Research Project Report

Gromov-Wassertein regularization and translation across latent spaces

by

Mathieu Chevalley

mchevalley@student.ethz.ch

Supervisor: Charlotte Bunne

Learning and Adaptive Systems Group

Report for the course
Research in Data Science
at
ETH Zurich

Abstract

In many critical tasks, we are interested in learning models that are able to translate between two different domains (for example, from one's genotype to a specific medical treatment). These models try to learn a joint distribution between the two domains given only unpaired samples from the individual domains. Successful models usually use combinations of VAE's and GAN's, adding several inductive biases to arrive at meaningful pairings, such as enforcing cycle-consistency [Gro+19; Zhu+17]. They also usually rely on the learning of a shared latent space, such as the UNIT framework [LBK17], where two Auto-Encoders (one per domain) share a latent space and are learned against an adversary.

In this project, motivated by previous work on the use of the Gromov-Wasserstein distance as a loss function in generative models [Bun+19], we implement a new framework using this distance in a model similar to UNIT. Instead of relying on a shared latent space, our framework keeps two separate latent spaces, one per domain, while minimizing their GW distance. This framework gives more flexibility to the two encoders while still enforcing latent spaces with similar structures. We first study the benefit of the GW regularization on the representation learning, and find that in some cases, the latent spaces learned by the two Auto-Encoders are better than when they are learned separately. We then attempt to translate between the two latent spaces by using the coupling matrix associated to the GW distance. We find that the alignments found by the coupling matrix are meaningful and may be used to train a translation network.

Contents

1 Introduction 1

2 Background 3
- 2.1 Optimal Transport . . . 3
  - 2.1.1 Gromov-Wasserstein . . . 4
- 2.2 Generative Adversarial Network . . . 5
- 2.3 Variational Auto-Encoder . . . 6
- 2.4 Unsupervised Domain Translation . . . 8

3 Proposed Model 11
- 3.1 Architecture and training . . . 11
- 3.2 Loss . . . 11
- 3.3 Translation . . . 12

4 Results 15
- 4.1 Gromov-Wasserstein Regularization . . . 15
  - 4.1.1 Visual assessment of latent spaces . . . 15
  - 4.1.2 Classification on learned latent representations . . . 15
- 4.2 Translation . . . 16

5 Conclusion 25

Acronyms 27

Bibliography 29

1 Introduction

Given the abundance of collected data available, machine learning techniques have demonstrated impressive capabilities at extracting valuable information from it. Unfortunately, most successful methods rely on large datasets that have been labeled or paired. This labeling or pairing of the dataset is a time and resource consuming task. Furthermore, it is prone to human bias and error. New *unsupervised* methods and models thus needs to be developed. Thanks to the recent breakthroughs in generative modeling, a new area called Unsupervised Domain Translation (UDT) has emerged. Given two unpaired datasets from distinct domains, UDT consists in finding a *meaningful* mapping from one domain to the other. For example, if one domain consists of color images and the other one of gray-scale images, a successful model should be able to *translate* an image from gray-scale to color, even though the corresponding color image does not exist in the dataset of color images.

Theoretically, UDT consists in inferring a joint probability distribution from two datasets drawn from the respective marginal distributions. Since the set of joint distributions that agree with the two marginal distributions is infinite, some inductive bias needs to be enforced on the modeled joint distribution to arrive at meaningful pairings. This ill-posed nature of the UDT problem partly explains why most successful UDT models to date have applications in computer vision, as it is easier to assess and infer new inductive biases in perceptual tasks. These models manage to solve the *image-to-image translation* problem, which is a subset of UDT, by enforcing *cycle-consistency* (see section 2.4) [Gro+19; Zhu+17] or other inductive bias.

Unsupervised Domain Translation, given its broad definition, encompasses a large set of tasks, and thus many specialized tasks can be recast as a UDT problem, such as: text translation, text-to-speech, any types of image-to-image translation (gray-scale to color, low to high resolution, night time to day time), but also more critical applications such as mapping a human genotype to a tailored drug composition. Unsupervised Domain Translation can also be used to solve the *domain adaptation* problem, where the goal is to adapt a classifier trained on a labeled source domain and use it to classify an unlabeled target domain. Domain adaptation can be approached by first translating a sample from the target domain to the source domain, and then pass it trough a classifier trained on the source domain.

In this project, we propose a new UDT framework for translation. Taking inspiration from the UNsupervised Image-to-image Translation (UNIT) framework [Zhu+17] and motivated by previous work on the use of the Gromov-Wasserstein distance as a loss function in generative models [Bun+19], we jointly train two Auto-Encoders, one per domain, and add a Gromov-Wasserstein (GW) loss between the two latent spaces, so as to minimize their GW distance. We use the coupling associated to the GW distance to translate between the two latent spaces. By keeping the two latent spaces separated (contrary to UNIT), we expect our framework to give more flexibility and modeling capability while still enforcing meaningful pairings. We study the effect of adding this GW regularization and explore the translation capabilities of our framework.

2 Background

In this chapter, we review the necessary mathematical background, mainly Optimal Transport (OT) theory, as well as the related work – generative modeling techniques and Unsupervised Domain Translation (UDT) models.

2.1 Optimal Transport

Optimal Transport has recently gained traction in machine learning research, especially for generative modeling. It is particularly interesting as it defines multiple distances between distributions, where the two distributions may have non-overlapping supports or even supports in different spaces. Indeed, some common distances like the Kullback-Leibler divergence become infinite or undefined when the data is supported in low-dimensional manifolds, whereas OT divergences do not suffer from this problem [AB17].

We present the main formulations of the OT problem and some distances – notably the Gromov-Wasserstein (GW) distance – following the notations of [PC19]. We concentrate on distances defined on discrete distributions.

Originally, OT can be described as the problem of transporting the mass from a set of points $\{x_1, \dots, x_n\}$ to a set of destination points $\{y_1, \dots, y_m\}$ that have finite capacities. Furthermore, the cost of transporting from a point to another is fixed by a cost matrix. There is obviously a natural analogy to logistics and planning. The OT problem thus is to fulfill this mass transport while minimizing the cost. To recast this as a distance between distribution, the mass of the source points is now described by an histogram $\mathbf{a} \in \{\mathbb{R}_+^n : \sum_i \mathbf{a}_i = 1\}$ and the destination capacities as an histogram $\mathbf{b} \in \{\mathbb{R}_+^m : \sum_i \mathbf{b}_i = 1\}$. The cost matrix is $\mathbf{C} \in \mathbb{R}_+^{n \times m}$ where $\mathbf{C}_{ij}$ describe the cost of transport from x_i to y_j. The Kantorovich formulation of this problem is:

$$\mathrm{L}_{\mathbf{C}}(\mathbf{a}, \mathbf{b}) := \min_{T \in \mathbf{U}(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, T \rangle := \sum_{i,j} \mathbf{C}_{i,j} T_{i,j} \tag{2.1}$$

where

$$\mathbf{U}(\mathbf{a}, \mathbf{b}) := \left\{ T \in \mathbb{R}_+^{n \times m} : T \mathbb{1}_m = \mathbf{a}, T^T \mathbb{1}_n = \mathbf{b} \right\} \tag{2.2}$$

$T_{i,j}$ describes how much mass flows from x_i to y_j. The constrain imposes that all the mass leaves $\mathbf{a}$ and that $\mathbf{b}$ is *filled*. Given the cost matrix, we hence can compute the divergence between two distributions, and this divergence elegantly comes with an exact description (the coupling matrix T) of how to go from the first configuration to the other. Furthermore, if $x_i, y_i \in \mathcal{X}$ lie in the same metric space with distance d and the cost matrix satisfies $\mathbf{C}_{i,j} = d^p(x_i, y_j)$ for some $p \geq 1$, then:

$$\mathrm{W}_p(\mathbf{a}, \mathbf{b}) := \mathrm{L}_{\mathbf{C}}(\mathbf{a}, \mathbf{b})^{1/p} \tag{2.3}$$

is a *distance*, called the p-Wasserstein distance. This distance has had a large range of applications, notably in generative models (see section 2.2 and section 2.3). Unfortunately, this distance requires that the points x_i and y_i lie in the same metric space. To alleviate this, we present the Gromov-Wasserstein (GW) distance [Mém11], which only requires a distance defined among the points in each respective space, that is, intra-spaces distance matrices.

