Neural Optimal Transport

Submitted in partial fulfilment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Akshay Raman

19BCE0467

Under the guidance of

Prof. Jayakumar Sadhasivam

SCOPE

VIT, Vellore

VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

May, 2023

DECLARATION

I hereby declare that the thesis entitled "Neural Optimal Transport" submitted by me, for the award of the degree of *Bachelor of Technology* in *Computer Science and Engineering* to VIT is a record of bonafide work carried out by me under the supervision of Dr. Jayakumar S.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: 19-05-2023

Akshay Raman

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "Neural Optimal Transport" submitted by **Akshay Raman (19BCE0467)**, **SCOPE**, VIT, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*, is a record of bonafide work carried out by him under my supervision during the period, 01. 07. 2022 to 30.04.2023, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfils the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: Vellore
Date: 19-05-2023

Signature of the Guide

Internal Examiner **External Examiner**

Dr. Vairamuthu S
Computer Science and Engineering

Acknowledgements

I am deeply grateful to my guide, Prof. Jayakumar S, for his support and guidance throughout the project. His expertise and patience have been invaluable to me and have played a crucial role in the success of this thesis.

I am grateful to VIT for the opportunity to conduct this project and for their resources and help. I would also like to thank faculties for serving on my committee and providing valuable feedback and suggestions during project reviews. Their insights were crucial in helping me to shape my research and write this thesis.

I am deeply thankful to my friends and family for their love and support during this process. Without their appreciation and encouragement, I would not have been able to complete this journey.

Executive Summary

Optimal transport is the study of the optimal allocation of resources. The study of optimal transport lies at the intersection of mathematics, economics, and computer science, with applications spanning numerous fields. This project approaches the optimal transport problem from a computational viewpoint and explores some of its applications in the real world. Different types of algorithms, including deep learning solutions, are developed. Moreover, these algorithms are tested on various inputs, such as probability distributions, binary grids, and images. The project introduces a novel deep-learning algorithm to solve multi-marginal optimal transport. Current optimal transport algorithms are computationally expensive and need to be more scalable. The performance of the popular optimal transport algorithms on discrete and continuous data is tested and compared with proposed deep learning solutions. Optimal transport algorithms like Sinkhorn are used in Density Functional Theory (DFT) to study the interaction of electrons inside atoms. Other applications of optimal transport, such as the Wasserstein distance metric and Wasserstein GANs, are also discussed in detail.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

OT	Optimal Transport
GAN	Generative Adversarial Network
ICNN	Input Convex Neural Network
DFT	Density Functional Theory
ITF	Inverse Transform Sampling
EMD	Earth Movers Distance

Symbols and Notations

ϵ	Sinkhorn value
a, b	Input marginals/distributions
γ, P	Optimal coupling
C	Cost Matrix
1_n	Ones vector
u, v	Scaling Vectors
R	Dissociation distance

1. Introduction

1.1 Theoretical Background

The shortest path principle guides most decisions in life and nature: When a commodity, a person or a single bit of information is available at a given point and needs to be sent at a target point, one should favour using the least possible effort. The Optimal Transport (OT) theory generalises that intuition in the case where, instead of moving only one item at a time, one is concerned with simultaneously moving several items from one configuration onto another.

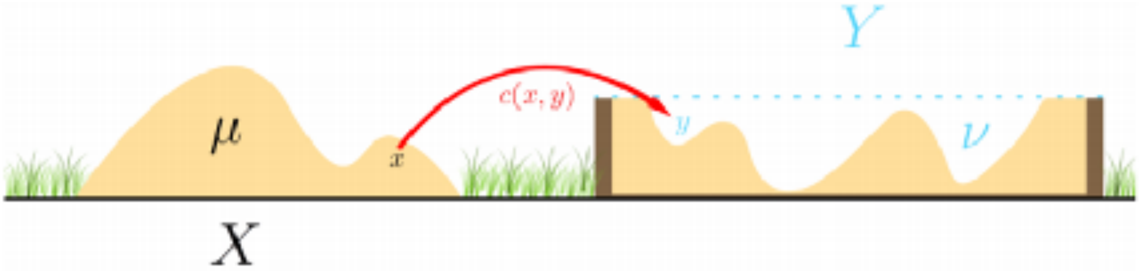

Figure 1.1: Illustration of optimal transport

Optimal transport theory is the study of optimal transportation and allocation of resources. The problem was formalised by the French mathematician Gaspard Monge in 1781. The original optimal transport problem involves transporting resources from source locations to target destinations at the least possible cost. Kantorovich generalised the original problem by finding an optimal coupling between measures instead of deterministic push-forward maps. For two marginals ‘a’ and ‘b’ with cost matrix ‘C’ defining the cost between elements, the optimal transport (primal) formulation is:

$$OT_C(a,b) := \min_{P \in U(a,b)} \langle C, P \rangle$$

Subject to the constraints,

$$P*1_m = a \text{ and } P*1_n = b$$

The goal of optimal transport solvers is to find the coupling ‘P’, which minimises $\langle C, P \rangle$, which is the element-wise dot-product of ‘C’ and ‘P’ while obeying the given constraints. The cost can be of many types, such as the distance between locations or the time taken to travel. We can reformulate the above minimisation problem into a maximisation problem involving dual multipliers ‘f’ and ‘g’. The Kantorovich dual formulation is:

$$OT(a, b) := \max_{f,g} \langle f, a \rangle + \langle g, b \rangle$$

Subject to the constraints,

$$f_i + g_j \leq C_{ij}$$

1.2 Motivation and Aim of the Proposed Work

The study of optimal transport lies at the intersection of mathematics, economics, and computer science, with applications spanning numerous other fields. Thus, developing fast, computationally efficient solutions to the optimal transport problem is essential. The project aims to develop and analyse various algorithms to solve the OT problem for different kinds of data. The project also focuses on finding solutions for multi-marginal optimal transport: a generalisation of the optimal transport problem. A lack of efficient algorithms to solve multi-marginal OT and scarcity of literature about the topic persuades the search for novel algorithms and approaches to tackle this problem. In particular, deep learning has been shown to solve such complex problems over the past decade.

1.3 Objectives of the Proposed Work

The objectives of this project are as follows:

- Analyse the performance of current OT algorithms.
- Implement deep learning solutions for the original OT problem.
- Test the algorithms on a variety of discrete as well as continuous data.
- Design and develop deep learning solutions for the multi-marginal optimal OT.
- Study applications of OT in different fields.

2. Literature Survey

2.1 Overview

Various algorithms have been developed to solve the OT problem in different domains. [14] G. Peyre; M. Cuturi, in their seminal textbook, summarise the forms of optimal transport along with their solutions. Traditional algorithms which solve the primal/dual form are versions of linear programming or approximate solutions like the Sinkhorn algorithm. Linear programming solutions deal with the dual form as an optimisation problem with constraints. Sinkhorn is an iterative algorithm that works on an entropic-regularised version of the OT problem.

Recently, with the advancement of computation power, new algorithms are constantly developed that utilise the power of today's computers. Moreover, the possibility of deep neural networks to solve such optimisation problems is being studied extensively. Neural networks are excellent universal approximators and can closely model various functions. The two most common applications of neural networks are (i) data prediction and (ii) data generation. Both applications can solve the OT problem.

[4] A. Korotin; L. Li show that generative neural networks can predict accurate transport maps and potential functions. Typical generative algorithms involve two neural networks competing with each other similar to a generator and discriminator in a GAN. Input Convex Neural Networks (ICNNs) can model convex functions and perform well in simple toy problems and image-to-image style transformation. Generative Modelling algorithms convert the optimisation problem of OT to a min-max problem so that neural networks can be used to find the optimal solution. These neural networks give good results in data generation tasks.

The previous algorithms solve OT for a particular pair of marginal 'a' and 'b'. Neural networks can learn the shared representation of various marginals by training them over many different pairs. [1] B. Amos, S. Cohen's Meta OT uses amortised optimisation to solve multiple OT problems and learns the shared structure and correlations between them. The neural networks return the solutions of the OT problem, which is fine-tuned using traditional algorithms like Sinkhorn. The output of the neural networks can be used as initialisation for Sinkhorn for faster convergence. The performance of Meta OT is multiple orders of magnitude faster than standard OT solvers.

Optimal transport has numerous applications in economics, physics, math, and biology. One application is in Density Functional Theory - A subfield of Quantum Chemistry. OT can be used to find the ground state energy of a system of strictly correlated electrons, which is crucial to understand the structure of other complicated atoms and molecules. Moreover, these OT algorithms are also used to solve problems in economics and game theory. Neural network algorithms can be combined with existing GANs to create Wasserstein GANs, which give state-of-the-art results compared to traditional GANs. Wasserstein GANs consist of a generator and discriminator, similar to ordinary GANs but with the Wasserstein (OT) distance as the objective function.

2.2 Survey of Existing Work

Meta Optimal Transport

This paper introduces the Meta optimal transport algorithm, which uses amortised optimisation to predict optimal transport maps from input measures. The algorithm involves a combination of various algorithms. The input is fed through a pipeline consisting of ResNet neural networks to get latent variables from input measures followed by ICNNs to solve the optimal transport problem. Finally, the Sinkhorn algorithm is used to fine-tune the results. The solution works on the dual or Kantorovich dual of the optimal transport problem. Meta OT models surpass the traditional convergence rate of log-Sinkhorn solvers in the discrete and continuous setting. The algorithms were tested on various images, spherical data, and colour palettes. The algorithm still needs to solve complicated optimal transport problems with different cost functions and performs poorly on out-of-distribution generalisation.

Neural Optimal Transport

This paper presents a novel neural-networks-based algorithm to compute optimal transport maps and plans for strong and weak transport costs. The paper's main contribution involves theoretical proving that neural networks can be used as universal approximators of transport plans between probability distributions. This paper works with the dual form of the OT problem. They first use the neural network to find one dual and then calculate the other using c-transforms. While the solver can work with one-to-one and many-to-one OT problems, it still cannot handle all kinds of tasks. The algorithm is only suitable for transport maps and cannot be applied or extended to solve the multi-marginal optimal transport problem.

Optimal transport mapping via input convex neural networks

In this paper, the authors present a novel approach to learning the optimal transport plan between two distributions using ICNNs. They use the dual form of the OT problem, which involves learning two convex functions, by solving a minimax optimisation. They propose a new framework to estimate optimal transport mappings as the gradient of a convex function trained via minimax optimisation. When trained between a simple distribution in the latent space and a target distribution, the learned optimal transport map acts as a deep generative model. Scaling this technique to larger data is challenging, but it is robust and can provide discontinuous solutions. The authors also found that the solution is independent of the initialisation of the model. Further, they concluded that a gradient of a neural network could easily represent discontinuous mappings, unlike standard neural networks that are constrained to be continuous. Neural networks allow the learned transport map to match any target distribution with many discontinuous supports and achieve sharp boundaries.

Do Neural Optimal Transport Solvers Work? A Continuous Wasserstein-2 Benchmark?

The authors of this paper aim to provide a standard benchmark to evaluate the performance of neural network-based solvers for optimal transport. The paper focuses on quadratic or Wasserstein-2 cost. The authors use input-convex neural networks (ICNN) to construct pairs of measures whose ground truth OT maps can be obtained analytically.

This strategy yields pairs of continuous benchmark measures in high-dimensional spaces such as spaces of images. The authors thoroughly evaluate existing optimal transport solvers using these benchmark measures. Even though these solvers perform well in downstream tasks, many do not faithfully recover optimal transport maps. To investigate the cause of this discrepancy, they test the solvers in a setting of image generation. Their main conclusion of the study reveals crucial limitations of existing solvers and shows that increased OT accuracy does not correlate with real-world performance.

Multi-Marginal Optimal Transport: Theory and Applications

This paper summarises the theory and applications of multi-marginal optimal transport, a generalisation of the original optimal transport problem. The authors analyse the primal and dual formulations of the OT problem. Furthermore, they also indicate the importance of the cost function in solving the OT problem. The paper addresses fundamental theoretical questions about multi-marginal OT, including the uniqueness and structure of solutions. Finally, the paper discusses some applications of multi-marginal optimal transport, mainly focusing on matching and economics and density functional theory in physics. In density functional theory, the ground state energy of a system of electrons is reformulated as a multi-marginal optimal transport problem. Its solutions can be used to understand the structure of various molecules and atoms.

Wasserstein GAN

This white paper outlines the main features of Wasserstein GAN – a variant of traditional GANs which use optimal transport. The paper first provides the theoretical analysis of Earth Mover (EM) distance, analogous to optimal transport/Wasserstein distance. This distance is compared with Euclidean and KL-divergence metrics. The authors show that EM distance is the better choice for comparison in specific scenarios. Wasserstein GANs are built on top of original GANs comprising of a generator and discriminator but use EM distance to measure the difference between their two outputs. This difference (the loss for both neural networks) can then be used to train the networks during back-propagation. The main limitation of Wasserstein GANs still lies in solving the optimal transport problem efficiently and accurately.

