x_i^s\}_{i=1}^{N_s} \), which are extracted by a learnable embedding function \( f_E(\cdot) : D \rightarrow X \), with its corresponding set of labels \( Y^s = \{y_i^s\}_{i=1}^{N_s} \), where \( N_s \) is the total number of source samples. Let \( f_C(\cdot) : X \rightarrow Y \) be a classifier function that predicts the labels of the extracted latent representations. The classifier can be trained via supervised learning of source data, and it is said to be a well-trained classifier when predicted labels are closed to true labels.

Then, consider a set of unlabeled features from the target domain, \( X^t = \{x_j^t\}_{j=1}^{N_t} \), where features may be generated by the same or additional embedding functions from the observed target data. While no label information is assumed known for the target data, our goal is to use the same classifier trained on \( X^s \) to predict the labels of \( X^t \), with an understanding that source and target data share the same label space. The predictions on target data would only achieve the best performance if \( X^t \) has the same distribution as \( X^s \). However, source and target data are possibly collected by different modalities, which lead to a potentially substantial gap between source and target domains. Simply using the same embedding function and classifier generally falls short on account of this gap in general.

Addressing this issue led to the idea of domain adaptation by minimizing the gap in the latent space. The domain adaptation problem is guaranteed by the following theorem in,

**Theorem 6.** [BD10] For a hypothesis \( h \),

\[
\epsilon_t(h) \leq \epsilon_s(h) + d(X^s, X^t) + \min\{E_X,[[f_s(x) - f_t(x)]], E_X,[[f_s(x) - f_t(x)]]\}
\]

where \( \epsilon(h) \) defines the error made by a hypothesis \( h \) in the source or target domain, and \( f^s(\cdot) \) and \( f_t(\cdot) \) are true labelling function of source and target domain, respectively.

Theorem 1 shows that the target error of a given hypothesis (classifier) is bounded by the summation of the source error, the distance between source and target domain, and an optimal difference. Source error can be optimized by the supervised learning, and the optimal difference is
induced by factors beyond control. Minimizing the distance between source and target domain is thus the alignment term which can effectively improve the prediction performance in the target domain. Note that the essence of unsupervised domain adaptation problem is still a classification problem, that achieves the best predictions for the target data in the absence of any target label information during training.

4.2.3 Optimal Transport

Optimal transport (OT) theory was originally proposed in [Mon81] for the study of resource allocation problems. A geometric comparison of probability distributions is carried out by introducing a distance measure. The distance effectively transporting one distribution to another is achieved by minimizing the overall cost $C$. Formally speaking, let $X^s$ and $X^t \in \mathbb{R}^n$ be two measurable spaces and denote the set of all probability measures over $X$ by $\mathcal{P}(X)$. Given two probability measures $\mu_s \in \mathcal{P}(X^s)$ and $\mu_t \in \mathcal{P}(X^t)$, the Monge-Kantorovich relaxation formulates an optimization of the transport cost to find a probabilistic coupling $\gamma$ as a joint probability measure over $X^s \times X^t$ with marginals $\mu_s$ and $\mu_t$ for all $x^s \in X^s$ and $x^t \in X^t$ w.r.t. some cost measure $c(\cdot, \cdot) : X^s \times X^t \rightarrow \mathbb{R}_0^+$:

$$
\gamma^* = \arg\min_{\gamma} \int_{X^s \times X^t} c(x^s, x^t) d\gamma(x^s, x^t)
$$

where $A$ and $B$ are measurable subsets in measurable spaces $X^s$ and $X^t$, respectively. $\gamma^*$ denotes the coupling that achieves the minimal cost between the two probability densities.

In practice, with finite observations of distributions accessible to the experiments, we consider two discrete measures $\mu_s = \sum_{i=1}^{N_s} p_{x_i} \delta_{x_i}$ and $\mu_t = \sum_{j=1}^{N_t} p_{x_j} \delta_{x_j}$, where $\delta_x$ represents the Dirac function at point $x$, and $\sum_{i=1}^{N_s} p_{x_i} = \sum_{j=1}^{N_t} p_{x_j} = 1$, $p_{x_i}, p_{x_j} \geq 0$. The optimal transport in the discrete case is defined as following:

$$
\gamma^* = \arg\min_{\gamma \in \Gamma(\mu_s, \mu_t)} \langle C, \gamma \rangle = \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \gamma_{ij} C_{ij},
$$

where $C$ is the cost matrix computed by a pairwise cost measure $C_{ij} = c(x^s_i, x^t_j)$, with $L_2$-norm as a commonly used cost function, and $1_n$ is a vector of ones. Note that the discrete version of optimal transport is a linear programming task, with a dual formulation and alternative solutions exploited in various recent applications.

4.2.4 Geodesic Distance

Most machine learning approaches rely on pairwise distances between samples, and the choice of distance metric has a significant impact on the performance[Li19]. The selection of such a metric
depends on the data and the associated representation. As illustrated in Fig.4.1, the widely applied Euclidean distance, hence, is suited for when the data are concentrated on an unknown submanifold, and the intrinsic geometry of the data needs to be taken into account. An associated geodesic is the shortest path along the manifold with its geometric properties.