2.1.1 Gromov-Wasserstein

Let $n = m$ and $D, D' \in \mathbb{R}_+^{n \times n}$ be two distance matrices, where $D_{i,j} = \mathrm{dist}(x_i, x_j)$ and $D'_{i,j} = \mathrm{dist}(y_i, y_j)$. Given $\mathbf{a}, \mathbf{b}$ and those two matrices, the GW discrepancy is defined as:

$$\mathrm{GW}(D, D', \mathbf{a}, \mathbf{b}) := \min_{T \in \mathbf{U}(\mathbf{a},\mathbf{b})} \mathcal{E}_{D,D'}(T) \tag{2.4a}$$

$$:= \min_{T \in \mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{i,j,k,l} |D_{ij} - D'_{kl}|^2 T_{ij} T_{kl} \tag{2.4b}$$

The main advantage of this distance is that it does not require any distance to be defined between the two spaces, making it more flexible and applicable to a large range of problems. Unfortunately, this objective becomes intractable to solve for large n. This shortcoming can be alleviated by adding a regularization term, which makes this new smoothed objective approximable through an iterative algorithm.

Regularized Gromov-Wasserstein

As proposed by [PCS16], GW can be regularized by adding an entropy term:

$$\mathrm{GW}_\epsilon(D, D', \mathbf{a}, \mathbf{b}) := \min_{T \in \mathbf{U}(\mathbf{a},\mathbf{b})} \mathcal{E}_{D,D'}(T) - \epsilon \mathbf{H}(T) \tag{2.5}$$

$$\mathbf{H}(T) := -\sum_{ij} T_{ij}(\log(T_{ij}) - 1) \tag{2.6}$$

where T is the coupling matrix, $\mathbf{H}(T)$ is the entropy of this coupling and ϵ controls the strength of the regularization. The entropy regularization forces the solution T^* to be sparser, which is convenient as we may be interested in alignments that are *softer*. This new objective is also easier to optimize, can be approximated using the Sinkhorn iterative algorithm, and can also be used as a loss function.

However, GW_ϵ is not a distance anymore because of the regularization term. To remedy this, [Bun+19] propose to normalize it, such that it is a distance again. This distance reads:

$$\overline{\mathrm{GW}}_\epsilon(D, D', \mathbf{a}, \mathbf{b}) = \\ 2 \times \mathrm{GW}_\epsilon(D, D', \mathbf{a}, \mathbf{b}) - \mathrm{GW}_\epsilon(D, D, \mathbf{a}, \mathbf{a}) - \mathrm{GW}_\epsilon(D', D', \mathbf{b}, \mathbf{b}) \quad (2.7)$$

As [Bun+19] demonstrate, this regularized and normalized GW distance can successfully be used as a loss term, especially in generative models, as a way to align two distributions. This is the loss that we use in this project and in the subsequent sections of this report, we refer to it as the GW LOSS. The GW LOSS can be thought as enforcing both distributions to have *similarly* structured supports, and we conjecture that the coupling matrix may be able to match the modes of the two distributions.

2.2 Generative Adversarial Network

Generative Adversarial Network (GAN), which were first introduced in [Goo+14], is an implicit generative model. Based on game theory, it can intuitively be described as a two player game, where each player is parameterized by a neural network. In this game, a Generator tries to map some random noise given as input to samples similar to the target dataset, i.e it transforms a simple distribution to a more complex one. We call it the generated distribution and denote it by p_g. On the other hand, a Discriminator tries to distinguish between samples coming from the target dataset and samples produced by the Generator. At convergence, the Generator produces data that is distributed similarly to the target dataset, and thus it becomes impossible for the Discriminator to distinguish real and synthetic samples. See Figure 2.1 as an example of the training of a GAN on the MNIST dataset.

Formally, the objective of the two-player minimax game reads:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \quad (2.8)$$

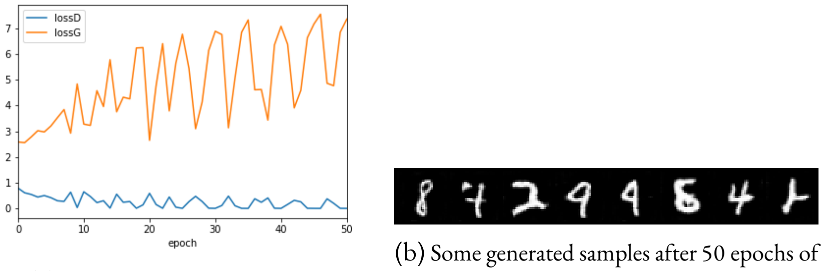
where $\mathbf{z}$ is the input noise, $\mathbf{x}$ comes from the target distributions, and the Discriminator D should output 1 when its input is a real samples, and 0 otherwise. If the Discriminator is optimal for a given G, Equation 2.8 can be rewritten to show that the Generator actually minimizes the Jensen–Shannon divergence (JSD) between the generated and target distribution. It also can be shown that if both networks have sufficient capacity, and if the Discriminator is trained to optimality after each optimization step of the Generator, then the distribution of the Generator converges to the target distribution. In practice, however, GANs have been observed to be hard and unstable to train, as well as suffering from mode collapse. Mode collapse happens when the Generator models only part of the distribution, i.e only a subset of the support. Many models similar to GAN have been proposed so as to try to remedy these limitations. In the following, we present a few of them, which are based on the OT theory.

One of the shortcoming of the GAN is that it minimizes the JSD, which is non-continuous when the two distributions have non-overlapping supports. To remedy this, [ACB17] propose the Wasserstein GAN, which relies on the 1-Wasserstein distance (see section 2.1). Contrary to

JSD, this distance, also called the Earth Mover distance, is continuous and thus provide a more useful gradient. As directly minimizing the 1-Wasserstein distance is intractable, WGAN actually minimizes the dual formulation of this distance. The objective is as follows:

$$W(p_{data}, p_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim p_{data}}[f(x)] - \mathbb{E}_{x \sim p_g}[f(x)] \tag{2.9}$$

where p_g is distributed as $G(z), z \sim p_z(z)$. Here, the Discriminator can be any function that is 1-Lipschitz. Arjovsky, Chintala, and Bottou [ACB17] thus generalizes the idea of the GAN to the minimization of any distance between the target and generated distribution. Following this insight, different models that attempt to approximately minimize OT distances have been proposed [Bun+19; GPC18; Sal+18]. In these models, the Generator directly minimizes an approximation of the distance and the Discriminator, which is often optional, adversarially learns a latent representations of the data that is used to compute the cost matrix (see section 2.1).