Computational Optimal Transport

This textbook comprehensively reviews optimal transport with a bias toward numerical methods. It covers the theoretical properties of optimal transport that can guide the design of new algorithms. The textbook starts with reviewing the mathematical knowledge of measures necessary to formulate the OT problem. The two main versions of OT, namely – the Monge problem (primary) and the Kantorovich problem (dual), are then discussed in detail. Next, an entropic regularised version of optimal transport is explained, which focuses on the Sinkhorn algorithm. The log-Sinkhorn algorithm is also discussed, which is faster in some instances. The following chapters discuss other forms and extensions of optimal transport.

2.3 Gaps identified in the Survey

The Sinkhorn algorithm is a fast iterative algorithm, but the convergence time depends on the amount of entropy in the OT problem. Generally, high entropy in the OT problem results in faster convergence but a less accurate solution, whereas low entropy results in slower convergence but a more accurate solution. The convergence rate also depends on the algorithm's initialisation, which is generally random. Choosing a better starting point may result in faster convergence to the optimal solution. Linear programming solutions, on the other hand, perform well on small-size input but suffer when the input dimension is large.

Neural network solutions perform well on most OT problems but can give inaccurate solutions in specific out-of-distribution scenarios. ICNNs have shown to be very good approximators of convex functions but cannot predict non-convex solutions to OT problems. Additionally, different input data are dealt with by different algorithms. For example, generative modelling only supports continuous probability distributions, whereas Meta OT works with discrete data. It is crucial to develop algorithms capable of handling both kinds of data.

Lastly, very few algorithms have been developed to solve multi-marginal optimal transport - a generalisation of optimal transport involving any number of marginals. Multi-marginal OT algorithms have considerable time as well as space complexities. The extension of Sinkhorn: multi-marginal Sinkhorn, can solve OT, but the performance degrades exponentially. Deep learning algorithms are yet to be tested on multi-marginal OT and have the potential to be faster than current solvers. Moreover, various techniques involving symmetry can be used to reduce memory requirements considerably.

3. Overview of System Components

3.1 System Architecture

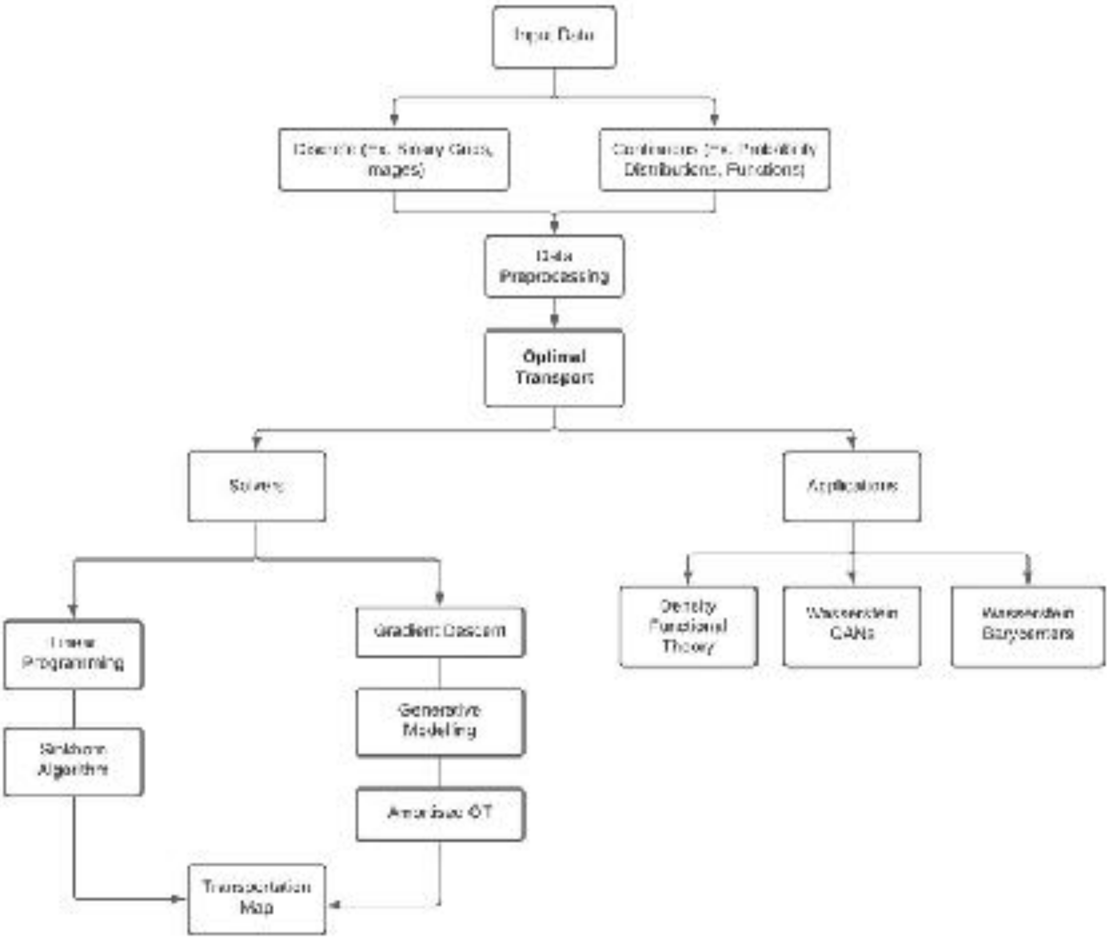

Figure 2.1: System Architecture and Components

3.2 System Description

Input data consists of source and target data. OT solvers aim to find the most optimal coupling between source and target data. The input consists of discrete data, such as images and binary grids, and continuous data, such as probability density functions and distributions. Data is preprocessed and then fed into the solvers. The primary preprocessing step is Inverse Transform Sampling which converts continuous distributions to discrete data. Additional preprocessing steps may include normalisation, resizing, and rescaling.

Many kinds of solutions for optimal transport are tested. Traditional algorithms that solve OT, like the Sinkhorn algorithm and Linear Programming, are implemented and tested. The neural network solutions include generative modelling (for continuous data), amortised OT (trained using MNIST), and gradient descent over the OT objective function. The output of the neural networks may be used as initialisation for the Sinkhorn algorithm to fine-tune the results. The final output is the transportation map or coupling of the inputs. Multi-marginal Sinkhorn and deep learning solutions which solved multi-marginal OT are also tested.

The project also implements and discusses applications of OT in different fields. The algorithms developed are used to solve problems in Density Functional Theory (DFT) to find the ground state energy of a system of strictly correlated electrons, which is crucial to understand the structure of other complicated atoms and molecules. Neural network algorithms can be combined with existing GANs to create Wasserstein GANs, which give state-of-the-art results compared to traditional GANs. Another application of OT involves Wasserstein distance which is extremely useful for comparing probability distributions.

4. Methodology

4.1 Introduction

This section discusses the methodology and algorithms used for each OT solver. Solvers are of two kinds: Traditional OT Solvers and Neural OT Solvers. Traditional Solvers include Linear Programming and Sinkhorn algorithms to solve OT, and Neural OT Solvers are deep learning solutions to the OT problem. Three types of neural solvers are discussed: (i) Generative Modelling, (ii) Amortised OT and (iii) Gradient Descent (for multi-marginal OT). Additionally, various pre-processing steps are described in detail, along with the novel use of Inverse Transform Sampling techniques to convert continuous input to discrete input.

4.2 Pre-processing

Input data consists of source and target data. We aim to find the most optimal way to find the coupling between source and target data. The primary constraint on the input is that the ‘masses’ of both the source and distribution are the same, i.e. they sum up to the same value. The input consists of different kinds of data:

1. Probability Distributions: Probability distributions can be discrete or continuous data which sum up to 1.0. It is the most common input type as it applies to many real-world problems. Additionally, distributions can also be multi-dimensional.

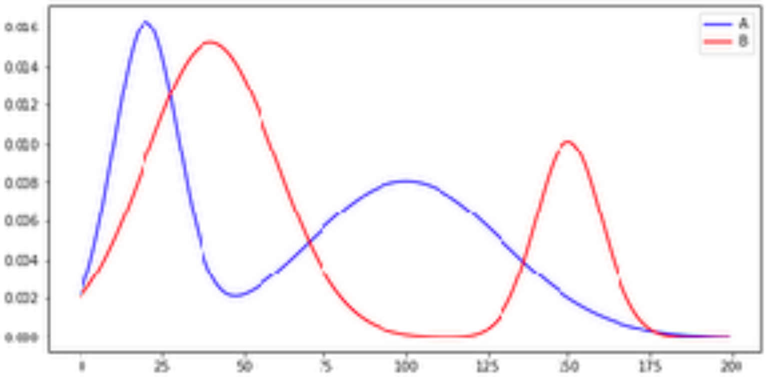

Figure 3.1: Probability Distribution Example

2. Binary Grids: Binary grids are a particular case of 2D probability distributions with uniform probabilities in some parts of the space. The optimal transport problem transforms into a tile moving problem where tiles must be moved to minimise the total cost.

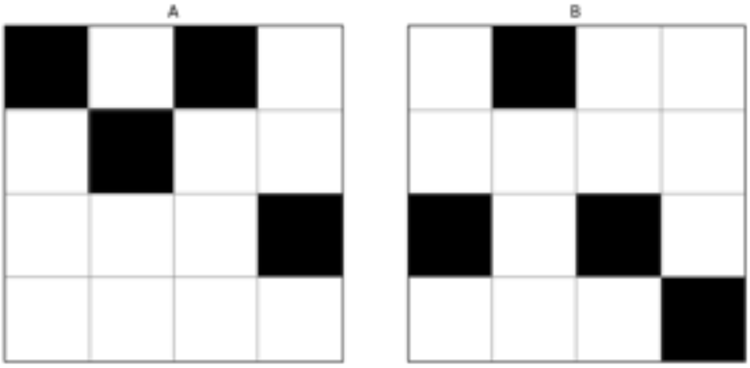

Figure 3.2: Binary Grid Example

3. Discrete Point Clouds: The source and target contain equal number of points in $\mathbb{R}^2$ space. The cost matrix associated contains the cost between source points and target points.

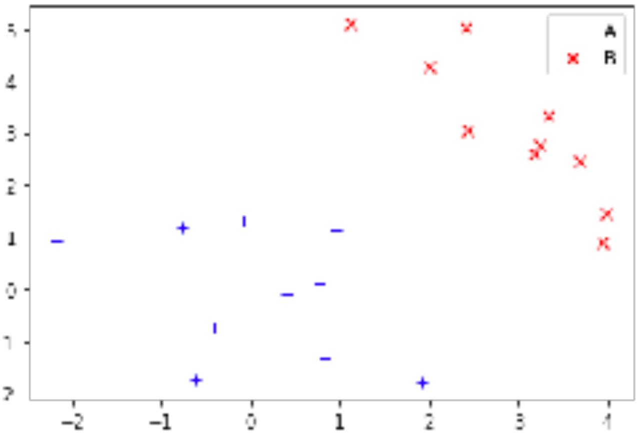

Figure 3.3: Point Cloud Example

4. Images: There are two ways images can be used as input for OT: (i) as a 2D probability distribution or (ii) Distribution of the colour space of the image. Both serve different OT problems. The OT solution for two colour spaces gives the ideal colour map between two images.

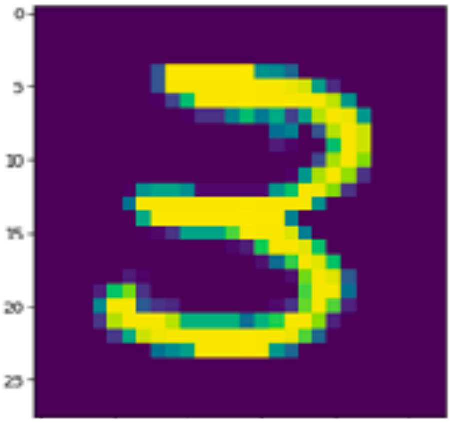

Figure 3.4: MNIST Image Example

Many other pre-processing steps, such as resizing, normalisation, and rescaling, are also used before the input is fed into the relevant OT solver. However, the main issue with most OT algorithms is the inability to work with continuous input data. Discrete data is suitable for computers as they store data as n-dimensional arrays, and various libraries like NumPy exist to work with them. Inverse Transform Sampling is proposed to convert any continuous input into discrete input while preserving the main characteristics of the data.

4.2.1 Inverse Transform Sampling

Inverse Transform Sampling (ITF) is a sampling technique used to sample from a continuous distribution. ITF maps the range [0-1] to the cumulative distribution function and then randomly choose a value from the range to generate sample datapoints from the distribution.

Assume we want to generate a random variable X with a cumulative distribution function (CDF) F_X. The ITF algorithm is:

1. Generate $U \sim$ Uniform(0,1)
2. Normalise inverse CDF.
3. Let $X = F_X^{-1}(U)$

Then, X will follow the distribution governed by the CDF F_X, which was our desired result.