Recall that a geodesic is a curve on a manifold whose velocity is constant [Lee09]. For a given $M$ be a differentiable manifold, let $x \in M$, and let $v \in T_x M$ be a tangent vector to the manifold at $x$. Then there is a unique geodesic $\gamma_v$ satisfying,

$$
\begin{align*}
\gamma_v(0) &= x \\
\gamma_v'(0) &= v
\end{align*}
$$

The corresponding exponential map is defined by $exp_x(v) = \gamma_v(1)$. In general, the exponential map is only locally valid, where $exp_x : N(x) \to M$, where $N(x)$ is a small neighbourhood of $x$ in $T_x M$. In addition, the Hopf–Rinow theorem [Hop31; Atk75] asserts that in the case of a complete finite-dimension manifold as a metric space, the exponential map is defined on the whole tangent space.

Accounting for the stated specificities of the problem at hand as well as its associated theoretical underpinnings discussed in the background, we propose the next section, an alternative approach to the domain adaptation problem that yields an improved solution when co-opted with the proper geometry of the related ambient data spaces.

### 4.3 Equalizing Domain Variation: An Optimal Transport Approach

In introducing here, our proposed model first invoke the geometric structure of the conventional cost matrix, and relax its evaluation using a near-geodesic metric. By next adopting a Bayesian approach when invoking some problem-associated parameters, we introduce a collaborative optimal transport learning, with a proper geodesic-based cost together with the underlying structures of the predicted and true label information. The double-map transport learning collaboratively exploits the information from both predicted and true labels, and reduces the effect of the uncertainty of the training classifier.
4.3.1 Cluster-to-Cluster Optimal Transport

As noted in the previous section, one assumption in the existing approaches is that all data points lie in a Euclidean space, hence ignoring the underlying intrinsic space of the latent data. Bearing in mind that DA is fundamentally an inference problem, which entails the minimization of the intra-class distance between entities and the maximization of their inter-class distance. This class-induced clustering structure of the source and target domains injects some constraining ordering property which we implicitly introduce as a Bayesian prior. The source data labels are accessible as \( X_s = \{ X_s^k \}_{k=1}^K \), where \( K \) is the total number of classes. The predictions of target labels \( \hat{y}_j^t \) can be assigned by using the classifier or applying the \( k \)-nearest-neighbor (\( k \)-NN) classification method as, \( X_t = \{ X_t^k \}_{k=1}^K \). The cluster-to-cluster optimal transport (CCOT) is defined as following:

\[
CCOT(X^s, X^t) = \sum_k OT(X^s_k, X^t_k),
\]  

(4.3)

Using the label information as a basis for the source domain structure, yields an associated point cloud with a submanifold ambient space structure embedded in a \( n \)-dimensional Euclidean space. The intrinsic geometry thus imposes a modification of the pairwise transport cost given in Eq.(2). The geometric influence on the cost matrix is analytically quantified as follows,

**Proposition 7.** The following statements are equivalent:

(a) The optimal coupling between corresponding clusters is \( \gamma^*_k = \min_{\gamma_k} \langle C_k, \gamma_k \rangle \), where \( C_{k,i} = \| x_i^s - x_j^t \|_2 \) and \( x_i^s, x_j^t \) are in \( k \)th class.

(b) The optimal coupling \( \gamma^* \) between \( X^s \) and \( X^t \) is a block-diagonal matrix. In other words, \( \gamma^* = \begin{bmatrix} \gamma^*_1 & 0 \\ \vdots & \ddots \\ 0 & \gamma^*_K \end{bmatrix} \) up to permutation.

(c) The cost matrix is also a block-diagonal-like matrix, where the off-diagonal elements are infinity. \( C = \begin{bmatrix} C_1 & \infty \\ \vdots & \ddots \\ \infty & C_K \end{bmatrix} \).

(d) When source and target domains have not been reordered,

\[
\hat{C}_{ij} = \begin{cases} \| x_i^s - x_j^t \|_2^2, & \text{if } y_i^s = \hat{y}_j^t \\ \infty, & \text{otherwise} \end{cases}
\]

**Proof:** (a)\(\rightarrow\)(b) readily follows as we first determine the label prediction of the target data, and only allow transportation between source and target samples from the same class, which results in no inter-class assignment. \( \gamma^*_k = \gamma^*_{k} \), otherwise one can always find an optimal one \( \gamma^*_k = \min \{ \gamma^*_k^{(a)}, \gamma^*_k^{(b)} \} \) in either (a) or (b) statement. (b)\(\rightarrow\)(c) assumes that \( 0 \times \infty = 0 \). Off-diagonal elements in the coupling matrix exactly being 0 means that the off-diagonal elements in the cost matrix are much larger than block-diagonal elements. Consider the transportation probability goes to 0, the cost goes to infinity. (c)\(\rightarrow\)(d) is merely a restatement. (d)\(\rightarrow\)(a) is an alternative explanation of the
cluster-to-cluster optimal transport by setting the cost as infinity to lift any possible transfer between inter-class samples.

The $L_2$ deviation adopted in the cost meets the optimality of the transport when the data lie in a Euclidean space. Accounting for the intrinsic geometry of the submanifold of latent features used in clustering the various classes, and as noted earlier, is in order. Note that this cluster-to-cluster optimal transport is also a valid distance metric between source and target domains. The resulting reduction of the search space yields property, $d(X^s, X^t) \geq OT(X^s, X^t)$. In reference to Theorem 1, there is a trade-off between introducing the geometric property and the bound of target error. In so doing, our optimal distance between source and target domains is not exactly attained, and to better capture the associated geodesic distance between them on the unknown manifold, we use a refinement on the coupling matrix as detailed next.