(a) Evolution of the training loss

(b) Some generated samples after 50 epochs of training

Figure 2.1: Training of a GAN on the MNIST dataset

2.3 Variational Auto-Encoder

Variational Auto-Encoder (VAE), which were first introduced in [KW13], is another type of generative model. Contrary to GANs, it is an explicit likelihood based model that directly maximizes the likelihood of the samples in the dataset. It assumes the existence of a latent variable that controls the generative process. The generative model thus is $p_\theta(z)p_\theta(x|z)$, where $p_\theta(z)$ is a prior on the latent variable. As the posterior is intractable, it is approximated by $q_\phi(z|x)$, parameterized by ϕ. Both $p_\theta(x|z)$ and $q_\phi(z|x)$ are represented by neural networks. As computing the likelihood is also intractable, Kingma and Welling [KW13] introduce the Evidence Lower Bound (ELBO), which is a lower bound to the data likelihood:

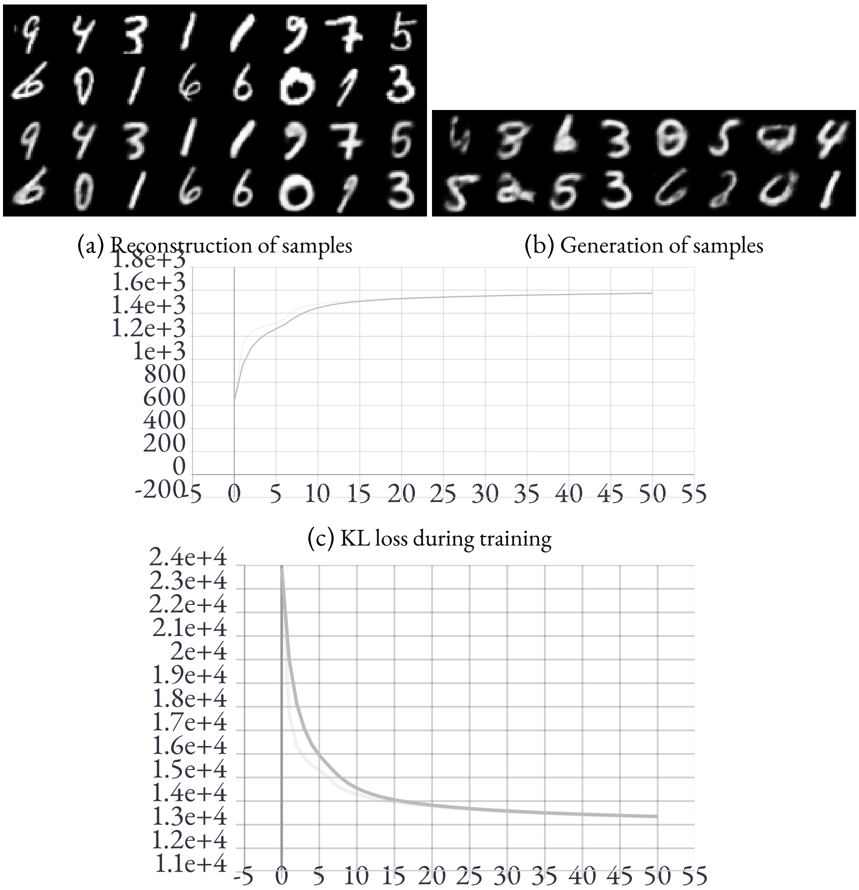
$$\mathcal{L}(\theta, \phi; x) = -D_{KL}(q_\phi(z|x) \| p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \tag{2.10}$$

where the Kullback-Leibler (KL) divergence term can be seen as a regularizer on the posterior and the second term as a reconstruction loss. VAEs are similar to Auto-Encoders (AEs), as $q_\phi(z|x)$ and $p_\theta(x|z)$ are essentially an encoder and a decoder respectively. VAEs can be used as generative models simply by decoding samples from the prior distribution. Unfortunately, simple likelihood models like VAEs have been observed to produce samples of lesser quality compared to GANs.

In [Tol+18], Tolstikhin, Bousquet, Gelly, and Schoelkopf generalize the idea of the VAE and provide a more flixible framework based on OT theory. Starting from the 1-Wassertein distance, they propose the Wasserstein Auto-Encoder (WAE) objective:

$$D_{WAE}(p_{data}, p_g) = \inf_{q(z|x)} \mathbb{E}_{p_x} \mathbb{E}_{q(z|x)}[c(x, G(z))] + \lambda \cdot \mathcal{D}(q(z), p(z)) \tag{2.11}$$

where the first term is the reconstruction loss, which can use any cost function c, and the second term is the regularizer, which can be any divergence. In the paper, they propose two choices for $\mathcal{D}$: the JSD implemented by a GAN, and the Maximum Mean Discrepancy (MMD). These two models are called WAE-GAN and WAE-MMD respectively.



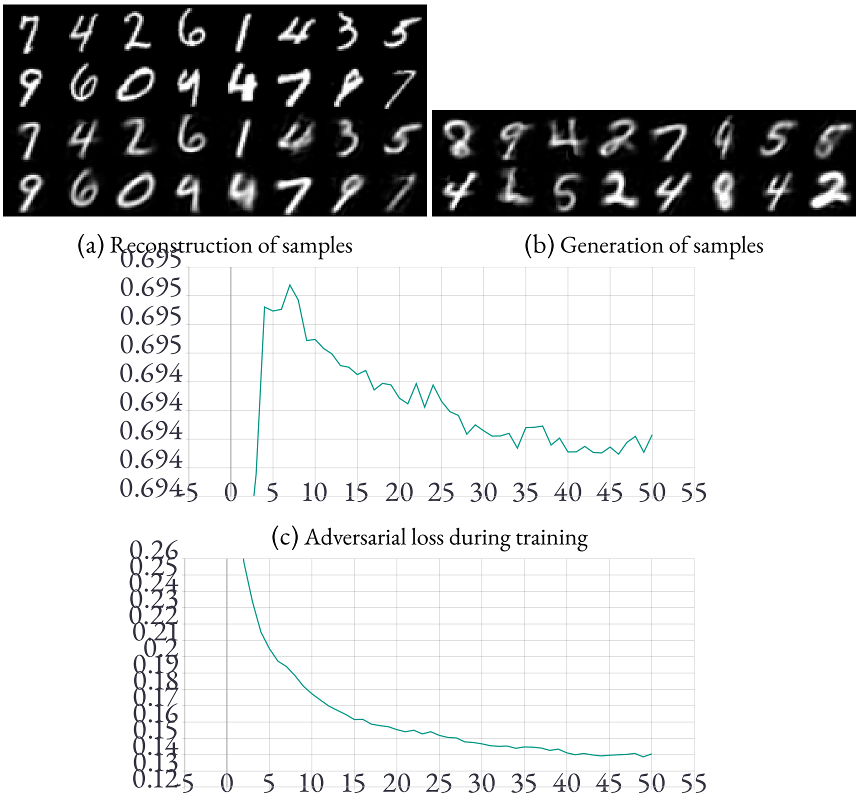
(a) Reconstruction of samples

(b) Generation of samples

(c) KL loss during training

(d) Reconstruction loss during training

Figure 2.2: Training of a VAE for 50 epochs on the MNIST dataset



(a) Reconstruction of samples

(b) Generation of samples

(c) Adversarial loss during training

(d) Reconstruction loss during training

Figure 2.3: Training of a WAE-GAN for 50 epochs on the MNIST dataset

2.4 Unsupervised Domain Translation

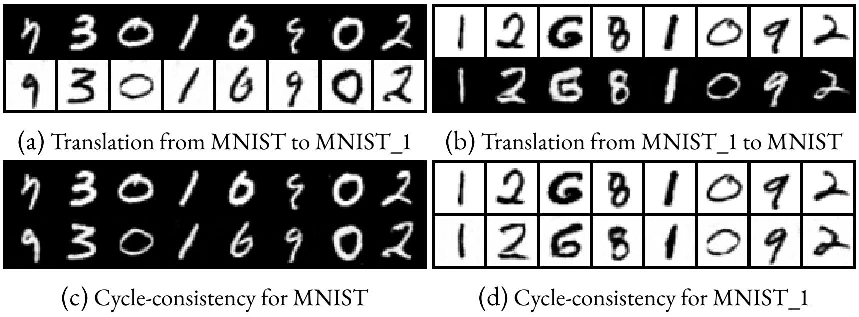
Unsupervised Domain Translation (UDT) can be described as the problem of finding meaningful pairing between two domain. Unfortunately, having datasets of paired samples from the two domains is a very costly task. We are thus interested in having mechanisms that find those pairings without supervision. Theoretically, this consists in finding a joint distribution given both marginal distributions, which is an under-constrained problem as there is an infinite set of joint distributions that are consistent with the marginal distributions. The set of possible solutions thus needs to be reduced by adding constrains and inductive biases.