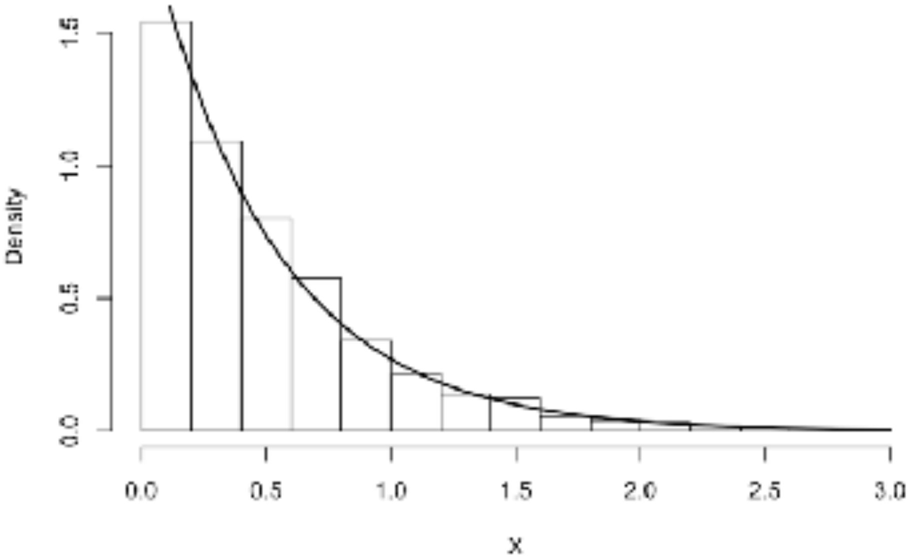

Figure 4.1: Inverse Transform Sampling

4.3 Traditional Optimal Transport Solvers

4.3.1 Linear Programming

These algorithms solve the original linear program OT program. There are multiple linear programming algorithms which solve the OT program. The most common is the Earth Movers Distance (EMD) algorithm which solves the EMD problem and returns the coupling matrix:

$$\gamma := \min_{\gamma} \langle \gamma, C \rangle$$

$$s.t. \, \gamma * 1 = a$$

$$\gamma^T * 1 = b$$

$$\gamma \geq 0$$

EMD is a distance that measures the similarity between two probability distributions, densities, or measures over a space. For probability distributions and normalised histograms, it reduces to the Wasserstein metric (OT cost). The idea was introduced by Gaspard Monge in 1781 through transportation theory. EMD can be found using many minimum-cost flow algorithms like the network simplex algorithm. The Python Optimal Transport (POT) library is an open-source library which provides several solvers for optimisation problems related to OT for signal, image processing and machine learning. The solution using the EMD function from the library solves the linear OT problem and returns the coupling matrix for the corresponding marginals and cost matrix.

4.3.2 Entropic-Regularised OT and Sinkhorn Algorithm

Since the solution to the primal problem can be highly non-linear, it is difficult to find solutions using linear programming. Therefore, an entropy term (H) is introduced to smoothen out the problem space. ϵ is introduced that decides the degree of entropy in the problem. The closer ϵ is to 0, the closer is the solution to the true OT solutions. The entropic-regularised OT problem formulation is:

$$OT_C(a,b) := \min_{P \in U(a,b)} \langle C, P \rangle + H(P)$$

where $\langle C, P \rangle$ is the element-wise dot product of C and P and $U(a,b)$ is the set of all possible couplings between a and b. $H(P)$ is the discrete entropy for a coupling matrix defined as:

$$H(P) := -\sum_{i,j=1}^{mn} P_{ij}\left(\log P_{ij} - 1\right)$$

The sinkhorn algorithm is an iterative algorithm to find solutions to entropy-regularised OT problems. It involves iteratively converging to the optimal solution using the Gibb's Kernel K and scaling vectors. The stopping time can be determined manually by specifying a particular number of iterations or check if the marginal predicted by the coupling coincides with the original marginals up to an error.

For the above entropy formulation, the OT solution is unique and has the form:

$$P* = diag(u)Kdiag(v)$$

Where (u, v) are scaling vectors, $(u, v) \in \mathbb{R}^n \times \mathbb{R}^m$, and Gibb's kernel K associated with the cost matrix C is:

$$K_{ij} := e^{\frac{-C_{ij}}{\epsilon}}$$

Subject to the constraints,

$$P^{*}1_m = a \text{ and } P^{*}1_n = b$$

Sinkhorn Algorithm Steps

1. Initialise $u = 1_n$
2. $u = a/Kv$
3. $v = b/K^T u$
4. Repeat steps 2 and 3 until convergence

The sinkhorn algorithm can also be extended to solve the multi-marginal OT problem. In this case, the OT solution is a coupling tensor and for n marginals, n scaling vectors are used. While the algorithm converges to the optimal solution, it is slow and has high memory requirements. The extended sinkhorn algorithm for multi-marginal OT is given is Appendix B.

4.4 Neural Optimal Transport Solvers

Neural networks are capable of solving a large number of highly non-linear problems. Hence, they can be used to predict solutions for OT. The OT problem is an optimisation problem; hence algorithms like gradient descent can be used to train the neural network to find optimal solutions for the transport problem. To this extent, three deep learning algorithms are implemented which solve OT for different types of discrete and continuous data. The hardware and software specifications and the model architectures used can be found in Appendix A.

4.4.1 Generative Modelling

Generative Modelling algorithms involve generating the transport map of the OT problem at each iteration by optimising the objective function. The algorithm comprises two neural networks: mapping network T and potential network f. The mapping network takes a position X and a vector of random values Z as the input and outputs another position Y indicating that mass from X must be transferred to Y. The potential network takes position Y from the input measure and outputs the potential associated with that position.

Generative Modelling Algorithm

```
1. Sample batches $X \sim A$ and $Y \sim B$
2. For each $x \in X$, sample batch $Z \in Uniform(0,1)$;
    2.1. $L_f = \overline{f(T(x,z))} - \overline{f(y)}$
    2.2. Update $w_f$ using $\frac{dL_f}{dw_f}$
3. Repeat $k$ times;
    3.1. Sample batches $X \sim A$
    3.2. For each $x \in X$, sample batch $Z \in Uniform(0,1)$;
        3.2.1. $L_T = \overline{C(x, T(x,z)} - \overline{f(T(x,z)}$
        3.2.2. Update $w_T$ using $\frac{dL_T}{dw_T}$
4. Repeat until converged
```

These two neural networks work together to find the optimal transport map between the input probability distributions. Inverse transform sampling is used to sample from the input measures, which are then fed into the networks for training. This algorithm supports discrete and continuous data (via inverse transform sampling). Additionally, the transport map is trained for more iterations than the potential network in each training epoch to produce optimal results.

4.4.2 Amortised Optimal Transport

Amortised optimisation considers solving multiple OT problems and learning the shared structure and correlations between them. A neural network f takes the marginals as input and returns the potential associated with the OT problem. Unlike generative modelling, amortised OT require a large number of training data. Hence, data is sampled from the MNIST dataset. The MNIST dataset consists of over 60000 hard-written images of numbers from 0-9. The OT problem treats each number as a 2D probability distribution, and the coupling gives the optimal way to transform one number into another.

Amortised OT Algorithm

```
1. Initialise potential model $f$
2. Sample Batch from dataset
    2.1. Sample $a, b, C$ from batch
    2.2. Calculate loss
        2.2.1. $g = \ln b - \ln(f * e^{\frac{-C}{\epsilon}})$ (c-transform of $f$)
        2.2.2. $P = e^{\frac{f}{\epsilon}} * e^{\frac{-C}{\epsilon}} * e^{\frac{g}{\epsilon}}$ (Coupling Matrix)
        2.2.3. $L_f = \langle f, a \rangle + \langle g, b \rangle + \epsilon \left( \sum a * b - \sum P \right)$
3. Calculate average loss of entire batch
4. Update $w_f$ using $\frac{dL_f}{dw_f}$
5. Repeat from step 2.
```

The loss function consists of (i) objective function and (ii) marginal condition. The dual formulation of the OT problem is used as the objective function. The marginal condition is checked using the optimal coupling P generated from the potentials. The second potential g is calculated by taking the c-transform of potential f. The final loss of the network is the average loss of each pair of numbers in a particular batch. The neural network output can be used as initialisation for the Sinkhorn algorithm to fine-tune the results as it serves as a much better initialisation for Sinkhorn than random values.

4.4.3 Multi-marginal Optimal Transport and Gradient Descent

Multi-marginal OT is a generalisation of the original optimal transport problem. It solves the OT problem for an arbitrary number of marginals, and the output is a coupling tensor (instead of a matrix). This coupling tensor can be interpreted as finding a relationship (or coupling) between all the concerned marginals. Due to the problem's high dimensionality, few algorithms can solve the multi-marginal case. The Sinkhorn algorithm mentioned above can be generalised to solve multi-marginal OT by updating multiple scaling vectors. The multi-marginal Sinkhorn algorithm is mentioned in Appendix B. This solver's space and time complexity is high and requires powerful computers.

The gradient descent algorithm thus aims to explore using neural networks to solve multi-marginal OT. The dual OT formulation is the objective function that the neural networks seek to minimise. Moreover, additional terms such as marginal constraints and regularisation terms can also be added. Different types of objective functions can be used to train the neural network. The algorithm for three marginals of the same type 'a' is given below:

Gradient Descent Algorithm for Multi-marginal OT

1. Initialise potential network f
2. Predict potential $f(a)$
3. Generate coupling tensor $P = K \odot e^{\frac{f}{\epsilon}} \odot \cdots \odot e^{\frac{f}{\epsilon}}$
4. Calculate objective function $J = \sum_{i=1}^{n} \langle f, a \rangle + \epsilon \sum P$
5. $L_f = -J$
6. Update w_f using $\frac{dL_f}{dw_f}$
7. Repeat from step 2.

5. Results and Discussions

5.1 Traditional Optimal Transport Solvers

Traditional OT solvers include Sinkhorn and algorithms which use Linear programming solutions. Different types of data are tested on both kinds of algorithms. The main constraint on the input is that the 'masses' of both the source and distribution are the same, i.e. they sum up to the same value. The input consists of different kinds of data, such as probability distributions, images, binary grids, and point clouds.

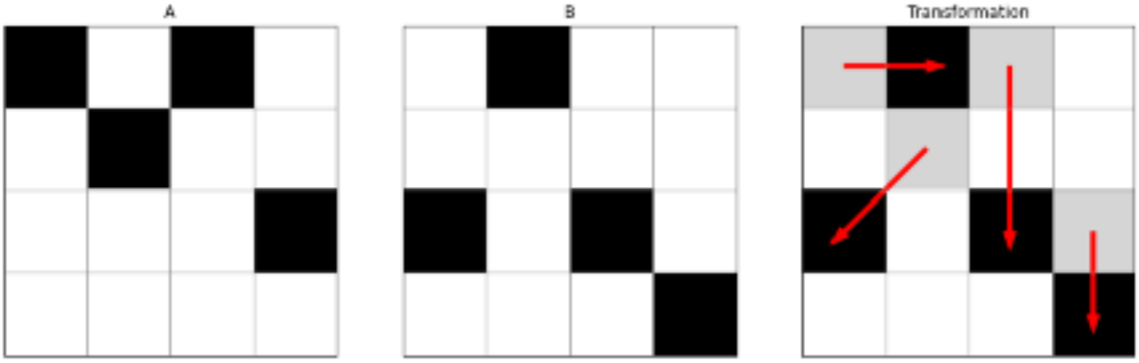

Figure 5.1: Optimal Transport on Binary Grids

Figure 5.1 shows two binary grids of size 4x4. The OT algorithm finds the optimal way from state A to state B. The left matrix in Figure 5.2 is the cost matrix associated with A and B. The cost matrix stores the cost between black tiles in A and B. The left matrix is the associated coupling matrix from which the transport map is generated. A binary grid is a special kind of 2D probability distribution. So, if A and B do not have the name number of tiles, then the algorithm will split mass from a black tile instead of moving that particular tile.

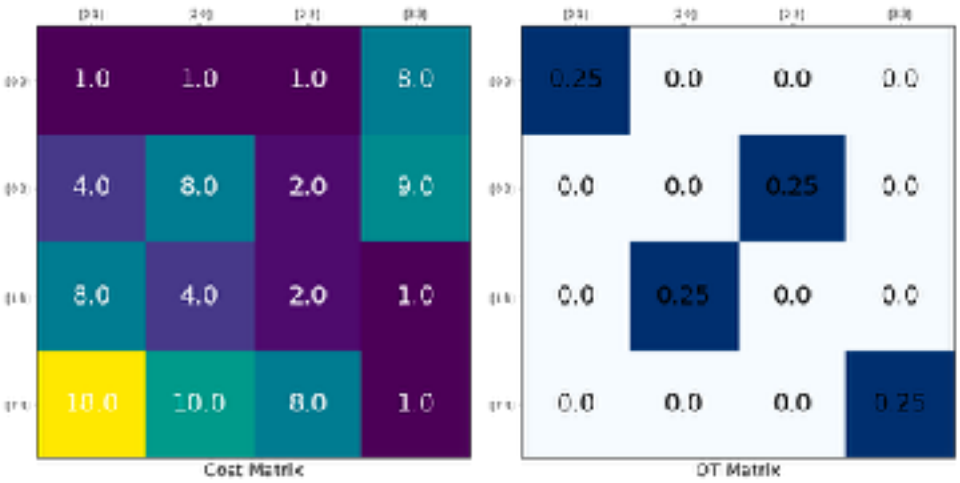

Figure 5.2: Cost Matrix and Optimal Coupling for Binary Grids

Another example is 2d discrete point clouds. Source and target contain an equal number of points in $\mathbb{R}^2$ space. The cost matrix associated contains the cost between source points and target points.