4.3.2 Relaxation of CCOT

To account for the geometric properties of the relevant data space, the cost matrix is properly adapted to the cluster-to-cluster optimal transport, as just noted. This approximate refinement imposes, however, a constraint on the samples from different classes, preventing a prediction to cross a decision boundary. In other words, the transportation is only true when the information of target labels is known and accessible. We consequently, relax the infinite cost between inter-class connections by a penalty to preserve the block-diagonal structure of the transport matrix, $C_{ij} = \|x_i^s - x_j^t\|_2^2 + m$, if $y_i^s \neq \hat{y}_j^t$. A straightforward and naive implementation, however, yields a discontinuity in the cost function on account of the hard-decision condition, thus causing a jump for samples near and on different sides the decision boundary. This discontinuity in the optimization will cause Dirac delta singularity in the gradient calculation, making it non-smooth, and affecting the convergence with induced oscillations in the optimization.

To mitigate this difficulty, we proceed to (1) Soften the hard decision boundary allowing for cross-class decisions; (2) Safeguard the geometric properties underlying the space and reconstruct associated geodesic metric.

For a better estimate of distance between points concentrated on a submanifold, the exponential map provides us a simpler practical solution to build an inverse map that projects a point on the tangent space back to the manifold itself. On account of its analytical expression, we may view the label space as a tangent space on the manifold, which is constructed using a point cloud of latent features. Given $\tilde{y}_i^s - \tilde{y}_j^t$ as the tangent vector on the tangent space, constructed by the labels of the latent space $X$, the exponential map of the vector provides additional information and enhancement of the Euclidean distance. Intuitively, the Taylor expansion of the exponential function, $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$, clarifies the recovery of the geodesic curve via all information of different powers’ which is ordinarily only captured by a vector velocity on the tangent space, $\sum_i w_i (||\tilde{y}_i - \tilde{y}_j||^2)^i$. The higher-order moments will thus complement the curved space missing information of the
Euclidean distance. This newly proposed pairwise transport cost can thus be written as,

$$C_{ij} = c(x_i^s, x_j^t) = \alpha \|x_i^s - x_j^t\|_2^2 + (1 - \alpha)e^{\beta\|\hat{y}_i^s - \hat{y}_j^t\|_2^2},$$  \hspace{1cm} (4.4)$$

where $\hat{y}$ is some label information of latent representations, $\alpha$ is the convex combination parameter that balances the Euclidean distance and the geodesic-induced recovery information, $\beta$ controls the geodesic velocity on the tangent space.

**Proposition 8.** The optimal transport between the true source label $\hat{Y}^s$ and $k$-NN -based estimated target label $\hat{Y}^t$, is based on an implicit prior $p(k)$ to achieve the optimal cluster-to-cluster mapping.

**Proof:** The given true labels provide the strict clustering information of the source domain, and it thus also unveils the following empirical mean and covariance $m_k$ and $\Sigma_k$ parameters for each cluster under a Gaussian mixture model assumption., Moreover, the $k$-NN method, by assigning labels to close samples, indirectly forces the latent domain $X = X^s \cup X^t$ to be a mixture Gaussian distribution. The effective class prior probability is implicit in the optimal transport map's marginal of the source distribution, which is the measure of the source domain, $p(k) = \sum_k p(x_k^s) = \sum_k \mu(x_k^s) = \mu(X_k^s)$. The true source labels also further infer the mean $m_k$ and variance $\Sigma_k$ of each source cluster. The probability $p_{k,j}, x_j^t$ being from $k$-th cluster, can be associated with the joint probability between the $k$-th source cluster and the target sample, $p(X_k^s, x_j^t) = \sum_k \sum_{i,j} p(x_k^s, x_j^t)$, where the joint probability is accounted for in the optimal transport plan. Denoting by $f_{OT}(\cdot, \cdot; \theta)$ the model observed source and target samples from an unknown joint probability distribution resulting from an optimal transport map in terms of an embedding function parameters $\theta$, we can write the following the proposed model can thus be approximately expanded as,

$$\log f_{OT}(X^s, X^t; \theta) = \sum_{j} \sum_{k} \sum_{k_i} p(x_k^s, x_j^t) \log p(x_j^t|m_k^s, \Sigma_k^s),$$

where mean $m_k^s$ and variance $\Sigma_k^s$ as modeling information of the $k$-th source cluster. For a particular observation $x_j$, the optimal $\{p_{l,k}\}$ will be 1 for $k$-th cluster and 0 otherwise[VR14; Ric97]. Showing that target samples from the same cluster should follow the same probability distribution, $p(x_j^t|m_k^s, \Sigma_k^s) = p(X_k^s|m_k^s, \Sigma_k^s)$. MLE of the proposed model follows from the joint probability between corresponding clusters of source and target domains.

$$\log f_{OT}(X^s, X^t; \theta) = \sum_k \sum_{j} \sum_{k_i} \log p(X_k^s|m_k^s, \Sigma_k^s) = \sum_k \sum_{j} \log p(X_k^s|m_k^s, \Sigma_k^s),$$

thus highlighting the implicit prior knowledge of class probability to determine the parameters $\theta$ for which the clusters of the observed data have the highest joint probability.
4.3.3 Collaborative Optimal Transport