In the seminal work of CycleGan [Zhu+17], the main inductive bias introduced is *cycle-consistency*. The cycle-consistency constraint imposes that if an image from domain A is translated to domain B, and then translated back to domain A, it should result in an image *identical* to the original one. The model consists of two translation networks, mapping images from domain A to B and *vice-versa*. The cycle-consistency constrain is enforced by adding the following loss:

$$\mathcal{L}_{cyc}(G,F) = \mathbb{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1] \tag{2.12}$$

where $p_{data}(x)$ and $p_{data}(y)$ are the marginal distributions of the two domains and G and F are the two translation networks.

Building on the idea of cycle-consistency, UNsupervised Image-to-image Translation (UNIT) [LBK17] additionally assume the existence of a shared latent representation for the two domains. To achieve that, the translation networks G and F now are two VAEs with shared latent spaces. Lastly, UNIT imposes that the last layers of the two encoders and the first layers of the two decoders have shared weights. See Figure 2.4 for a reproduction of the UNIT framework on the MNIST and MNIST inverted (denoted MNIST_1) datasets.



(a) Translation from MNIST to MNIST_1

(b) Translation from MNIST_1 to MNIST

(c) Cycle-consistency for MNIST

(d) Cycle-consistency for MNIST_1

Figure 2.4: UNIT model trained on MNIST and MNIST_1 (MNIST with inverted colors)

Using normalizing flows, AlignFlow [Gro+19] proposes a translation model that guarantees *exact* cycle-consistency, by taking advantage of the invertibility of normalizing flows. The architecture is similar to UNIT, with a shared latent representation and shared weights in the decoders, the only difference being that each encoder now is the exact inverse function of the corresponding decoder.

Lastly, in [BAG19], Bézenac, Ayed, and Gallinari develop an analysis and description of the UDT problem from the perspective of Optimal Transport. First, they argue that past method such as CycleGan are actually biased towards small transformations, which is why it works well mainly when the two distributions are close. They reformulate the problem as an OT transportation cost minimization problem. Using the dynamical formulation of OT, they are able to provide a link with NN models used for OT, indicating that residual layers used in CycleGan implicitly minimize a transportation cost.

3 Proposed Model

In this chapter, we present the setup of our experiments. For all models tested, we use the same architecture and add the GW loss in a modular way. We also present a first attempt at translating between the two domains.

3.1 Architecture and training

For all the models, we use the same architecture for the encoders and for the decoders. To keep things simple and to avoid introducing bias with the architecture, we use simple multi-layer perceptrons, which consists of linear layers with ReLu activations. We test three different dimensions for the latent representation: 2, 4, and 8.

We train the models on the MNIST and MNIST_1 (i.e, MNIST with black and white inverted) for 50 epochs with a batch size of 100, using the Adam [KB14] optimizer with `beta1` = 0.5, `beta2` = 0.99 and a learning rate of 0.001. We apply a standard normalization to the images. For the reconstruction loss, we use the Mean Square Error (MSE) loss. For the WAE, we train both WAE-GAN and WAE-MMD, i.e a regularization loss that is an adverserial discriminator and the MMD loss respectively. We also train simple AEs, i.e with only a reconstruction loss. See Figure 3.1 for an overview of the model.

3.2 Loss

At training time, we introduce a GW loss Equation 2.7 between batches of latent samples from both domains and minimizes it. For each batch, we compute the $\overline{\mathrm{GW}_\epsilon}$ with $\mathbf{a} = \mathbf{b} = \mathbb{1}_n/n$, where n is the number of samples in a batch, D is the matrix of pairwise Euclidean distances between the latent codes in domain A, and D' the matrix of pairwise Euclidean distances between the latent codes in domain B. The loss of the entire model with both Auto-Encoders thus become:

$$\mathcal{L}_{cotraining} = \mathcal{L}(x_1; \theta_1) + \mathcal{L}(x_2; \theta_2) + \lambda \cdot \overline{\mathrm{GW}_\epsilon}(\mathbf{a}, \mathbf{b}, D, D'; \theta_1, \theta_2) \tag{3.1}$$

where x_1 and x_2 are batches from domain A and B respectively, $\mathcal{L}$ the loss for the single Auto-encoder, λ an hyperparameter controlling the strength of GW loss minimization, and θ_1 and θ_2 the network parameters of the first and second Auto-encoder. We use $\epsilon = 0.005$ for the $\overline{\mathrm{GW}_\epsilon}$ loss. To choose the hyperparameter λ, we run the model with different values going from 1 to 1000, and then manually choose the best one for each model by inspecting the latent spaces and their t-SNEs.

By adding this GW loss between the two latent spaces, we want to enforce them to be *similar*, which should make downstream tasks such as translation and alignment easier. Intuitively,

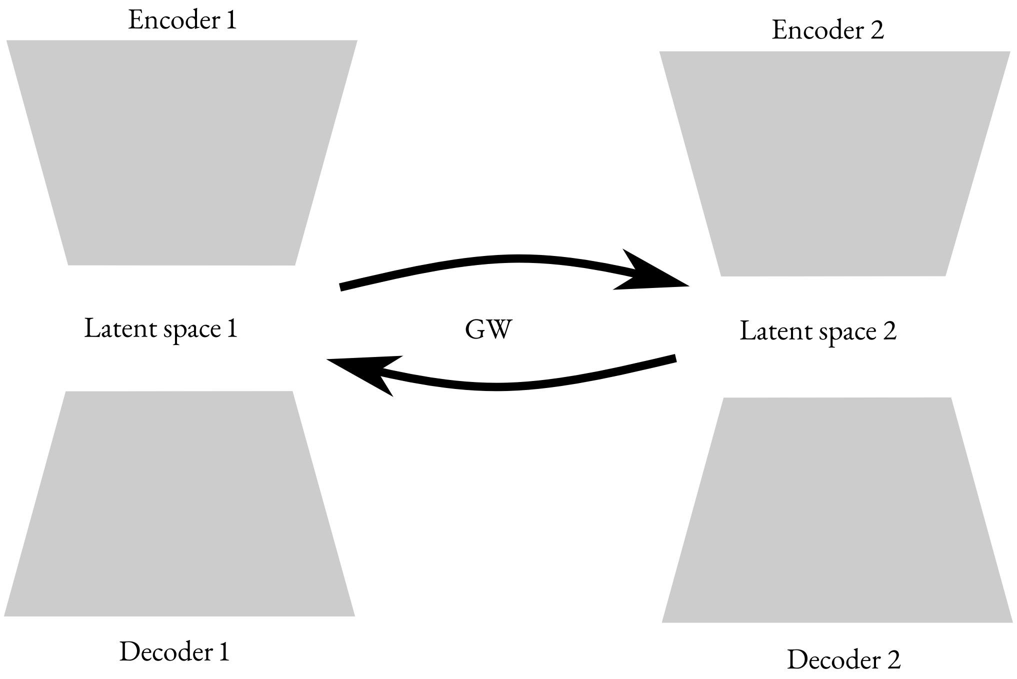


Figure 3.1: General architecture of the model, with two Auto-encoders, one for each domain. The GW loss acts on both latent spaces for the cotraining. We translate between domains across the latent spaces.

the GW loss is low when the points in the two spaces have similar distributions, and thus by minimizing this loss, the two latent spaces should converge to having similar structures. Concurrently with the learning of a latent representation that allow for a good reconstruction, we should then obtain latent spaces that have clearer separations between the different classes. Both Auto-encoders could also individually benefit from the cotraining as learned structures could be shared across the the two encoders through the GW loss and its gradient. This information sharing between the two encoders could result in the learning of better representations for each domain individually than if the Auto-encoders were learned separately. Finally, one additional advantage of the GW distance is that the two latent dimensions need not be the same, as we only need intra-space pairwise distances to compute it. It gives more flexibility as the optimal latent dimension of each domain may not be the same, and using a higher dimension than necessary may result in the latent space living on a low-dimensional manifold.