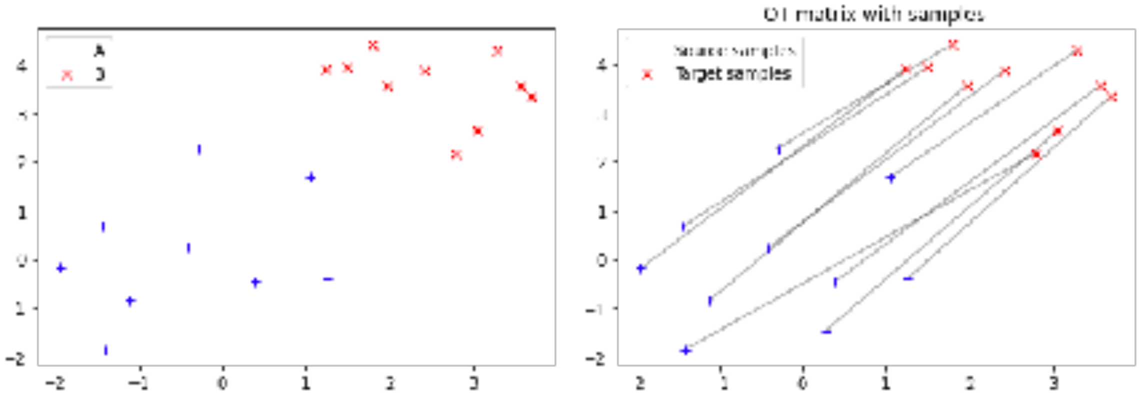

Figure 5.3: Optimal Transport on Discrete Point Clouds

The OT solver returns the transport map between the point clouds, which minimises the total cost (euclidean distance) of moving points in A to B. Figure 5.4 shows the cost matrix and coupling matrix associated with the above point clouds. The sparse coupling matrix has only one non-zero value in each row/column, indicating an optimal transport map between A and B.

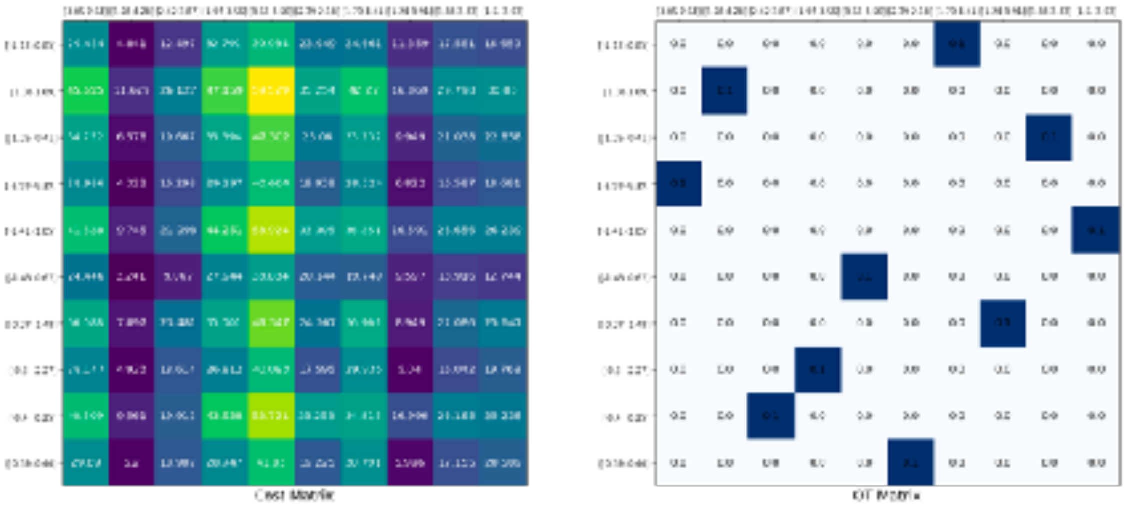

Figure 5.4: Cost Matrix and Optimal Coupling Point Clouds

The Sinkhorn algorithm is an iterative algorithm for solving entropy-regularised OT problems. It involves iteratively converging to the optimal solution using Gibb's Kernel K and scaling vectors. The example below shows the solution of Sinkhorn on 1D marginals which are a mixture of Gaussian distributions. The distributions are normalised to 1.0 so that they have equal masses. Figure 6.1 show the marginals and associated cost matrix. Euclidean square distance is chosen as the cost function.

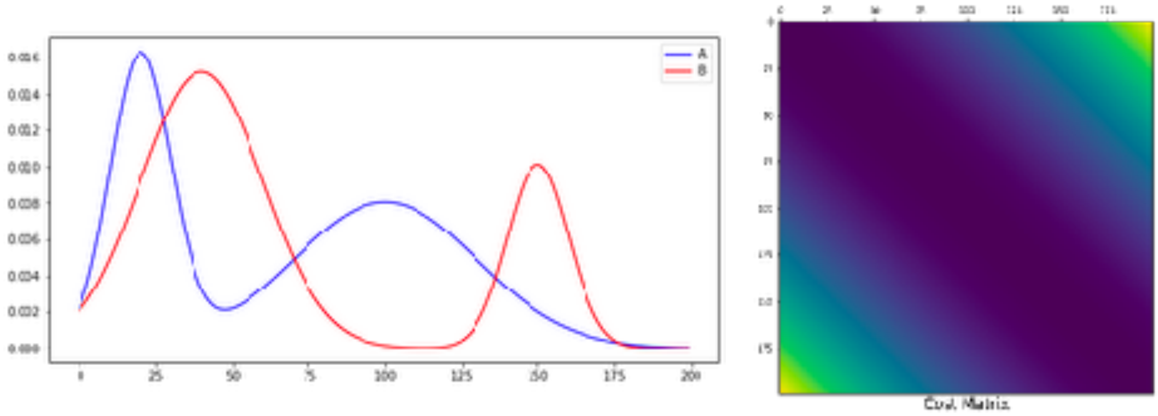

Figure 6.1: Mixture of Gaussian Distributions

Many types of distance functions were tested for the cost matrix. Some include Euclidean, Euclidean squared, l-n metric, and Coulomb cost. The coupling matrix for $\epsilon = 0.01$ is shown in Figure 6.2 below. On comparing the optimal coupling for different values of ϵ and with the coupling generated by linear programming, the Sinkhorn solutions are approximate and do not generate an accurate transport map between the marginals. Additionally as $\epsilon \to 0$, the solution converges to the true optimal solution, and the Wasserstein distance reaches its minimum.

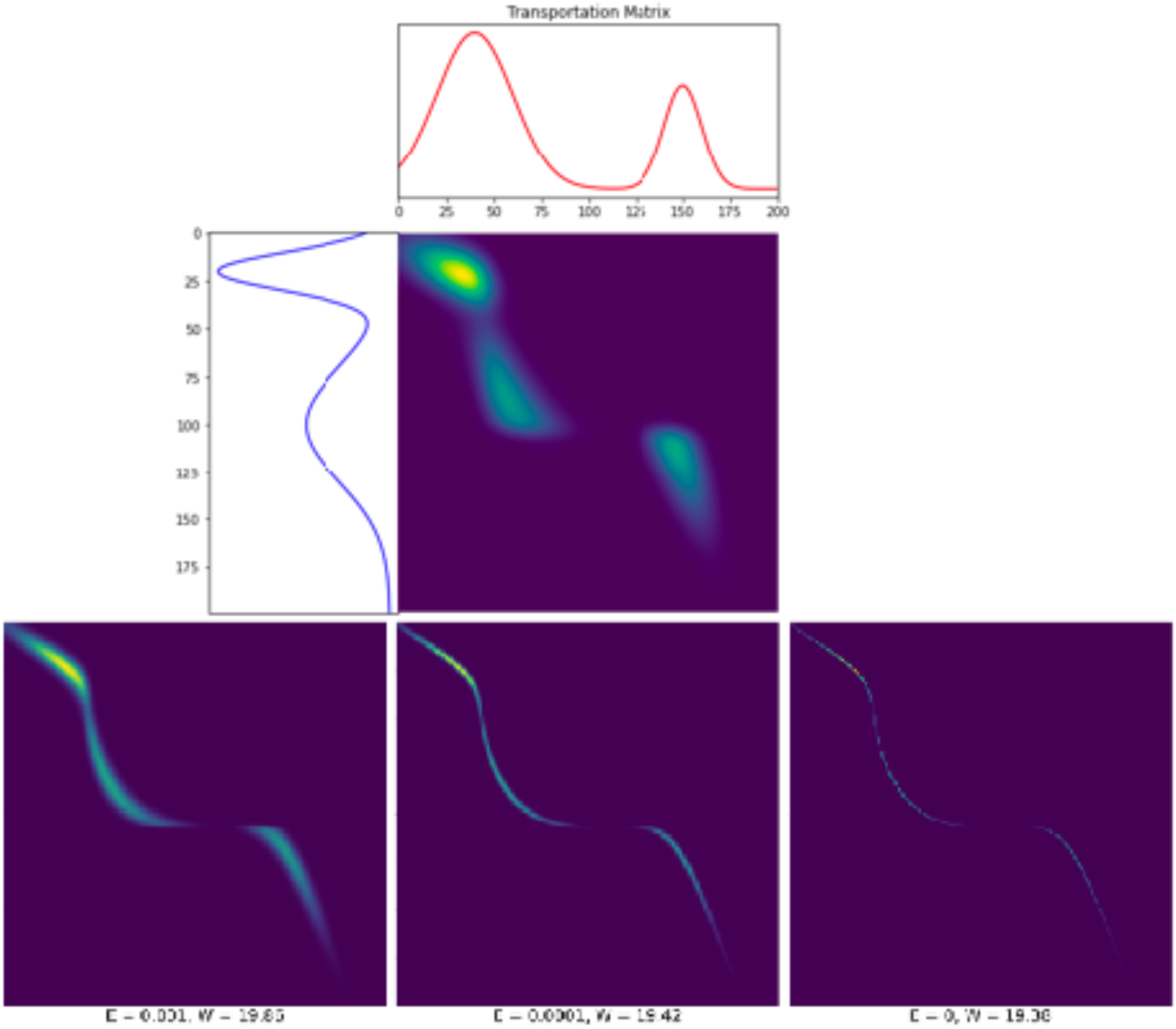

Figure 6.2: Optimal coupling for different values of ϵ

The Sinkhorn algorithm is a fast iterative algorithm, but the convergence time depends on the amount of entropy in the OT problem. Generally, high entropy in the OT problem results in faster convergence but a less accurate solution, whereas low entropy results in slower convergence but a more accurate solution. Additionally, the performance of Sinkhorn also depends on the input size. Figure 6.3 shows the exponential growth in time as ϵ gets smaller. Table 1 summarises the performance (time taken in seconds) of Sinkhorn for different values of epsilon and the input size.

Epsilon	0.1	0.01	0.001	0.0001
Time (sec)	0.004	0.033	0.329	3.039

Table 1: Sinkhorn algorithm Performance vs. ϵ

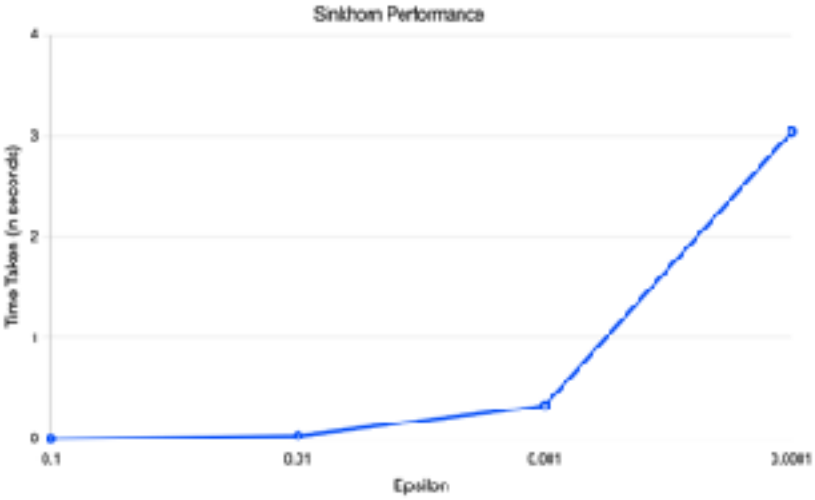

Figure 6.3: Plot of Time taken by Sinkhorn vs. ϵ

There are only two ways to make OT solutions more accurate: (i) Decrease ϵ value or (ii) Increase the size of input data, i.e. discrete input is a better representation of the continuous distribution. From Table 2 below, running Sinkhorn for large input sizes is impractical. Moreover, powerful processors and large amounts of memory are required for large input sizes. Neural OT solvers can be used to solve some of these problems. However, space complexity remains a challenge for such algorithms.