As shown in Eq.(4), the proposed geodesic distance aims to mollify the cost with respect to the label information of both source and target domains under any possible classification and subsequent transport/assignment and to mitigate a number of limitations. At first, the adopted classifier, on one hand, will highly influence the recovery quality of the manifold structure, and its uncertainty on the other hand, will impact the embedding function (e.g. only reaching a local minimal during training). The estimated source label in computing the discrepancy between source and target domains will, in addition, affect the clustering membership information in the model. To help alleviate this difficulty, we propose an implicit Bayesian approach using prior information in carrying out a (more informed) transport to be contrasted with its fully agnostic counterpart, and use the deviation between the two for a systematic adaptation. This so-referred collaborative transport as we next elaborate, helps in better accounting for the underlying manifold structure of the latent data, and in improving the training for the embedding function.

Model Description:

First, we demonstrate that the essence of the domain adaptation problem is to train a meaningful embedding function to minimize the distance between source and target domains, with the classifier free of a discontinuity inducing constraint in the training procedure discussed in Section 3.2.

**Proposition 9.** Minimizing the distance \( d(X^s, X^t) \) between source and target domains, using OT of Eq. (4), yields an generically optimal classifier \( f_c(\cdot): X \rightarrow Y \) for both source and target domains.

**Proof:** By Theorem 1, the error of a given hypothesis on the target domain is bounded by the distance, the optimal difference between source and target domains, and the error made in the source domain. Therefore, any applicable classifier of source domain, which minimizes the source error, should also work for the target domain, since the target error is bounded by the optimal distance.

Based on Proposition 3, the inclusion of an additional classifier can effectively decrease the perturbation (error) effect of the training classifier and increase exchangeable clustering infor-
where the elements of $\tilde{C}_1$ and $\tilde{C}_2$ are calculated via Eq.(4) with predicted labels, $\hat{Y}^s$ and $\hat{Y}^t$, given by the classifier $f_C(\cdot)$, and true source labels $Y^s$ and alternative predicted target labels $\hat{Y}^{t'}$, given by the $k$-NN method, respectively. We want the embedding function to collaboratively take advantages of information from both sides, and thus the two optimal transport maps should be further constrained tightly due to the fact of transportation between same pair of distributions, i.e., source and target domains.

**Proposition 10.** A dually collaborative transport map pair $\gamma_1$ and $\gamma_2$, converges to a single transport map at training completion, which results in a convergence between target predicted labels, $\hat{y}_j^t$ and $\hat{y}_j^{t'}$, given by different classifiers, reducing the label inaccuracy predicted by a single classifier.

**Proof:** At the conclusion of sufficient training, any classifier error between the predicted source labels $\hat{Y}^s$ and the true source labels $Y^s$ is sought to be $\|\hat{y}_i^s - y_i^s\|_2^2 \leq \epsilon$, where $\epsilon$ is some small positive value, and $\hat{y}_i^s = f_C(x_i^s)$. Given the optimal transport map as a joint distribution, showing the probability of one source sample being transported to one target sample, the modified cost reveals the additional information inserted the original distance expression. The corresponding optimal transportation is thus the conditional joint probability, where predicted and true labels support the conditions, $\gamma_{1,i_j}^s := p(x_i^s, x_j^s | \hat{y}_i^s, \hat{y}_j^s, \Theta)$ and $\gamma_{2,i_j}^s := p(x_i^s, x_j^s | y_i^s, \hat{y}_j^t, \Theta)$, where $\Theta$ is a set of model parameters and the optimal transport constraints, and Bayes’ rule $p(a_{i_j}|b_{i_j}, \Theta) = \frac{p(a_{i_j}b_{i_j}|\Theta)}{p(b_{i_j}|\Theta)} = \frac{p(b_{i_j}|a_{i_j}, \Theta)p(a_{i_j})}{p(b_{i_j}|\Theta)}$ may further be used for clarity, where $a_{i,j} = (x_i^s, x_j^s)$, $b_{i,j} = (\hat{y}_i^s, \hat{y}_j^t)$, and $\hat{b}_{i,j} = (y_i^s, \hat{y}_j^t)$. $Y$ is deterministic given a defined classification function, $f_C : X \rightarrow Y$, the term $p(y|x, \Theta)$ is then determined. Since $\Theta$ is a set of conditions on latent representations $X$, $y$ and $\Theta$ are independent, where $p(b_{i,j}, \Theta) = p(b_{i,j})p(\Theta)$. Based on the convergence of $\hat{y}_i^s$ to $y_i^s$ by the supervised learning of source data, and the modified cost is relaxed to a fully smooth formula in terms of classes and decision bounds, the convergence between two transport maps infers that $p(x_i^s, x_j^s | \hat{y}_i^s, \hat{y}_j^t, \Theta)$ and $p(x_i^s, x_j^s | y_i^s, \hat{y}_j^t, \Theta)$ should converge to the same point., which leads to $p(\hat{y}_j^t) = p(\hat{y}_j^{t'})$. The

\[ L_{OT} = \min_{\gamma_1, \gamma_2} a(\gamma_1, \tilde{C}_1) + (1 - a)(\gamma_2, \tilde{C}_2), \]
minimization of gaps between two transportation plans collaborates the local information by different classifiers, which highly reduces the prediction uncertainty by a single classifier.