3.3 Translation

To preliminarily test whether the cotraining of the two Auto-encoders benefits downstream tasks, we experiment a first attempt at translating between the two domain across their latent spaces. To achieve this, we use the coupling matrix T computed by the Gromov-Wasserstein distance (see Equation 2.5). Informally, the matrix value T_{ij} measures how strongly similar element x_i from domain A and element y_j from domain B are. We conjecture that a high value of T_{ij} may imply that x_i can be translated to y_j.

Following this conjecture, we propose a *Greedy translation* mechanism. To translate between two batches from each domain, we first encode them in their respective latent representation. We then compute the Gromov-Wasserstein distance between the two batches of latent codes and only keep the coupling matrix T. We then translate each element x_i in the following way:

$$Trans(x_i) = \arg\max_j T_{ij} \tag{3.2}$$

The translation process is similar if we want to translate the other way around. Other translation mechanism may be considered, but we initially only test this greedy approach, as it is simple and should already yield result that are significantly better than random translations. Note that with this greedy approach, a sample may only be translated to one and only one sample from the other domain, which may be a limitations if the two batches are unbalanced, i.e contains samples that are not similar.

4 Results

In this chapter, we present the experimental results of the representation learning and translation capabilities of our proposed model.

4.1 Gromov-Wasserstein Regularization

We begin with assessing the potential benefit of adding the GW loss and of cotraining the two Auto-Encoders on the representation learning process. Indeed, the GW loss can be seen as an additional regularizer on the latent space having an influence on the learned representations.

4.1.1 Visual assessment of latent spaces

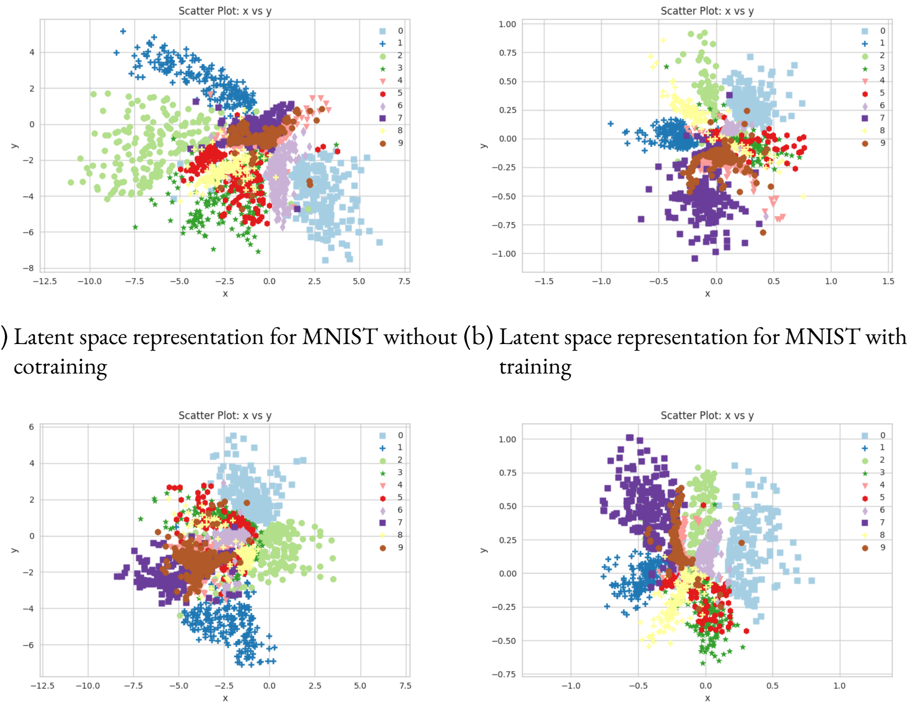
We train AEs, WAE-GANs and WAE-MMDs with and without cotraining on MNIST and MNIST_1 with a latent dimension of 2. The results can be visualized in Figure 4.1, Figure 4.3 and Figure 4.5. We also train the same models but with a latent dimension of 4 and then apply the t-SNE dimensionality reduction. The t-SNE results can be visualized in Figure 4.2, Figure 4.4 and Figure 4.6.

Generally, we observe that cotraining has a positive effect on the representation learning, by making the different point clusters more compact and reducing overlaps between different clusters. Unfortunately, we also observe multiple adverse effects. First, the effectiveness of adding the GW loss greatly depends on the λ hyperparameter, with small value changing leading to degenerate results and training. Furthermore, the GW loss forces the radius of the latent space to be small, as having a very compact latent space makes pairwise distances smaller and thus reduces the GW distance between the two latent spaces. Lastly, we sometimes observe some uncentering of the latent space, meaning that the latent points are not centered around the origin anymore. One way to combat the collapsing of the latent space could be to constrain the encoder to be orthogonal. As done in [Bun+19], a Procrustes-based regularization can effectively enforce orthogonality. We leave this avenue as potential future work.

4.1.2 Classification on learned latent representations

To try to quantify what we previously only visually observed, we implement a second experiment that assesses the representation learning quality of the trained models. To do so, after having trained the Auto-Encoders and fixing their weights, we train a classifier that labels the point with the 10 classes of the MNIST dataset. This classifier is only given the latent code of each sample as input, et return its label as output. We then compare the testset accuracy of these classifier (see Table 4.1). For WAE-GAN, the test accuracy is almost identical with and without cotraining. For WAE-MMD, the accuracy is *worse* with cotraining. On the other hand, for AE, the results are slightly better with cotraining, and the best overall performance is achieved by the 8-dimensional

AE with cotraining. These results indicates that adding the GW LOSS may only be relevant for an AE, and that the GW LOSS indeed allows both Auto-Encoders to improve their representation learning by sharing their learned latent structure through the GW LOSS gradient. These empirical observations could be further studied from a theoretical point of view, which is left as future work.