Input Size N	$\epsilon = 100$	$\epsilon = 10$	$\epsilon = 1$	$\epsilon = 0.1$	$\epsilon = 0.01$
10	0.001	0.000	0.000	0.002	0.006
100	0.001	0.001	0.001	0.002	0.011
1000	0.042	0.043	0.046	0.092	0.520
10000	4.313	3.231	3.009	5.755	34.462

Table 2: Sinkhorn performance for different input sizes and ϵ

5.2 Neural Optimal Transport Solvers

5.2.1 Generative Modelling

Generative Modelling uses two neural networks which compete with each other to solve the OT problem. The objective function, a minimisation (primal form) or maximisation (dual form) problem, is transformed into a min-max function. The algorithm is similar to the working of GANs in which the generator generates data, and the discriminator predicts if a particular data sample is real or fake. The neural network T generates the transport map, and the potential network f outputs the potential associated with the given inputs.

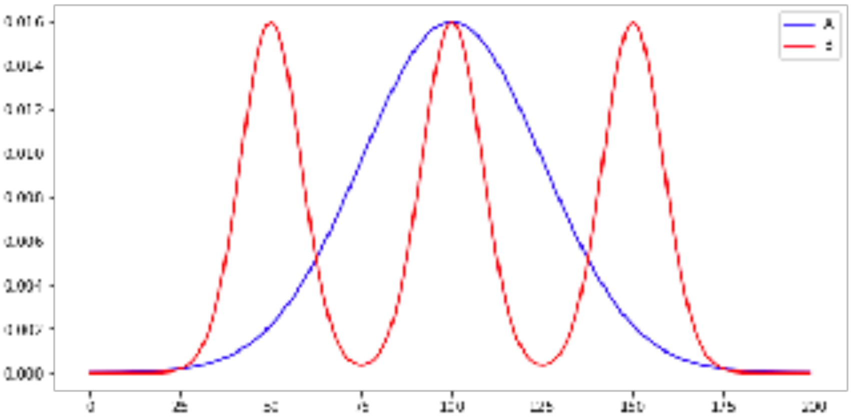

Figure 7.1: Source and target distributions for Generative Modelling

The generative modelling algorithm is tested on a variety of discrete and continuous inputs. Continuous 1D distributions are converted to discrete data using inverse transform sampling. Figure 7.1 shows an example of a mixture of Gaussians created using inverse transform sampling. The neural networks used were multi-layer perceptrons with 100 neurons in the hidden layer. A batch size of 64 was chosen for discrete probability distributions (of size 200). Adam optimisers with learning rates of 5e-3 and 1e-3, along with weight decay, were used for the potential and mapping networks, respectively.

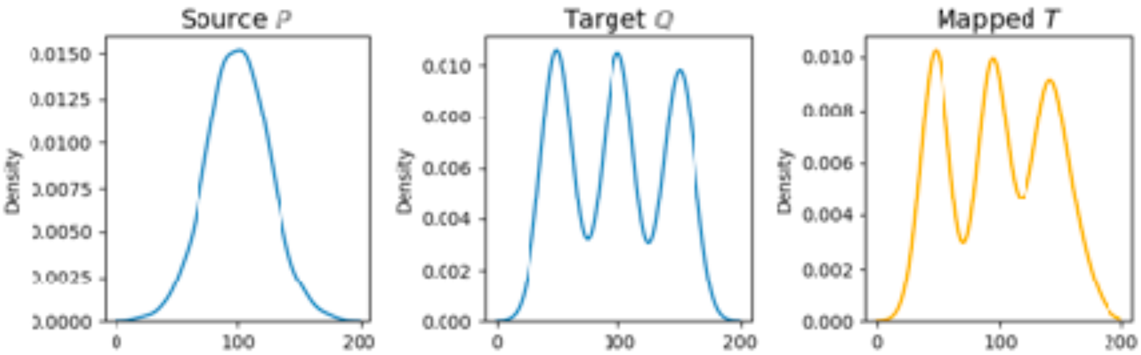

Figure 7.2: Sampled input distribution along with predicted target distribution

After training the networks for ~10000 epochs, the mapping network T could output the appropriate output position given the source position. Figure 7.2 shows the output of the distribution generated by the transport map when input was sampled from the source ℙ. Although the neural network can generate accurate transport maps, their hyper-parameters are specific to the input marginals and change when the input changes. Moreover, while training, the networks were more prone to getting stuck in local minima, even with optimisers with adaptive learning rates.

5.2.2 Amortised Optimal Transport

Amortised optimisation solves multiple OT problems and learns the shared structure and correlations between them. A neural network f takes the marginals as input and returns the potential associated with the OT problem. Input is sampled from the MNIST dataset. Figure 8.1 shows two examples of handwritten digits from the dataset. Each image is treated as a 2D probability distribution, and the neural networks find the optimal mapping between them.

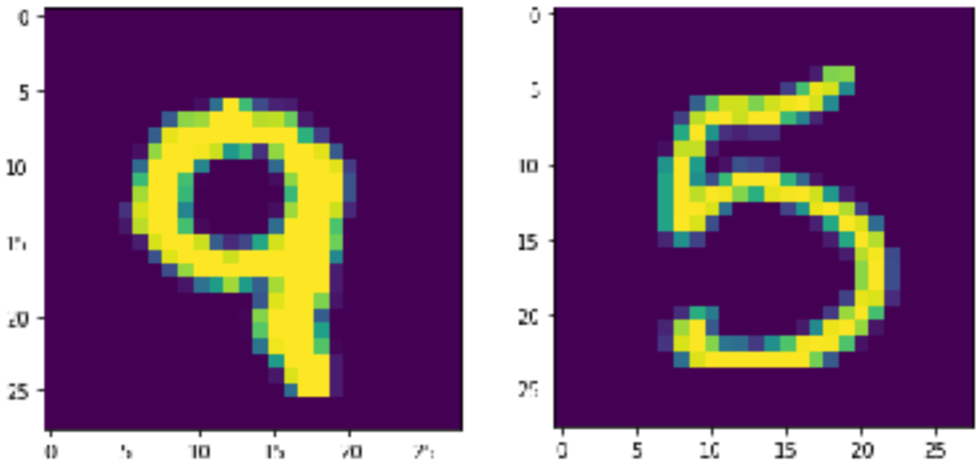

Figure 8.1: MNIST images samples

Amortised OT is very useful when multiple OT problems need to be solved from a single dataset. By training over multiple examples, the neural networks can predict accurate transport maps even for new data. Transfer learning can be used to train the neural network on a specific input. Additionally, the output of the neural network, which is the potential f can be used as initialisation for Sinkhorn. This initialisation is much better than constant or random inputs for scaling vectors.

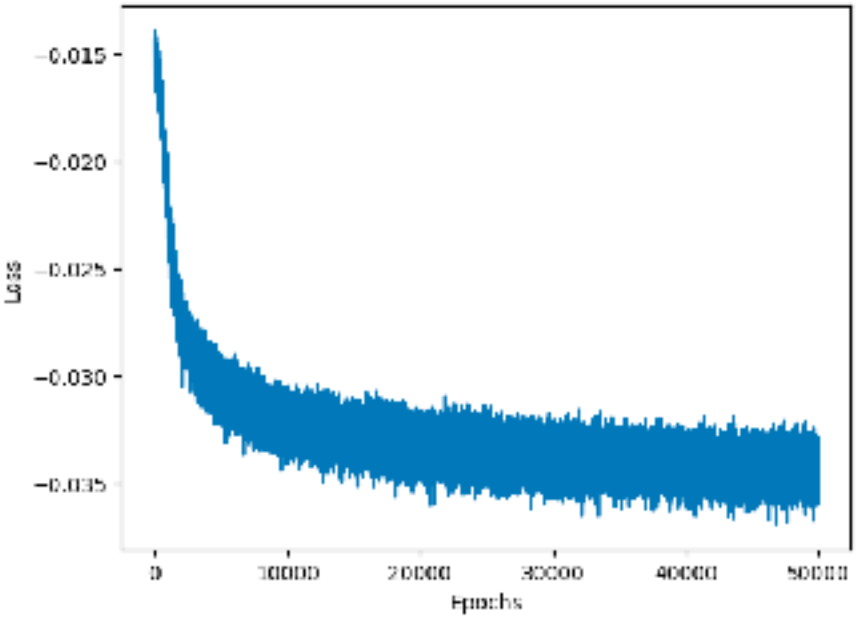

Figure 8.2: Plot of training loss vs epoch

Figure 8.2 shows the plot between training loss and the number of epochs. The model is trained for ~50000 epochs with a batch size of 128. The total loss is the average loss over the entire batch. Each element from the batch consists of two vectors of size 784 for each of the numbers for which the OT problem is solved. The neural network is a multi-layer perceptron with 1024 neurons in the hidden layer. The inputs are normalised (0-1) before being fed into the neural network. RMSProp with learning rate 1e-3 is the optimiser for training the neural network. A significant benefit of using the OT objective as the loss function is that the model can never overfit. That means the greater the training epochs, the higher accuracy of the model.

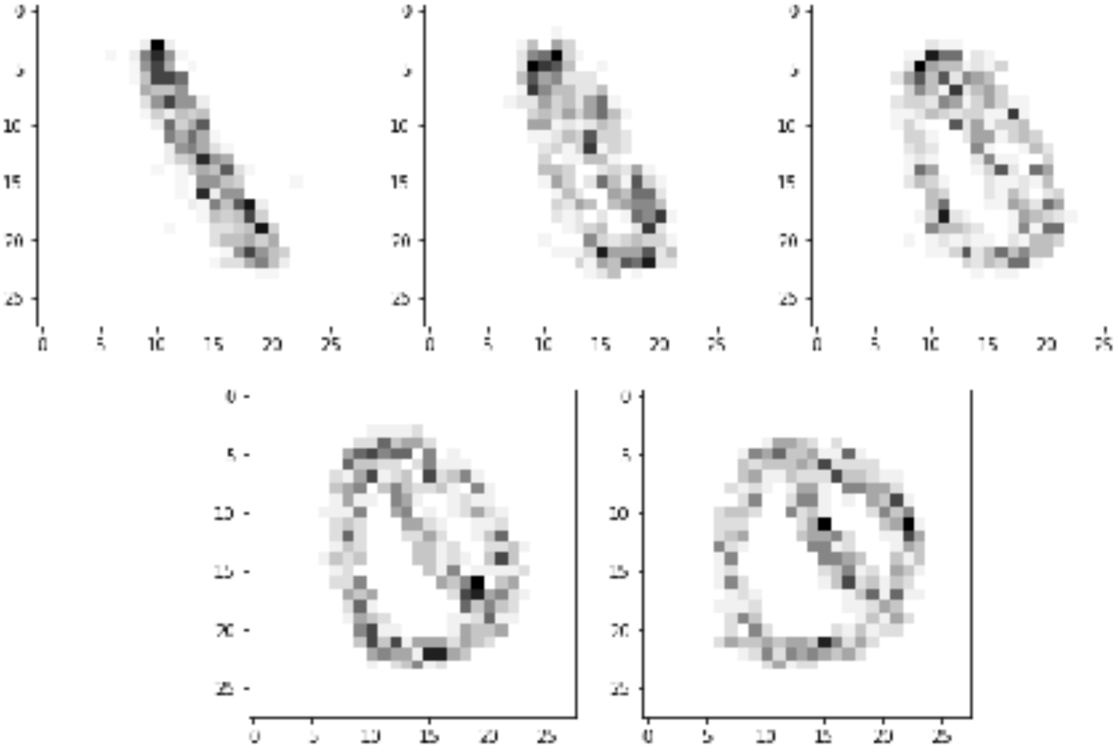

Figure 8.3: MNIST digits transformation using optimal transport

5.2.3 Gradient Descent

Since OT is an optimisation problem, it can be the objective function of the neural network directly, along with some constraints. The gradient descent algorithm works with the dual formulation as the objective function. Additionally, this algorithm can also solve multi-marginal OT, which involves solving the OT problems for multiple marginals (instead of two). Figure 9.1 below shows the input Gaussian of size 50, along with the cost matrix used. Shifted coulomb is chosen as the cost function. This cost function is a shifted version of coulomb distance which avoid infinity when the distance between the two points is 0.