The idea of the proposed model is to further shrink the search space for the optimal transport map by constraining the map to satisfy both classification conditions. The convergence of the two transport maps to one single map with respect to Proposition 4, is achieved by minimizing the Kullback–Leibler (KL) divergence \[K_{ul51}\], a difference measurement between probability distributions.

4.3.4 Algorithmic Solution

As noted in Theorem 1, the target-error bound also depends on the source error. To attain the optimal latent model of the source data, we can learn the embedding function \(f_E(\cdot)\) and the classifier \(f_C(\cdot)\) by minimizing the cross-entropy loss between the ground truth labels \(Y^s\) and the predictions \(\hat{Y}^s = f_C(f_E(X^t))\),

\[
\min_{f_E, f_C} \mathcal{L}_{CE}(Y^s, \hat{Y}^s),
\]

where \(\mathcal{L}_{CE}\) denotes the cross-entropy loss.

In \[Cut13\] an entropic regularization term was proposed to reduce the search space, and to achieve a computationally more efficient near-optimal transport. Similarly, a regularization of the predicted target samples is desired even if inaccessible to the training, it reduces the candidate space and also favors a model with more confident predictions of target samples, with a minimum entropy value vector achieved by a one-hot vector, which helps preserve distinct classes and maximize the inter-class "distance" in the target domain.

To sum up, the objective loss is written as:

\[
\mathcal{L} = \mathcal{L}_{CE}(Y^s, \hat{Y}^s) + \lambda_1 \mathcal{L}_{OT} + \lambda_2 \mathcal{H}(\hat{Y}^t) + \lambda_3 K L(\gamma_1^* || \gamma_2^*),
\]

where \(\mathcal{L}_{OT}\) is the collaborative optimal transport loss defined in Eq.(5), \(\gamma_i^*\), \(i = 1, 2\), are the optimal transport plans based on the cost matrix \(\tilde{C}_i\), the elements of \(\tilde{C}_i\) is defined by Eq.(4) with different label information, \(\mathcal{H}(a) = -\sum_i a_i \log a_i\) is the entropic regularization of each target prediction, \(K L(A||B) = -\sum_{i,j} A_{ij} \log B_{ij}\), and \(\lambda_1\), \(\lambda_2\), and \(\lambda_3\) are three hyper-parameters, balancing the contribution to the total loss.

4.4 Experiments and Results

To evaluate our proposed approach, we use some widely accepted domain adaptation datasets, for validation purposes as well as for comparison to several state-of-the-art methods with available online codes [Tze17; Dam18].
4.4.1 Datasets

The first task is adaptation three standard digit classification datasets: MNIST, USPS, and SVHN. Each dataset consists of 10 classes of digits, from 0 to 9. USPS is a low resolution dataset, while SVHN is more complex as each sample image is a mixed set of numbers in the same image. We follow existing works to construct three transfer tasks: USPS→MNIST, MNIST→USPS and SVHN→MNIST.

Office-31 is an advanced domain adaptation dataset. It consists of 31 categories in three domains: Amazon (W), DSLR (D), and Webcam (W). The 31 categories in the dataset include common objects in an office area, such as keyboards, calculator, and headphones. The Amazon data include a total of 2817 images, with about 90 images per class. These images were captured with a clean background at a uniform scale from Amazon. The DSLR data is the smallest set of images in three domains, with 498 low-noise high-resolution images in total. Specifically, 5 objects per class from 3 different viewpoints were captured for constructing the dataset. The Webcam data, consisting of slightly larger sets of DSLR objects, contain 795 low resolution (640×480) images with significant noise and white balance artifacts from from fore-discussed office data-set of 4110 images with 31 classes. By permuting the three domains, we obtain following adaptation tasks: A→W, A→D, D→W, W→D, D→A, and W→A.

Office-Home is a more challenging dataset for domain adaptation, consisting of four domains, each consisting of 65 categories. The four domains are Art (Ar) – artistic images in the form of sketches, paintings, ornamentation, etc.; Clipart (C) – a collection of clipart images; Product (P) – images of objects without a background and Real-World (R) – images of objects captured with a regular camera. The whole dataset of four domains contains 15,500 images, around 70 images per class on average. By permuting the four domains, we obtain following adaptation tasks: Ar→C, Ar→P, Ar→R, C→Ar, C→R, C→R, R→Ar, R→C, R→P, P→Ar, P→C, and P→R. Some samples are shown in Fig.4.3.

4.4.2 Experiments and Results

In our experiments, different architectures for the embedding networks were used in different data set scenarios,

- All digits were processed using LeNet-5.
- Office-31 and Office-Home data used Res-Net-50[He16] as the backbone network.