(a) Latent space representation for MNIST without cotraining

(b) Latent space representation for MNIST with cotraining

(c) Latent space representation for MNIST_1 without cotraining

(d) Latent space representation for MNIST_1 with cotraining

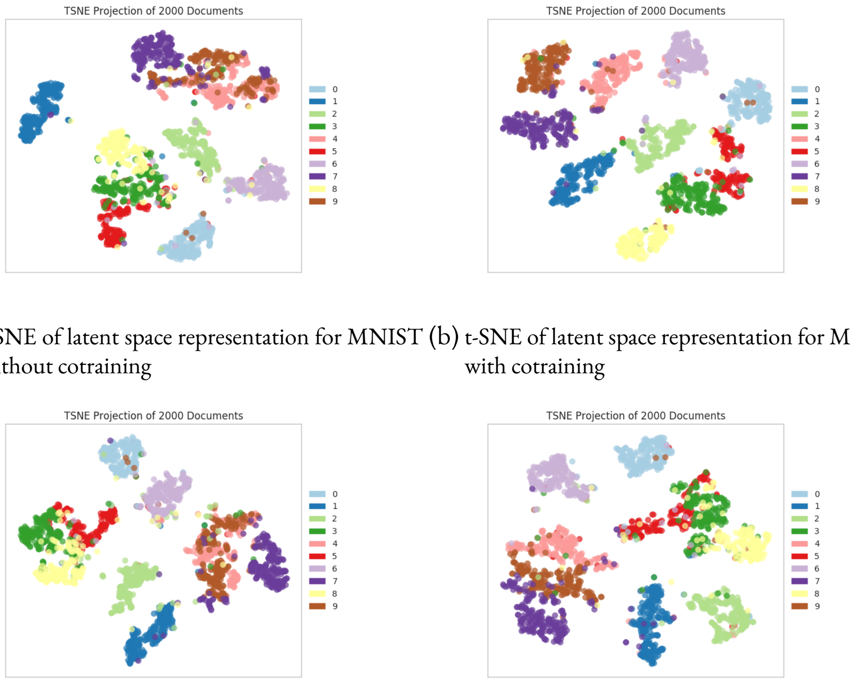
Figure 4.1: Two-dimensional latent representation learned by an AE with and without cotraining on MNIST and MNIST_1 with $\lambda = 1$. Points are colored with the class label they belong to. As can be observed, adding cotraining reduces the radius of the points (they are closer to the origin), and seems to degenerate clusters of point that are better separated.

4.2 Translation

To assess the benefit of the cotraining for downstream tasks, especially translation, we run an experiment using the Greedy translation presented in section 3.3. The results are summarized in Table 4.2. Unfortunately, the setup of this experiment gives inconsistent results that have high variance. Nevertheless, many models have a translation accuracy that is significantly above random, which indicates that the cotraining may make translation easier, and more importantly, that using the coupling matrix associated to the GW distance for translation may be a viable way of aligning the two latent spaces. Surprisingly, the best result (72% accuracy) is achieved by an 8-dimensional AE *without* cotraining. An interesting avenue of future work could thus be to train the AEs sepa-

Model	**Classification on MNIST**	**Classification on MNIST_1**
VAE:2d	82. ± 0.0	71.33 ± 11.74
VAE:4d	91.67 ± 1.43	90. ± 0.0
VAE:8d	94. ± 0.0	92.33 ± 2.87
WAE_GAN:2d	80.67 ± 1.43	80.67 ± 5.77
WAE_GAN:4d	90. ± 0.0	90. ± 2.48
WAE_GAN:8d	90.33 ± 2.87	91. ± 0.0
WAE_GAN_COTRAINING:2d	81.33 ± 3.79	79.67 ± 2.87
WAE_GAN_COTRAINING:4d	83 ± 1.96	84.5 ± 6.86
WAE_GAN_COTRAINING:8d	89.67± 1.43	89. ± 0.0
WAE_MMD:2d	79. ± 6.57	72.33 ± 1.43
WAE_MMD:4d	90.67 ± 3.79	89.33 ± 2.87
WAE_MMD:8d	93.67 ± 1.43	94. ± 0.0
WAE_MMD_COTRAINING:2d	67.33 ± 15.77	60.67 ± 3.79
WAE_MMD_COTRAINING:4d	87.33 ± 3.79	84. ± 2.48
WAE_MMD_COTRAINING:8d	92.67 ± 1.43	91.67 ± 1.43
AE:2d	77. ± 0.0	75.67 ± 3.79
AE:4d	90. ± 2.48	86.67 ± 1.43
AE:8d	94.67 ± 1.43	93. ± 2.48
AE_COTRAINING:2d	81.33 ± 5.17	75.33 ± 3.79
AE_COTRAINING:4d	92.33 ± 1.43	89.33 ± 1.43
AE_COTRAINING:8d	**95.67** ± 1.43	**94.33** ± 1.43

Table 4.1: Classification accuracy of a classifier trained on the respective learned representation. The results for VAE are given as a baseline. The best accuracy is achieved by the cotrained AE with an 8-dimensional latent space. The cotraining seems to be only beneficial for the AE. It has little effect for the WAE-GAN and an adverse effect for the WAE-MMD.



(a) t-SNE of latent space representation for MNIST without cotraining

(b) t-SNE of latent space representation for MNIST with cotraining

(c) t-SNE of latent space representation for MNIST_1 without cotraining

(d) t-SNE of latent space representation for MNIST_1 with cotraining

Figure 4.2: t-SNE dimensionality reduction of four-dimensional latent representation learned by an AE with and without cotraining on MNIST and MNIST_1 with $\lambda = 10$. Points are colored with the class label they belong to. We can see that the cotraining has a net positive effect on the representation learning process, as the clusters are better separated and there is less overlap between the different clusters.

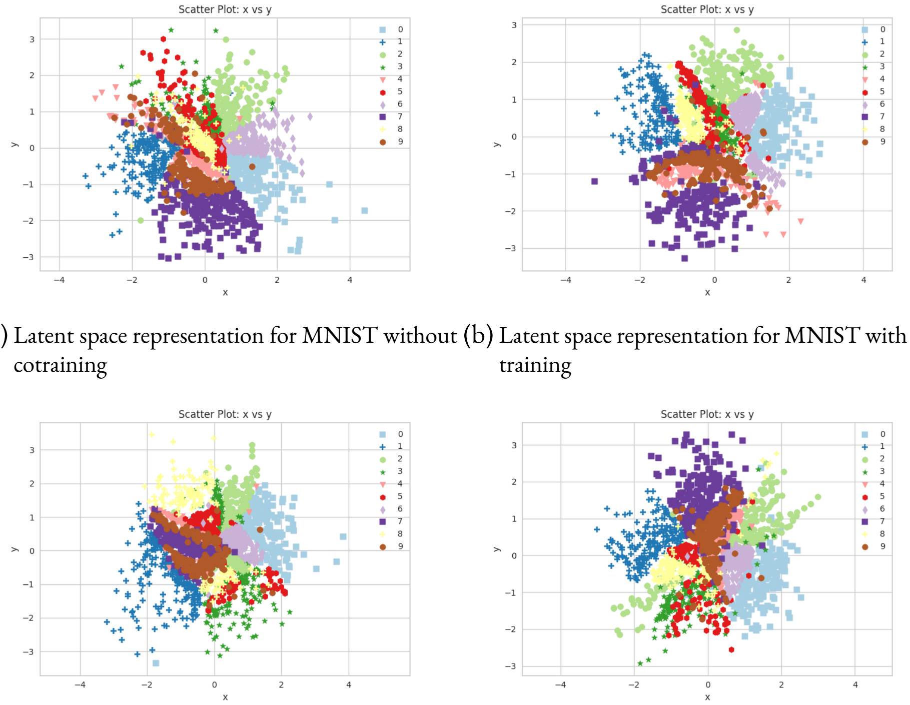
rately, and then training a translation network between the two latent spaces, using the coupling matrix to guide the translation.

One drawback of using the GW distance coupling matrix for translation is that it may only work if the two batches from each latent space have a *similar* distribution. Moreover, if the two batches are balanced, meaning that the proportion of samples from each class is uniform, any pairwise consistent translation may work, for example in MNIST, all the ones being translated to twos. Nevertheless, this translation mechanism could also be used for domain adaptation, using only a few paired samples to learn the correspondence between classes.

Lastly, we could also consider learning the translation while training the Auto-Encoders, similar to UNIT, which would allow us to introduce a cycle-consistency loss. This would require the translation to be differentiable, which could be implemented with a translation network between the two latent spaces.