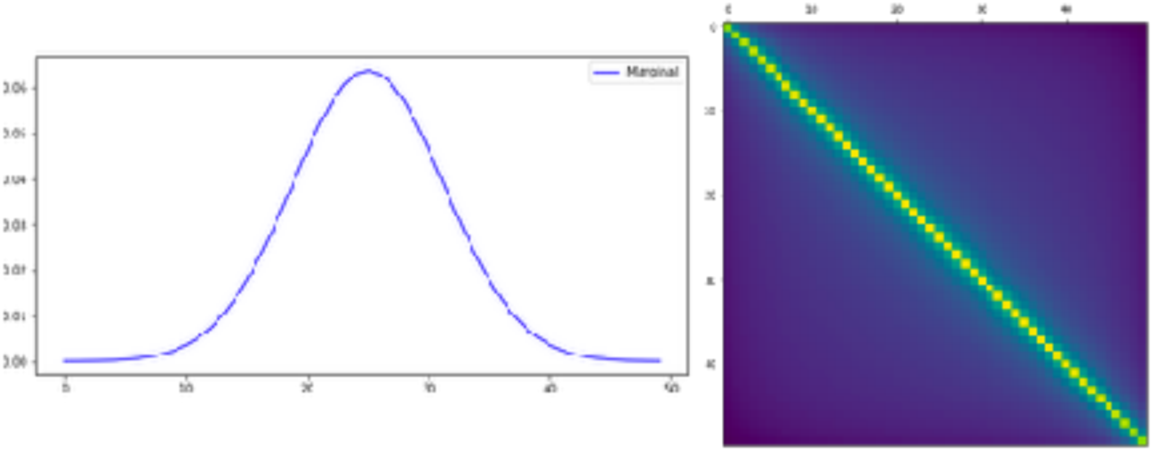

Figure 9.1: Input gaussian distribution and shifted-coulomb cost matrix

The neural network is a single-layer neural network with a hidden layer of size 200. The network's output is the potential (f) common to all the marginals. The objective function calculates the optimal dual subject to marginal constraints. Adam was chosen as the optimiser to train the neural networks. A custom learning rate schedule was also defined to control the learning rate further. While the neural network could predict accurate potentials and couplings for the problem, the training proved challenging as the objective function did not stabilise with the number of iterations. Figure 9.2 shows the projection of the optimal coupling tensor and the potential generated by the neural network.

5.3 Multi-marginal Optimal Transport

Multi-marginal OT is a generalisation of the original optimal transport problem. It solves the OT problem for an arbitrary number of marginals, and the output is a coupling tensor (instead of a matrix). This coupling tensor can be interpreted as finding a relationship (or coupling) between all the marginals. Due to the problem's high dimensionality few algorithms can solve the multi-marginal case. The Sinkhorn algorithm mentioned above can be generalised to solve multi-marginal OT by updating multiple scaling vectors. The multi-marginal Sinkhorn algorithm is mentioned in Appendix B. This solver's space and time complexity is high and requires powerful computers.

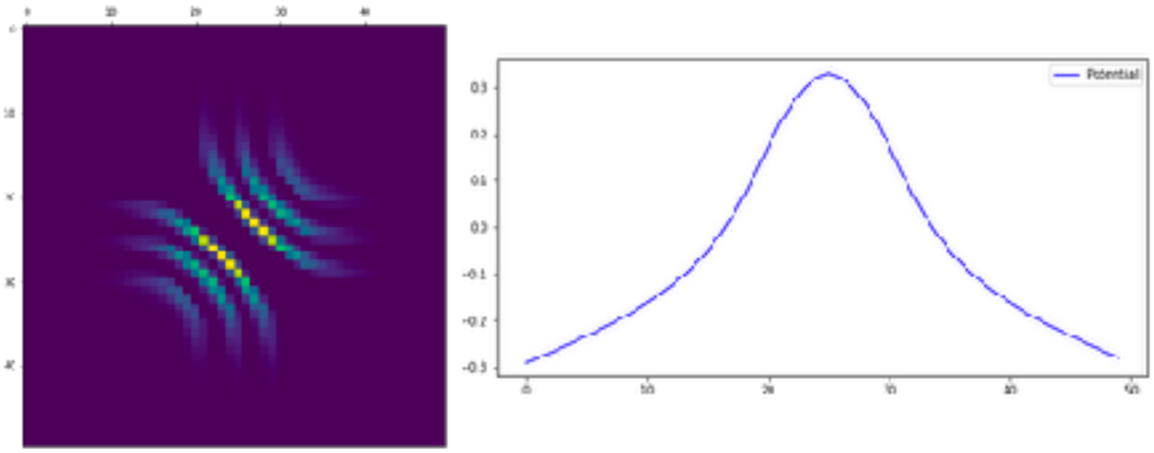

Figure 9.2: Projection of optimal coupling and kantorovich potential predicted by neural network

Figures 10.1 and 10.2 show the projections of the coupling tensor generated by the multi-marginals Sinkhorn algorithm for the inputs and cost function mentioned in the previous section. The coupling tensor is of size S^N where S is the size of the input marginals and N is the number of marginals. After testing for different values of ϵ, the multi-marginal Sinkhorn algorithm converges to the optimal solution as $\epsilon \to 0$. Figure 10.3 and table 3 summarise the Wasserstein distance for different ϵ values.

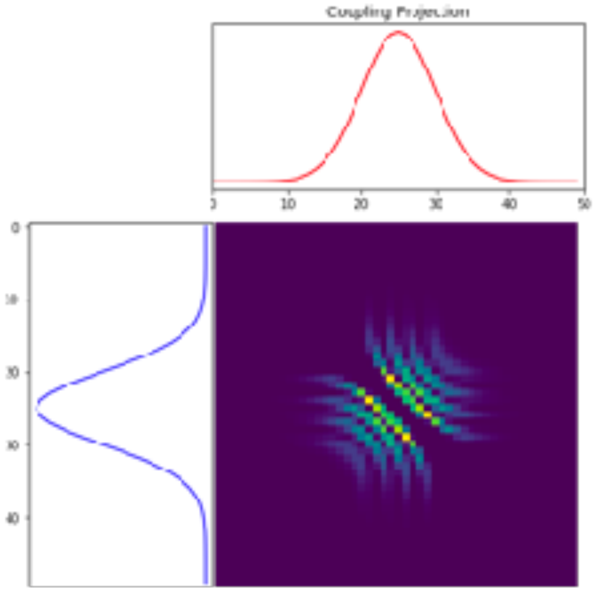

Figure 10.1: Optimal Coupling from Sinkhorn

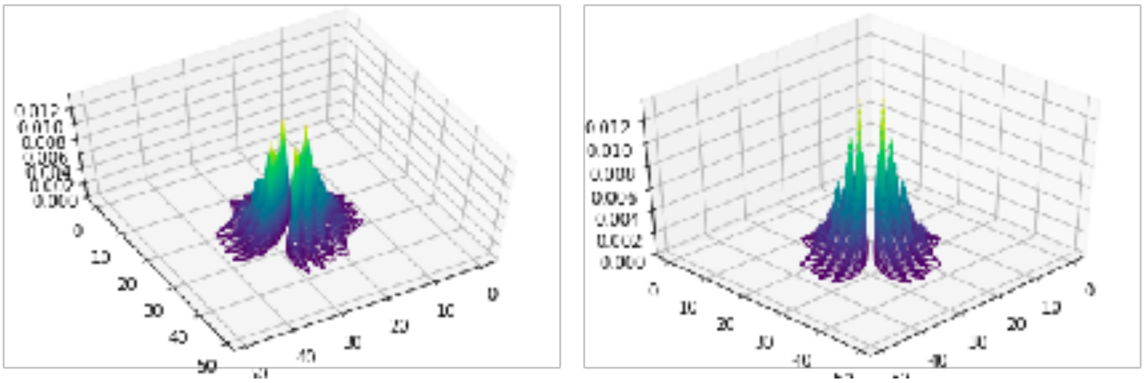

Figure 10.2: 3D projection of coupling tensor viewed from different angles

Epsilon	100	10	1	0.1	0.01	0.001
W	0.758	0.747	0.662	0.480	0.418	0.411

Table 3: Wasserstein distance vs. ϵ

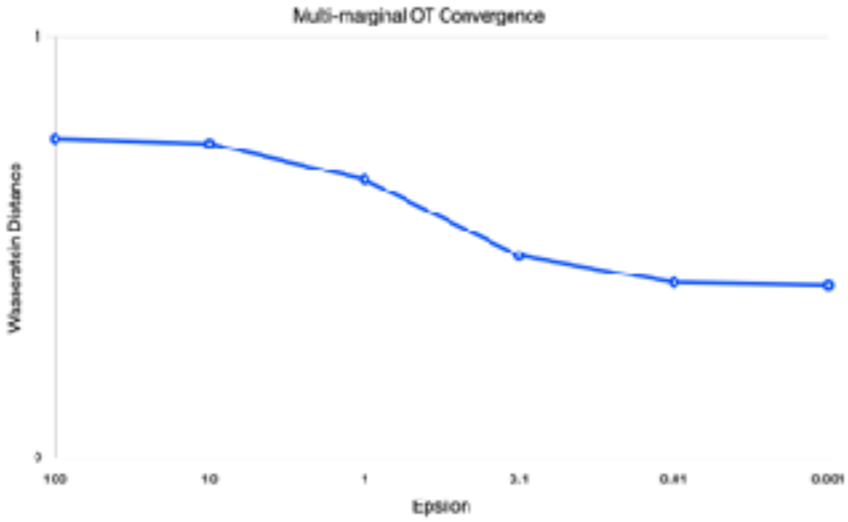

Figure 10.3: Convergence of Multi-marginal Sinkhorn

However, after comparing the time taken for the algorithm for the different numbers of marginals (refer Table 4), it was observed that the time complexity and the space complexity grow exponentially with the number of marginals. Hence, solving the OT problem using Sinkhorn for over seven marginals is infeasible for current computers. The performance and accuracy of the algorithm are non-complementary. The time taken by the algorithm can be improved but will affect the accuracy of the solution. The optimal coupling projections for the different number of marginals is given in Appendix C.

Marginals	$\epsilon = 100$	$\epsilon = 10$	$\epsilon = 1$	$\epsilon = 0.1$	$\epsilon = 0.01$
2	0.021	0.011	0.018	0.027	0.183
3	0.073	0.037	0.071	0.152	1.152
4	0.776	0.947	1.485	4.399	23.659
5	31.224	37.334	57.066	198.764	1525.699

Table 4: Sinkhorn performance for different number of marginals and ϵ

5.4 Applications of Optimal Transport

5.4.1 Density Functional Theory

Density Functional Theory (DFT) is a theory from Quantum Chemistry used to investigate the electronic structure of atoms and molecules. DFT treats atoms and their corresponding electronic structure as a one-body density function. By focusing on probability densities instead of the individual electron, many complicated problems in quantum chemistry, such as solving many-body problems, could be reformulated using DFT. It is a very flexible theory and can handle almost any arrangement or configuration of electrons.

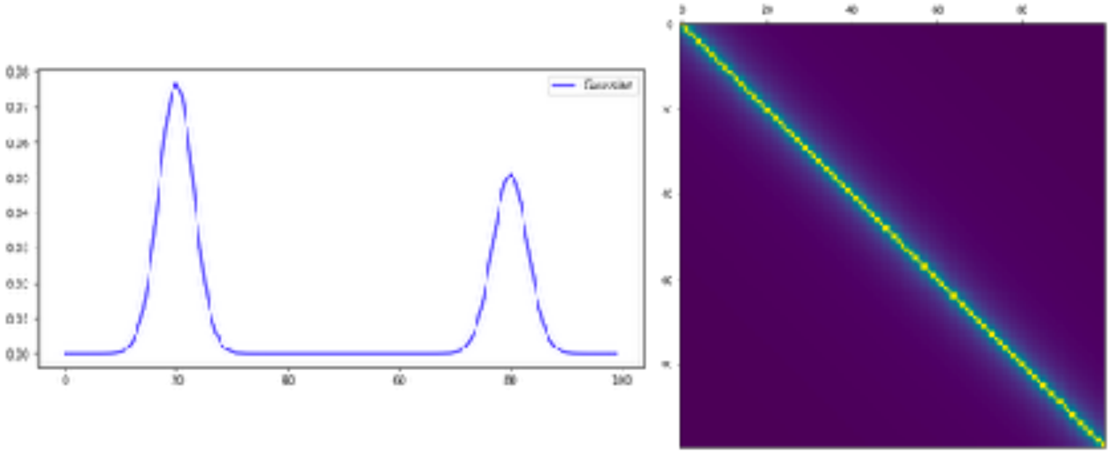

Figure 11.1: Input gaussian mixture and shifted-coulomb cost matrix

Optimal Transport can be used to study the dissociation of two atoms qualitatively. To model the dissociation of electrons, a superposition of two Gaussians with their centres separated by a distance R. The two Gaussians in Figure 11.1 represent individual atoms. The marginal function is given by:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x - R/2}{\sigma}\right)^2} + \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x + R/2}{\sigma}\right)^2}$$

where σ is the standard deviation of each Gaussian. So, for $R = 0$ the electrons will be close to each other, but as we increase R the separation between them becomes bigger and eventually, the electrons will dissociate. Coulomb cost is used since electrons in the atom repel each other.