Each Model started with a pre-trained neural network which was further fine-tuned accordingly. In all considered cases, a linear classifier (a single-layer network) is trained simultaneously with the embedding function. The final prediction and accuracy of the target dataset are based on this trained linear classifier. The neural network is optimized by the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and a batch size of 128. The learning rate is constant over the training procedure. All hyper-parameters are carefully tuned, to yield the best results.
Tables 1, 2, 3 list the performance of our proposed model compared to a series of other domain adaptation methods on Digit, Office-31, and Office-Home datasets, respectively. As can be seen, the proposed model outperforms all other models by a margin on all tasks. Moreover, our model shows a better stability and improvement for both simple and hard tasks compared with existing OT-based approaches. The label information provided by different classifiers shrinks the searching space of optimal transport, and stabilizes the information adaptation between different domains. Moreover, the two collaborative classifiers will also prevent each other to be stuck at some local minimum during the training procedure, due to the fact of convergence between predicted labels.

**Table 4.1 Digit number results**

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST→USPS</th>
<th>USPS→MNIST</th>
<th>SVHN→MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDA</td>
<td>93.5</td>
<td>94.3</td>
<td>86.2</td>
</tr>
<tr>
<td>Fast OT</td>
<td>96.6</td>
<td>94.6</td>
<td>88.4</td>
</tr>
<tr>
<td>DeepJDOT</td>
<td>96.3</td>
<td>95.8</td>
<td>92.7</td>
</tr>
<tr>
<td>Our model</td>
<td>97.1</td>
<td>96.5</td>
<td>93.1</td>
</tr>
</tbody>
</table>
Table 4.2 Office-31 results

<table>
<thead>
<tr>
<th>Method</th>
<th>A→W</th>
<th>A→D</th>
<th>D→W</th>
<th>W→D</th>
<th>D→A</th>
<th>W→A</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDA</td>
<td>81.2</td>
<td>78.2</td>
<td>92.8</td>
<td>99.1</td>
<td>63.2</td>
<td>64.5</td>
</tr>
<tr>
<td>Fast OT</td>
<td>83.9</td>
<td>82.2</td>
<td>93.2</td>
<td>99.0</td>
<td>65.2</td>
<td>66.1</td>
</tr>
<tr>
<td>DeepJDOT</td>
<td>86.2</td>
<td>85.5</td>
<td>95.2</td>
<td>99.6</td>
<td>67.1</td>
<td>69.1</td>
</tr>
<tr>
<td>Our model</td>
<td>87.3</td>
<td>86.2</td>
<td>95.8</td>
<td>99.6</td>
<td>71.5</td>
<td>72.1</td>
</tr>
</tbody>
</table>

Table 4.3 Office-Home results

<table>
<thead>
<tr>
<th>Method</th>
<th>Ar→C</th>
<th>Ar→P</th>
<th>Ar→R</th>
<th>C→Ar</th>
<th>C→P</th>
<th>C→R</th>
<th>P→Ar</th>
<th>P→C</th>
<th>P→R</th>
<th>R→Ar</th>
<th>R→C</th>
<th>R→P</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDA</td>
<td>35.4</td>
<td>46.0</td>
<td>57.2</td>
<td>42.3</td>
<td>52.1</td>
<td>53.8</td>
<td>38.4</td>
<td>37.1</td>
<td>61.7</td>
<td>53.2</td>
<td>43.7</td>
<td>70.5</td>
</tr>
<tr>
<td>Fast OT</td>
<td>36.2</td>
<td>49.4</td>
<td>60.2</td>
<td>45.3</td>
<td>54.5</td>
<td>55.6</td>
<td>41.5</td>
<td>37.2</td>
<td>65.2</td>
<td>63.5</td>
<td>46.1</td>
<td>75.8</td>
</tr>
<tr>
<td>DeepJDOT</td>
<td>36.9</td>
<td>50.4</td>
<td>59.9</td>
<td>46.8</td>
<td>54.3</td>
<td>58.0</td>
<td>42.6</td>
<td>37.8</td>
<td>63.2</td>
<td>64.6</td>
<td>47.7</td>
<td>77.1</td>
</tr>
<tr>
<td>Our model</td>
<td>38.7</td>
<td>51.8</td>
<td>62.5</td>
<td>48.2</td>
<td>56.3</td>
<td>59.1</td>
<td>43.7</td>
<td>39.3</td>
<td>66.5</td>
<td>65.2</td>
<td>48.5</td>
<td>78.1</td>
</tr>
</tbody>
</table>

4.5 Conclusions

In this paper, we proposed a collaborative learning model for unsupervised domain adaptation using optimal transport theory and geodesic distance recovery. The proposed method first aims at constructing a meaningful manifold, which includes clustering information, for both source and target domains. We propose a modified cost to recover the geodesic distance on the manifold based on the exponential map of Riemannian geometry. The structural information is also contained and relaxed by modifying the cost matrix with the information of true and predicted labels. We propose a dual-map collaborative-learning system to exploit information across contrasting classifiers to minimize their discrepancy. The proposed model first build up inner-domain structure, and then transport between corresponding clusters to reduce mismatching between samples near the decision boundary. Finally, our proposed model provided improved adaptation performance on numerous domain datasets.