Model	Translation MNIST->MNIST_1	Translation MNIST_1->MNIST
VAE:2d	35.08 ± 26.20	32.23 ± 14.64
VAE:4d	21.73 ± 14.43	22.33 ± 11.56
VAE:8d	31.55 ± 16.86	31.42 ± 15.99
WAE_GAN:2d	16.13 ± 21.98	15.13 ± 22.09
WAE_GAN:4d	31.13 ± 12.42	31.25 ± 16.06
WAE_GAN:8d	15.6 ± 7.32	16.52 ± 7.81
WAE_GAN_COTRAINING:2d	11.93 ± 8.11	12.43 ± 7.82
WAE_GAN_COTRAINING:4d	44.15 ± 4.31	46.83 ± 2.01
WAE_GAN_COTRAINING:8d	13.4 ± 7.04	14.07 ± 4.96
WAE_MMD:2d	4.5 ± 3.41	5.75 ± 7.92
WAE_MMD:4d	9.78 ± 6.41	9.95 ± 3.83
WAE_MMD:8d	10.12 ± 6.21	10.85 ± 4.07
WAE_MMD_COTRAINING:2d	15.9 ± 2.42	15.63 ± 2.56
WAE_MMD_COTRAINING:4d	23.23 ± 11.71	23.13 ± 13.88
WAE_MMD_COTRAINING:8d	36.17 ± 9.82	36.55 ± 13.12
AE:2d	12.67 ± 10.78	14.62 ± 11.58
AE:4d	24.58 ± 22.91	25.6 ± 23.96
AE:8d	72.77 ± 22.31	71.82 ± 25.78
AE_COTRAINING:2d	16.1 ± 7.4	17.03 ± 6.57
AE_COTRAINING:4d	28.25 ± 22.72	28.25 ± 21.
AE_COTRAINING:8d	56.25 ± 8.42	56.2 ± 6.32

Table 4.2: Accuracy of Greedy translation section 3.3 using the coupling matrix computed by the GW distance. The translation was performed on batches of size 100 (one for each domain) containing *exactly* the same samples, i.e each sample of the MNIST_1 batch is the inverse of a sample of the MNIST batch. This makes translation easier as the distribution of both batches is identical. Unfortunately, the results are very inconsistent and have a large variance. Nevertheless, some models perform far better than random, which would be 10% accuracy. The best results are achieved by AE:8d and AE_COTRAINING:8d, and surprisingly, the best of the two is the one without cotraining.



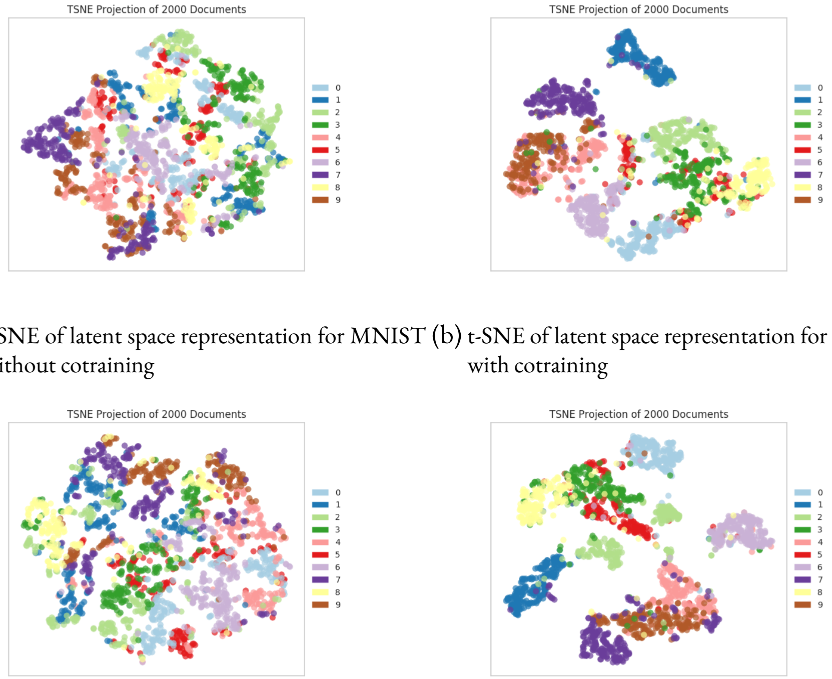
(a) Latent space representation for MNIST without cotraining

(b) Latent space representation for MNIST with cotraining

(c) Latent space representation for MNIST_1 without cotraining

(d) Latent space representation for MNIST_1 with cotraining

Figure 4.3: Two-dimensional latent representation learned by a WAE-GAN with and without cotraining on MNIST and MNIST_1 with $\lambda = 1$. Points are colored with the class label they belong to. Contrary to the results for AE, here, the radius remains the same with and without cotraining, which may mean that the Adverserial loss from the Discriminator *dominates* the GW Loss. This also reduces the effect of cotraining, which can be seen here as the difference between with and without cotraining is not significant. We could obviously choose an higher λ to give more weight to the GW Loss, but in our experiments, an higher λ lead to degenerate results for WAE-GAN.



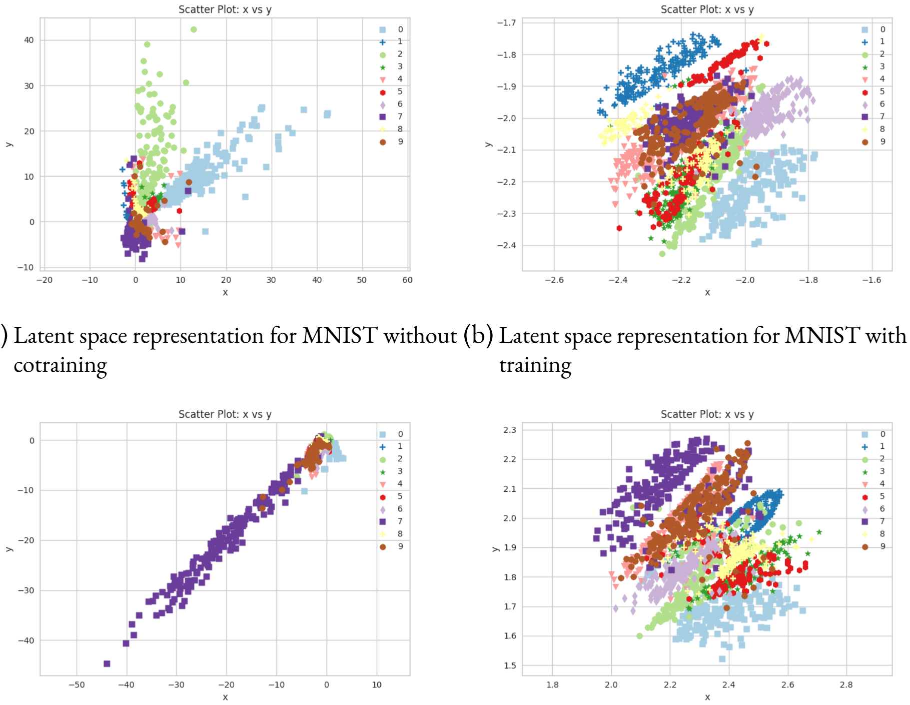
(a) t-SNE of latent space representation for MNIST without cotraining

(b) t-SNE of latent space representation for MNIST with cotraining

(c) t-SNE of latent space representation for MNIST_1 without cotraining

(d) t-SNE of latent space representation for MNIST_1 with cotraining

Figure 4.4: t-SNE dimensionality reduction of four-dimensional latent representation learned by a WAE-GAN with and without cotraining on MNIST and MNIST_1 with $\lambda = 100$. Points are colored with the class label they belong to. Here, the cotraining has a net positive effect on the latent representation learning, as clusters of point that are similar only appear when we impose the GW Loss.