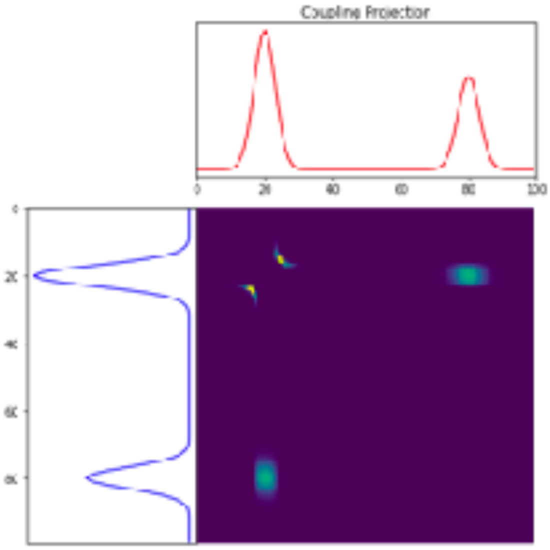

Figure 11.2: Optimal coupling matrix for one-body density

Solving the OT problem for the above marginal returns a lot of information about the interaction between the two atoms. The optimal coupling in Figure 11.2 shows that we can distinguish two types of solutions for the probability density depending on the region of interaction. The first solution in the top left-hand corner corresponds to the solution for the repulsive interaction between the electrons inside of the bigger Gaussian. The second solution, in the top right-hand corner and bottom left-hand corner, corresponds to the probability density associated with the attractive interaction between the electrons of one Gaussian and the nuclei of the atoms of the other Gaussian.

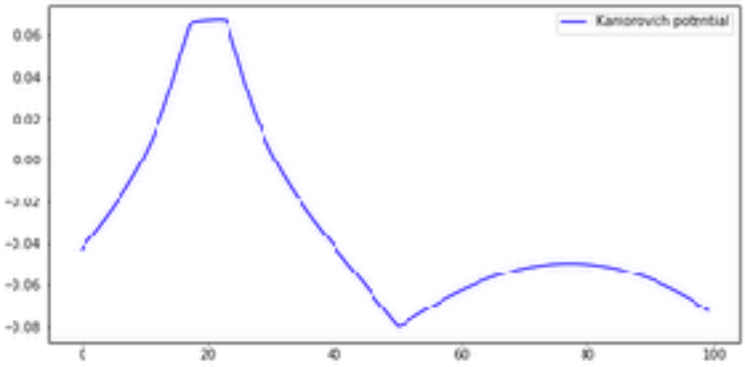

Figure 11.3: Kantorovich Potential generated by Sinkhorn

From the potential in Figure 11.3, three regions can be highlighted. For $0 < x < 75$ and For $125 < x < 200$ we have a repulsive potential that corresponds to the repulsion between the atoms inside of each Gaussian. The region $75 < x < 125$ is an attractive one, which shows us the interaction between the nuclei of the atoms in one Gaussian with the electrons in the other Gaussian. Figure 11.4 shows the optimal coupling for different values of R. For $R = 0$ the two Gaussians are together so we only have the probability distribution for N interacting electrons in one dimension. As we increase R the interaction, as can be seen for $R = 120$, that we have the four solutions, two repulsive and two attractive, with the same interpretation given before.

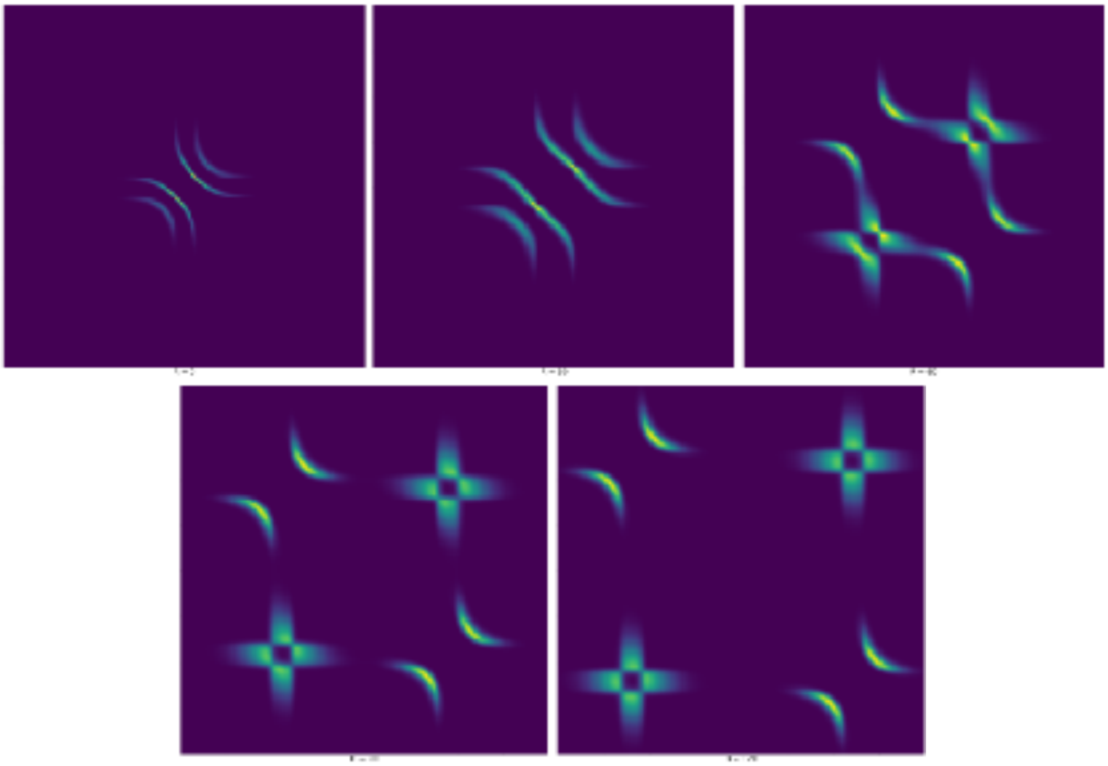

Figure 11.4: Evolution of optimal coupling as R increases

The OT cost of the problem gives the total potential energy of the system of electrons. Figure 11.5 plots the total potential energy of the system as a function of R. Initially, there is a strong repulsion between the atoms as $R \to 0$. Also, as R increases, a slight attraction can be observed. The value of the potential energy matches very closely with experimental results.

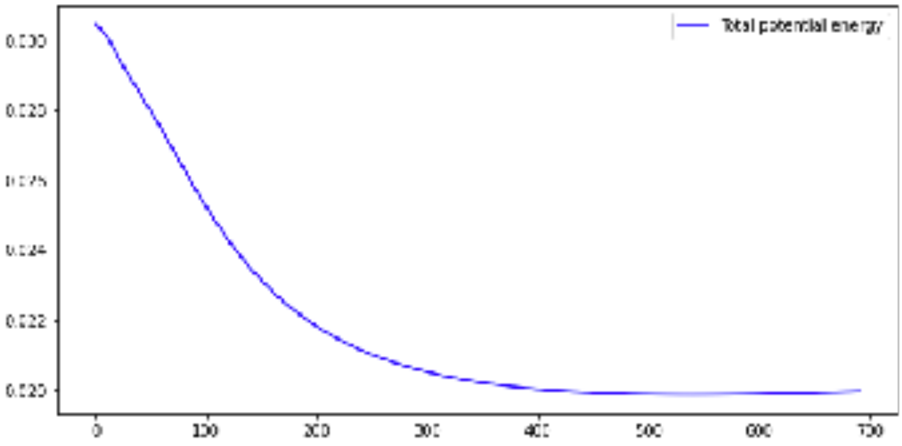

Figure 11.5: Potential energy as a function of R

The above examples dealt with Gaussian distributions. However, many other one-body densities exist in DFT. Some common examples include the Lorentzian density and Uniform density. OT can be used to study the couplings for any kind of one-body density. Figure 11.6 shows the optimal couplings for Lorentzian and Uniform densities. The coupling and potential match analytical solutions in DFT.

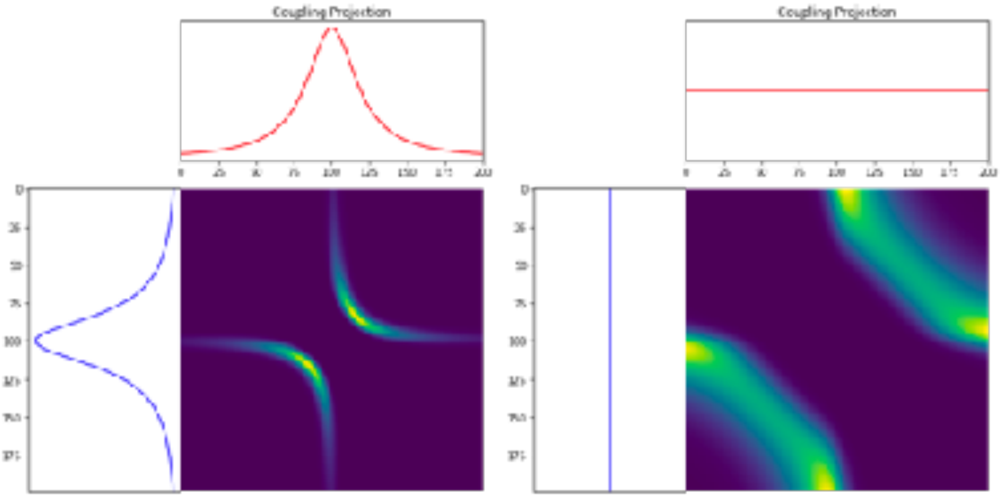

Figure 11.6: Optimal couplings for Lorentzian and Uniform densities

5.4.2 Wasserstein Distance

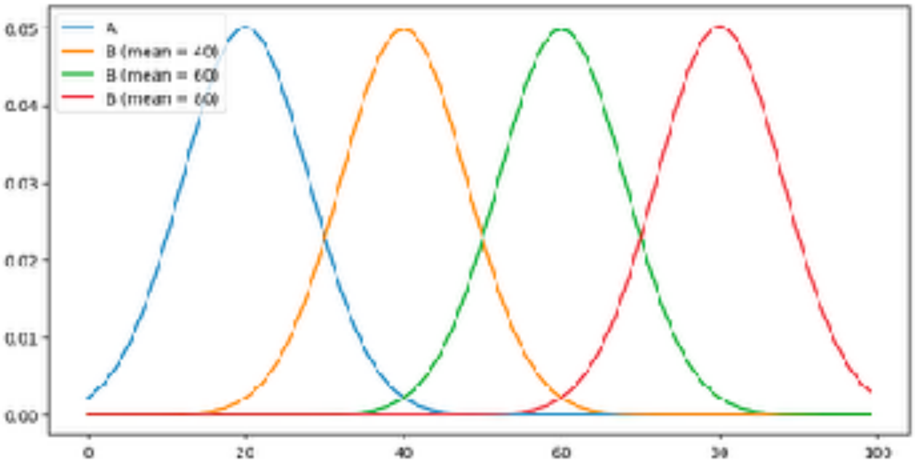

Figure 12.1: Probability distributions *a* and *b*

Distance measures are metrics which measure the similarity between probability distributions. Various distance measures focus on specific distributions' properties for comparison. The most popular distance measures are Euclidean distance and KL-Divergence. Euclidean distance is the length of the line segment between two points in Euclidean space. The distance between two probability distributions is the summation of distances between points in the distribution. The Euclidean distance between distributions *a* and *b* is given by:

$$d_E(a,b) = \sqrt{\int_{\mathbb{R}} |a(x) - b(x)|^2 dx} \approx \sqrt{\sum_{j=1}^{n} |a(x_j) - b(x_j)|^2}$$

The Kullback-Lelbler divergence (KL-divergence) is another distance metric to compare probability distributions. It is a non-symmetric distance measure. The formula for KL-divergence is:

$$d_{KL}(a||b) = \int_{-\infty}^{\infty} a(x)\log\left(\frac{a(x)}{b(x)}\right)dx \approx \sum_{x\epsilon X} a(x)\log\left(\frac{a(x)}{b(x)}\right)$$

The optimal transport cost, known as the Wasserstein distance is a valid distance measure and can be used to measure similarity/difference between two probability distributions. The p-Wasserstein distance between distributions *a* and *b* is:

$$W_p(a,b) = OT_C(a,b)^{\frac{1}{p}} = \left(\sum_{i,j} C_{ij} * P_{ij} \right)^{\frac{1}{p}}$$

Table 5 shows a comparison between the three distance measures discussed. The distribution a is fixed while b is translated to the right by a certain distance. While Euclidean or KL divergence is good for capturing the differences in shape between the two distributions, they do not prioritise the average distance between the distributions. The Wasserstein distance is a better choice in such scenarios as the OT problem focuses on the distances between the distributions.

Distance Measures	mean = 20	mean = 40	mean = 60	mean = 80
Euclidean	0.114	0.265	0.291	0.293
KL-Divergence	0.298	2.318	8.318	18.292
Wasserstein	4.542	20.622	40.312	59.553

Table 5: Comparison of distance measures

5.4.3 Wasserstein GANs

Wasserstein GANs are a variant of traditional GANs. A Generative Adversarial Network (GAN) is a system of neural networks that compete with each other and solve a min-max problem. It is a prominent method for generative AI. The system consists of two neural networks: (i) Generator and (ii) Discriminator. The generator generates an output from some random data. The discriminator decides if the data it receives is part of the dataset or generated by the other network. In short, the generator generates candidates while the discriminator network evaluates them.