Our present work highlights the importance of accounting for the intrinsic space of the data and the associated metrics, which with an appropriate analysis formalism, promise more performing machine learning methodologies as illustrated by our locally optimal refinement (using an exponential map on a manifold) of the adopted metrics.
5.1 Conclusion

Our introduction of an optimization-driven approach to learning the dynamics of graphs in Chapter 2, addressed the difficulty of low sample rate and high variability of neuronal signals for detecting graphs and discovering the functional connectivity based on specific stimuli without knowing the ground truth of the physical neuronal connections. Three alternative perspectives were presented to capture the evolution of a graph in time while accounting for the dynamics generated by the nodal activities. Specifically, we follow the working principles of neurons, where instantaneous changes in functional connections cannot occur due to the consecutive response of brains. We, therefore, apply $L_1$-norm to build up this property to seek these continuous changes in time. Moreover, we not only reconstruct the pairwise connections between two neurons but also focus on high-order information (connections among three or more neurons). We exploit notions from Topology to uncover the connectivity information among three neurons and provide a valid algorithmic solution. The testing results on the synthetic data and real-world (mouse's brains) data demonstrate consistency with the biological understandings.

The third chapter explicates the contrastive loss, whose original idea is to measure and contrast positive with negative pairs, from a hypothesis-testing point of view by simply distinguishing whether a pair of images are from the same category. We subsequently introduce a bi-projector model, where we assume one projection extracts common features as similarity and the other extracts difference features as dissimilarity. We propose to apply a Jaccard-similarity-based loss to fuse the similarity and dissimilarity information extracted by the two projectors. In practice, we apply our proposed model structure and an associated loss as a plug-in unit to the state-of-art approaches. We also show
that our model can help improve the performance of the state-of-art contrastive self-supervised learning methods. The accuracy improvement is more pronounced for more complex datasets proving that the similarity measure should account for non-linearity (i.e., the embedding space of such entities) in the latent space. The efficiency of the algorithmic solution is also shown by the running time, where the complexity of the model is rarely increased. In addition, we use some images, whales and sharks, as samples to illustrate more details of extracted features in our model, where two projectors, as complementary to each other, can help stabilize the training and the learnable feature space. The proposed fusion can also capture non-linear features compared with conventional models.

Finally, we propose a collaborative learning model for unsupervised domain adaptation using optimal transport theory and geodesic distance recovery in the Chapter 4. The proposed method aims at reconstructing a sensible manifold, by first building up an inner-domain structure and then computing transportation between corresponding clusters to reduce mismatching between samples near the decision boundary. Our work also highlights the importance of accounting for the intrinsic space of the data and the associated metrics in contrast to using a simple Euclidean space assumption. In particular, we propose a modified cost to recover the geodesic distance on the manifold based on the exponential map of Riemannian geometry. The structural (clustering) information, originally computed in the cluster-to-cluster optimal transport, is also contained and relaxed by modifying the cost matrix with the information of true and predicted labels of source and target domains, respectively. We then propose a dual-map collaborative-learning system to utilize different classifiers’ information by minimizing the discrepancy of the transportation maps. In the end, our proposed model provides improved adaptation performance on numerous domain datasets compared with several existing approaches.

5.2 Future Work

Our work started building the connection between the biological and artificial neural networks, mainly focusing on the high-order information contained in the underlying structures. More future research should be pursued with the following several perspectives.

1. Natural Images Processing among Neurons in the Visual System

Although visual information path has been comprehensively studied at the brain region levels, the collaborations among neurons remains largely elusive. While a non-stationary process of the connections between neurons, stimulated by moving bars in different directions, has been revealed by our proposed dynamic graph learning model, and a simplex further defines high-order information (connections among three neurons), more work remains in investigating natural movie stimuli. Natural images consist of moving stimuli with different combinations of all directions, which the binary neuronal signals should more abstractly encode. Moreover, since the collaborations between/among neurons are more complex, the topological and higher-order information should be
expected in a more prominent way, and this should reflect the essence of the neuronal working sys-

tem. Finally, sparse/binary coding is still one of the challenging and interesting future works in this

interdisciplinary area. Neurons work in the sense of spike, which is binary signals, and correctly de-
coding the binary signals will help capture the precise understanding of neurons’ working principles.

2. Feature Extraction Analysis

In Chapter 3, the Jaccard similarity is applied to measure all-moment information by a bi-

projector system. More future research should first give a deeper analysis of the captured higher-

order information compared with conventional linear approaches. We should then focus on defining

each projector with more specific details/conditions (e.g., define what similar patterns are), which

will help our further understanding of the feature representations’ distribution in both hidden
spaces and latent space, and thus help explain what machines learn from the training samples.

Finally, non-linear similarity measures can also be considered as one of the study tasks, where

Jaccard similarity in our model is an approximation based on training neural networks.

3. Manifold Structure and Geodesic Distance Recovery

Although we assume that the samples in one dataset are dense enough to map to an unknown

manifold of data representations across the neural network layers, Euclidean distance, one of the

common choices of distance metrics being selected, is an insufficient and inaccurate approximation

to the ground truth. In addition, reconstructing the manifold from the point cloud (sampling from

the true space) is another challenging problem. Simple approaches lead to a large margin of error,

while complex approaches may result in unacceptable time-consuming processes. Therefore, a

reasonable and applicable geodesic distance recovery, which reflects the essence of the manifold
structure and can be calculated efficiently, is needed for future research in studying latent repre-

sentation space. Multi-scale information can also be introduced to recover the manifold by providing

additional information of varying degrees.