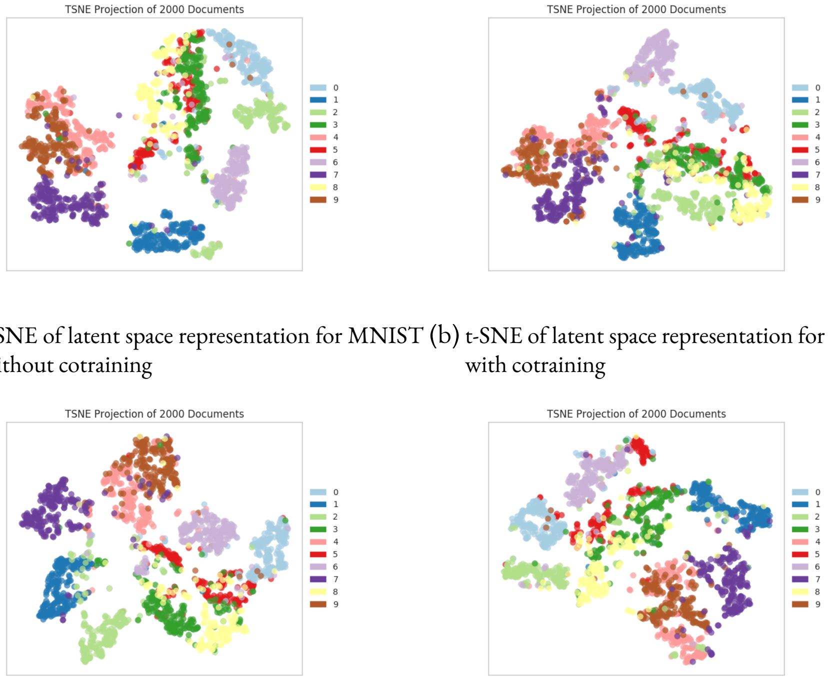
(a) Latent space representation for MNIST without cotraining

(b) Latent space representation for MNIST with cotraining

(c) Latent space representation for MNIST_1 without cotraining

(d) Latent space representation for MNIST_1 with cotraining

Figure 4.5: Two-dimensional latent representation learned by a WAE-MMD with and without cotraining on MNIST and MNIST_1 with $\lambda = 100$. Points are colored with the class label they belong to. Here we can clearly see the effect of the GW LOSS, where the points are far closer and more compact with cotraining. We can also observe that the points are not centered around the origin with cotraining, which is a way for the network to minimize the MMD loss, as the MMD gradient vanishes for outlier points. This uncentering of the latent representation was also observed in WAE-GAN, especially when λ is high, as then the network focuses on minimizing the GW LOSS as it has more weight than the regularization loss.



(a) t-SNE of latent space representation for MNIST without cotraining

(b) t-SNE of latent space representation for MNIST with cotraining

(c) t-SNE of latent space representation for MNIST_1 without cotraining

(d) t-SNE of latent space representation for MNIST_1 with cotraining

Figure 4.6: t-SNE dimensionality reduction of four-dimensional latent representation learned by a WAE-MMD with and without cotraining on MNIST and MNIST_1 with $\lambda = 1$. Points are colored with the class label they belong to. Here, the results are very similar, whether we use cotraining or not.

5 Conclusion

In this report, we presented a novel framework with a potential application to Unsupervised Domain Translation using the Gromov-Wasserstein distance from Optimal Transport theory. We introduced the GW loss between the latent spaces of two Auto-Encoders of different types. We studied the effect of this regularizer on the representation learning process and observe that for some model, the learned representation of the two Auto-Encoders are better than if they were learned separately. We also assessed how the GW loss may benefit downstream tasks such as translation and implemented a first attempt at using the coupling matrix from the GW distance to align both latent spaces. We found that the alignments from the coupling are meaningful and may be used for translation, but this mechanism is still too inconsistent to be reliably used for translation or other similar tasks. Unfortunately, we also observed multiple adverse effects of adding the GW loss, the main one being a collapse of the latent space, meaning that the pairwise distance between the latent samples vanishes.

We also mentioned multiple potential avenues of future work. To combat the collapsing of the latent space, we could introduce a loss forcing the encoder to be orthogonal. To embed translation in the learning process, we would need to make the translation differentiable, for examples through a translation network between the two latent spaces, potentially using the coupling matrix as a guide for translation. Learning the translation would allow us to introduce different inductive biases such as cycle-consistency, similar to the UNIT framework. Lastly, a more theoretical formalization similar to [BAG19] could provide precious insights and guide the modeling of our framework.

Acronyms

AEs	Auto-Encoders
ELBO	Evidence Lower Bound
GAN	Generative Adversarial Network
GANs	Generative Adversarial Networks
GW	Gromov-Wasserstein
JSD	Jensen–Shannon divergence
KL	Kullback-Leibler
MMD	Maximum Mean Discrepancy
MSE	Mean Square Error
NN	Neural Networks
OT	Optimal Transport
UDT	Unsupervised Domain Translation
UNIT	Unsupervised Image-to-image Translation
VAE	Variational Auto-Encoder
VAEs	Variational Auto-Encoders
WAE	Wasserstein Auto-Encoder
WGAN	Wasserstein GAN

Bibliography

[AB17] M. Arjovsky and L. Bottou. "Towards Principled Methods for Training Generative Adversarial Networks". *ArXiv* abs/1701.04862, 2017.

[ACB17] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein gan". *arXiv preprint arXiv:1701.07875*, 2017.

[BAG19] E. de Bézenac, I. Ayed, and P. Gallinari. "Optimal unsupervised domain translation". *arXiv preprint arXiv:1906.01292*, 2019.

[Bun+19] C. Bunne, D. Alvarez-Melis, A. Krause, and S. Jegelka. "Learning Generative Models across Incomparable Spaces". In: *International Conference on Machine Learning*. 2019, pp. 851–861.

[Goo+14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[GPC18] A. Genevay, G. Peyre, and M. Cuturi. "Learning Generative Models with Sinkhorn Divergences". In: ed. by A. Storkey and F. Perez-Cruz. Vol. 84. Proceedings of Machine Learning Research. PMLR, Playa Blanca, Lanzarote, Canary Islands, 2018, pp. 1608–1617. URL: http://proceedings.mlr.press/v84/genevay18a.html.

[Gro+19] A. Grover, C. Chute, R. Shu, Z. Cao, and S. Ermon. "AlignFlow: Cycle Consistent Learning from Multiple Domains via Normalizing Flows". *arXiv preprint arXiv:1905.12892*, 2019.

[KB14] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*, 2014.

[KW13] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". *arXiv preprint arXiv:1312.6114*, 2013.

[LBK17] M.-Y. Liu, T. Breuel, and J. Kautz. "Unsupervised image-to-image translation networks". In: *Advances in neural information processing systems*. 2017, pp. 700–708.

[Mém11] F. Mémoli. "Gromov–Wasserstein distances and the metric approach to object matching". *Foundations of computational mathematics* 11:4, 2011, pp. 417–487.

[PC19] G. Peyré and M. Cuturi. "Computational Optimal Transport". *Foundations and Trends in Machine Learning* 11, 2019, pp. 355–607.

[PCS16] G. Peyré, M. Cuturi, and J. Solomon. "Gromov-Wasserstein Averaging of Kernel and Distance Matrices". In: *ICML*. 2016.

[Sal+18] T. Salimans, H. Zhang, A. Radford, and D. Metaxas. “Improving GANs using optimal transport”. *arXiv preprint arXiv:1803.05573*, 2018.

[Tol+18] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. “Wasserstein Auto-Encoders”. In: *International Conference on Learning Representations*. 2018. URL: `https://openreview.net/forum?id=HkL7n1-0b`.

[Zhu+17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. DOI: `10.1109/iccv.2017.244`. URL: `http://dx.doi.org/10.1109/ICCV.2017.244`.