The generator uses the output of the discriminator to make better candidate data. Since the discriminator performs binary classification, binary cross entropy (similar to KL-divergence) is the most popular loss function used to while training. However, as mentioned in the previous section, such distance measures may be limited in specific scenarios. Wasserstein GANs use the Wasserstein metric as the loss function instead of binary cross entropy. This variant of GANs improves learning stability and gives more helpful information for the generative network. Wasserstein loss makes the training more stable when the generator is learning distributions in high-dimensional spaces.

6. Conclusion and Future Work

The project presented an extensive overview of current Optimal Transport algorithms. Different types of algorithms, including deep learning solutions, were developed and tested on various inputs, such as probability distributions, binary grids, and images. Sinkhorn is a fast iterative algorithm that can even be extended for multiple marginals. However, its convergence highly depends on the initialisation. Moreover, the time and space complexity grows exponentially with the number of marginals and the size of the input. Neural Networks have the potential to solve complicated and high-dimensional OT problems. Neural solvers can be slower than traditional solvers in some cases. But they reach an approximate solution much quicker. Nevertheless, both solvers suffer from out-of-distribution generalisation since they cannot generate good predictions/solutions on instances that are not close to theory or training datasets.

While various algorithms have been developed for Optimal Transport, more work needs to be done for Multi-marginal OT. Current algorithms are limited by the space and computational requirements of such high-dimensional problems. Future work in the field must develop more deep learning solutions like Amortised algorithms for multi-marginal OT. Finally, applications of OT in different domains must also be studied to advance both fields. Nature always chooses the most optimal path, and studying this core idea is crucial for understanding the world around us.

7. References

(1) B. Amos, S. Cohen, G. Luise, and I. Redko, ‘Meta Optimal Transport’. arXiv, 2022.

(2) A. Korotin, D. Selikhanovych, and E. Burnaev, ‘Neural Optimal Transport’. arXiv, 2022.

(3) A. V. Makkuva, A. Taghvaei, S. Oh, and J. D. Lee, ‘Optimal transport mapping via input convex neural networks.’ arXiv, 2019.

(4) A. Korotin, L. Li, A. Genevay, J. Solomon, A. Filippov, and E. Burnaev, ‘Do Neural Optimal Transport Solvers Work? A Continuous Wasserstein-2 Benchmark’. arXiv, 2021.

(5) B. Pass, ‘Multi-marginal optimal transport: theory and applications’. arXiv, 2014.

(6) M. Arjovsky, S. Chintala, and L. Bottou, ‘Wasserstein GAN’. arXiv, 2017.

(7) Amos, B. (2022) ‘Tutorial on amortized optimization for learning to optimize over continuous domains’. arXiv. doi: 10.48550/ARXIV.2202.00665.

(8) Korotin, A. *et al.* (2019) ‘Wasserstein-2 Generative Networks’. arXiv. doi: 10.48550/ARXIV.1909.13082.

(9) Rout, L., Korotin, A. and Burnaev, E. (2021) ‘Generative Modeling with Optimal Transport Maps’. arXiv. doi: 10.48550/ARXIV.2110.02999.

(10) Seguy, V. *et al.* (2017) ‘Large-Scale Optimal Transport and Mapping Estimation’. arXiv. doi: 10.48550/ARXIV.1711.02283.

(11) Lu, G. *et al.* (2020) ‘Large-Scale Optimal Transport via Adversarial Training with Cycle-Consistency’. arXiv. doi: 10.48550/ARXIV.2003.06635.

(12) Daniels, M., Maunu, T. and Hand, P. (2021) ‘Score-based Generative Neural Networks for Large-Scale Optimal Transport’. arXiv. doi: 10.48550/ARXIV.2110.03237.

(13) Eckstein, S. and Kupper, M. (2018) ‘Computation of optimal transport and related hedging problems via penalization and neural networks’. arXiv. doi: 10.48550/ARXIV.1802.08539.

(14) G. Peyre and M. Cuturi, ‘Computational Optimal Transport’, Foundations and Trends in Machine Learning, vol. 11, no. 5–6, pp. 355–607, 2019.

(15) Cuturi, M., & Doucet, A. (2014). Fast Computation of Wasserstein Barycenters. arXiv. doi: 1310.4375

(16) Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, et al.. Convolutional wasserstein distances. ACM Transactions on Graphics, 2015, 34 (4), pp.66:1-66:11.10.1145/2766963. hal-01188953

(17) Oh JH, Pouryahya M, Iyer A, Apte AP, Deasy JO, Tannenbaum A. A novel kernel Wasserstein distance on Gaussian measures: An application of identifying dental artifacts in head and neck computed tomography. Comput Biol Med. 2020 May;120:103731. doi: 10.1016/j.compbiomed.2020.103731.

(18) Ponti A, Giordani I, Mistri M, Candelieri A, Archetti F. The "Unreasonable" Effectiveness of the Wasserstein Distance in Analyzing Key Performance Indicators of a Network of Stores. *Big Data and Cognitive Computing*. 2022; 6(4):138.

(19) Iacobelli, M. A New Perspective on Wasserstein Distances for Kinetic Problems. *Arch Rational Mech Anal* **244**, 27–50 (2022). https://doi.org/10.1007/s00205-021-01705-9z

(20) Patrini, G., van den Berg, R., Forré, P., Carioni, M., Bhargav, S., Welling, M., Genewein, T., & Nielsen, F. (2019). Sinkhorn AutoEncoders.

(21) Villani, C. (2003) *Topics in Optimal Transportation*. American Mathematical Society (Graduate studies in mathematics). Available at: https://books.google.co.in/books?id=R_nWqjq89oEC.

(22) Buttazzo, G., Pascale, L. D., & Gori-Giorgi, P. (2012). Optimal-transport formulation of electronic density-functional theory. Physical Review A, 85(6). https://doi.org/10.1103/physreva.85.062502

(23) Amortised Optimal Transport: https://ruishu.io/2017/11/07/amortized-optimization/

(24) Python OT Library Documentation: https://pythonot.github.io/index.html

(25) Inverse Transport Sampling Guide: https://stephens999.github.io/fiveMinuteStats/inverse_transform_sampling.html

APPENDIX A

Hardware and Software Specifications

All the code is run on Google Colab, a cloud-platform which provides GPU for computing. The hardware specifications include:

- GPU: 1xTesla K80, compute 3.7, having 2496 CUDA cores, 12GB GDDR5 VRAM
- CPU: 1xCore hyper-threaded Xeon Processors @2.3Ghz
- RAM: ~12.6 GB Available
- Disk: ~33 GB Available

The programs were written in Python and ran on Jupyter Notebooks. Various third-party libraries are utilised. Some of them include NumPy is used to handle operations on n-d arrays; MatPlotLib for plotting data; Seaborn to interpolate data; and TensorFlow for building and training neural networks.

Hyper-parameters

The tables below briefly summaries the hyper-parameter used for training all the OT solvers.

Name	Value
Batch Size	64
Number of Epochs	7000
Adam learning rates	0.005; 0.001
Weight Decay	10^{-7}
MLP Hidden Sizes	[100,100]

Table 7: Generative Modelling Hyper-parameters

Name	Value
Input Size	(784,1)
Batch Size	128
Number of Epochs	50000
Adam learning rate	0.001
MLP Hidden Sizes	[1024,1024,1024]

Table 6: Amortised OT Hyper-parameters

APPENDIX B

Multi-marginal Optimal Transport with Entropic Regularisation

Let n be the number of marginals and s be the size of 1D probability distributions. Then, the matrix A of size nfs such that each row a_i is a 1D probability distribution of size s:

$$\sum_{j=0}^{s} A_{ij} = 1 \forall i \in \mathbb{N}_n$$

The cost tensor C and coupling tensor P is a tensor of size s^n. The optimal transport problem is defined as:

$$OT_C(A) := \min \langle C, P \rangle + \epsilon H(P)$$

$H(P)$ is the discrete entropy for a coupling tensor with 3 marginals is defined as:

$$H(P) := -\sum_{ijk} P_{ijk} \left(\log P_{ijk} - 1 \right)$$

For the above optimal transport problem, the solution is unique and has the form:

$$P = K \odot u_1 \odot u_2 \odot \cdots \odot u_{n-1} \odot u_n$$

where u_i are scaling vectors. These are stored row-wise in the matrix U. The kernel associated with the cost tensor is defined as:

$$K := e^{\frac{-C}{\epsilon}}$$

Subject to the constraints,

$$\sum_{\mathbb{N}_n - \{i\}} P = a_i \forall i \in \mathbb{N}_n$$

Then the iterative sinkhorn algorithm is:

1. Initialise u_i vectors.
2. To update u_i,

    2.1.$P = P \odot 1/u_i = K \odot u_1 \odot \cdots \odot u_{i-1} \odot u_{i+1} \cdots \odot u_{n-1} \odot u_n$

```
        2.2.$u_i = \frac{a_i}{\sum_{\mathbb{N}_n - \{i\}} P}$
        2.3.$P = P \odot u_i$
3. Update all $u_i$'s until convergence
4. Output $P$
```

Where $\odot$ is defined as the broadcasting operator which multiplies a vector with a tensor along a specific direction.

APPENDIX C

Multi-marginal Optimal Transport: Optimal Couplings

The figure below shows the 2D and 3D projections of the optimal couplings generated by the Multi-marginal Sinkhorn algorithm. All the inputs marginals are gaussian distributions with size 50 with mean at the centre (25).

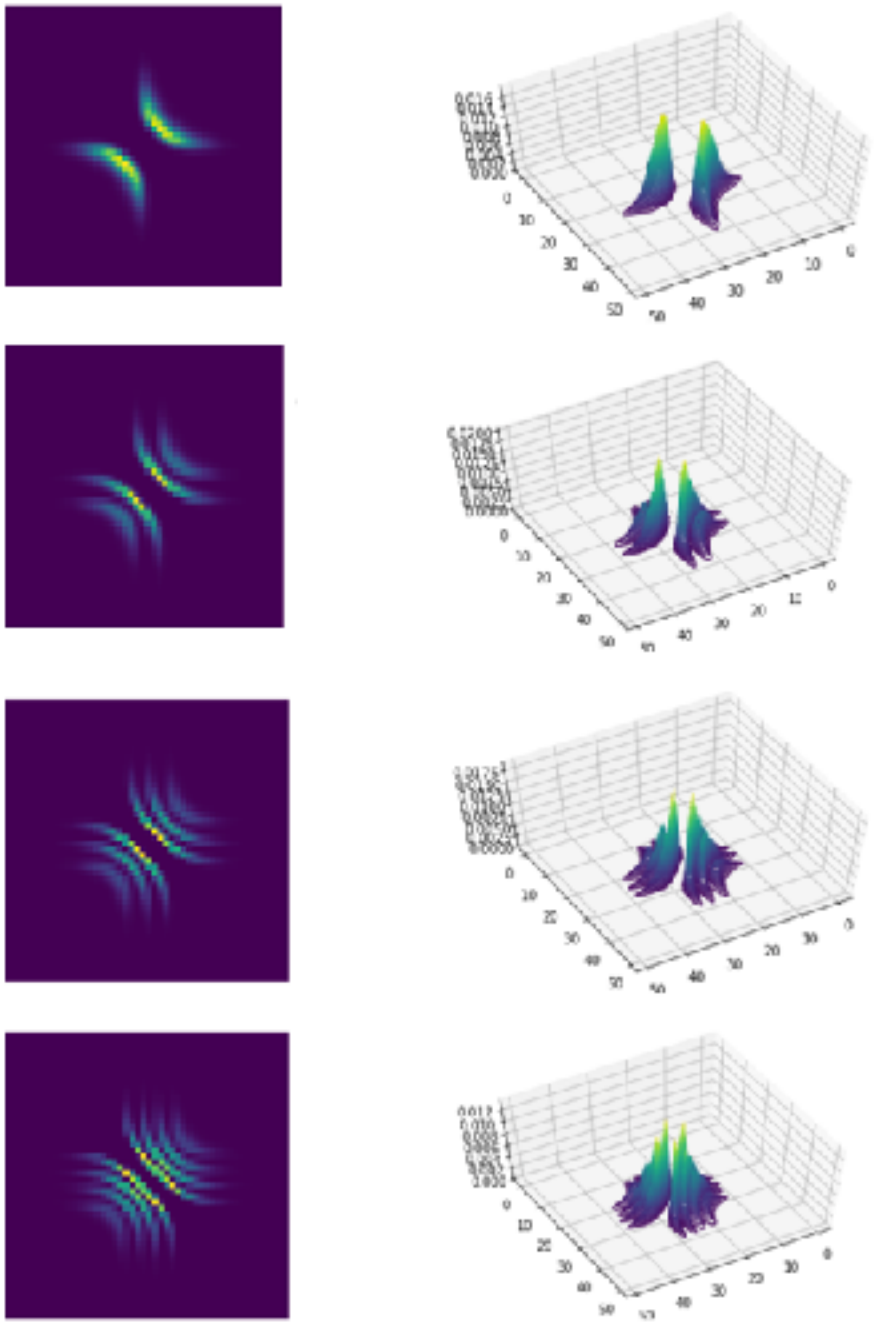

Figure 13.1: Multi-marginal Optimal Transport Couplings