In summary, there is still a significant gap between learning biological principles and their
applications to an artificial model. Moreover, much information is approximated and ignored in real-

world applications to decrease the computational complexity and realize practical solutions. How
to reconstruct and refine that information from the noisy signals both theoretically and practically
is likely a long term goal in learning.
## BIBLIOGRAPHY

<table>
<thead>
<tr>
<th>Reference</th>
<th>Author(s)</th>
<th>Title</th>
<th>Journal</th>
<th>Volume</th>
<th>Issue</th>
<th>Pages</th>
</tr>
</thead>
</table>


A

ADDITIONAL EXPERIMENT DETAILS AND RESULTS IN CHAPTER 3

A.1 Training Setup Details

More setup details are provided in this section for reproducing our results conveniently. All models, such as previous models [Che20b; He20] and our Jaccard-based model, are set up on the same basis. Transformation methods are selected to be the same for each experiment, including normalizing the dataset, cropping and resizing an image, horizontal flipping an image, gray scaling an image, and changing the brightness, contrast, saturation, and hue of an image with a certain probability. The implementation is based on the Transforms package in PyTorch [Pas17]. ResNet-50 [He16] is selected to be the backbone in all models. The output dimension of the encoder is 2048, and the output dimension of the projector is 128. In the MoCo comparison experiment, we choose the size of negative samples as 4096, and the momentum update rate equals 0.999. All experiments are running on Nvidia Tesla V100 GPUs.

A.2 Experiment Results

A.2.1 MoCo vs. Our model

We modify the MoCo model’s architecture to apply our Jaccard-based similarity. The query encoder and two query projectors, which extract latent and hidden features of the query image, respectively, are updated directly with the loss, and the momentum encoder and momentum projectors, which
extract latent and hidden features of the keys, respectively, are updated with the momentum update rules. The Jaccard-based fusion term in the loss is defined as follows,

\[
\mathcal{L}_J(x, x^{k+}, x_i^{k-}, \theta, \theta_1, \theta_2, \theta^{(m)}, \theta_1^{(m)}, \theta_2^{(m)}) = -\log \left( \frac{e^{s^+}}{e^{s^+} + \sum_i e^{s_i^+ + d s_i^-}} \right),
\]

(A.1)

where \( \theta, \theta_1, \theta_2 \) are parameters of the query encoder and projectors, respectively, \( \theta^{(m)}, \theta_1^{(m)}, \theta_2^{(m)} \) are parameters of the momentum encoder and projectors, respectively,

\[
d s^+ = \| f_{\theta} \circ f_\theta(x) - f_{\theta_2} \circ f_{\theta_2}^{(m)}(x^{k+}) \|_2^2, \quad s_i^- = \langle f_{\theta_1} \circ f_\theta(x), f_{\theta_1^{(m)}} \circ f_{\theta_1^{(m)}}(x_i^{k-}) \rangle, \quad \text{and} \quad d s_i^- = \| f_{\theta_2} \circ f_\theta(x) - f_{\theta_2^{(m)}} \circ f_{\theta_2^{(m)}}(x_i^{k-}) \|_2^2.
\]

Since the modification of the conventional MoCo model is mainly about an additional projector, which is a two-layer fully connected neural network, and the loss is calculated in the matrix format. Therefore, the modified model only takes around 3 more seconds under the same running conditions, which is 1.05 times of the conventional model running time. Fig.A.1 shows that in both CIFAR-100 and Tiny-Imagenet-200 datasets, our model shows efficiency and improvement in the training process with k-NN classification method.

### Table A.1

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CIFAR</th>
<th>Tiny-Imagenet-200</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimCLR</td>
<td>1m15s/epoch</td>
<td>8m54s/epoch</td>
</tr>
<tr>
<td>Our model</td>
<td>1m18s/epoch</td>
<td>8m57s/epoch</td>
</tr>
</tbody>
</table>

### A.2.2 Different initialization of Projectors

In this section, we present how different initialization would affect the performance of the encoder and projectors. Although the encoder’s performance may also be affected by the initialization, Fig.A.2 presents a similar trend of the blue curve, which is the similarity between the positive pair in the latent space, given an alternative random initialization of the projectors. Compared with results shown in Fig.3.7 and Fig. 3.6, we can still observe that the similarity between the positive pair stays at a constant level after some training epochs. Random initialization also affects the projectors’ performance. For this training initialization, Projector 1 seems to be confused with positive and similar negative samples, while Projector 2 extracts better distinguishable hidden features, so they overall give a stable measure of similarity between the positive pair.
Figure A.1 In this additional example, our result shows that our model also preserve the non-linearity compared to conventional SSL.

A.2.3 Non-linearity based on Different Samples

We keep using the Gram-Schmidt process to remove linear features and check the similarity between second-order features, $\hat{q}_1 = q_1 - \frac{(q_1, q_2)}{(q_2, q_2)} q_2$ and $\hat{q}_2 = q_2 \odot q_2$, where $\odot$ is the element-wise multiplication operator. We treat images of whales as the positive pair, while images of shark are negative samples. In Fig.A.4, the blue and orange curves are non-linear similarity between positive and negative pairs, respectively. A constant higher value of similarity between the positive pair can be observed based on our model's results. In contrast, the two curves do not have any comparable trend in the left plot.
Figure A.2 Similarity between the positive pair and negative pairs in the latent space.

Figure A.3 Similarity between the positive pair and negative pairs in the hidden space.
Figure A.4 In this additional example, our result shows that our model also preserve the non-linearity compared to conventional SSL.