Generative modeling project: Neural Optimal Transport

Paul Barbier[1] **and Bastien Le Chenadec**[1]

[1]École des Ponts ParisTech, Master MVA

1 Introduction

Optimal Transport (OT) is a mathematical framework that aims to find the most efficient way to transport a distribution of mass to another. This framework has been used extensively in the context of generative models, for instance as a loss function in the training of Generative Adversarial Networks (GANs) or by learning a mapping between two distributions. In this project, we aim to study the paper "Neural Optimal Transport" (Korotin, 2023) [1] which introduces an algorithm to train a neural network to learn the optimal transport between two distributions, with applications in image generation. We will introduce some optimal transport problems in various formulations, but we will focus on giving a general overview and some details and hypotheses will be omitted for the sake of clarity. We will then present the results of our experiments with this algorithm on synthetic data and a real dataset.

2 Background on Optimal Transport

2.1 Optimal Transport Problem

Let μ and ν be two probability distributions on $\mathcal{X}$ and $\mathcal{Y}$ respectively (typically $\mathcal{X}, \mathcal{Y} = \mathbb{R}^n, \mathbb{R}^m$). To give a meaning to "efficiently" transporting mass, we define a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that quantifies the cost of transporting a unit of mass in $\mathcal{X}$ to one in $\mathcal{Y}$. The (Monge) optimal transport problem consists in finding a **transport map** $T^* : \mathcal{X} \to \mathcal{Y}$ such that:

$$T^* \in \underset{T\#\mu=\nu}{\text{Argmin}} \int_{\mathcal{X}} c(x, T(x)) d\mu(x) \qquad \text{Cost}(\mu, \nu) = \int_{\mathcal{X}} c(x, T^*(x)) d\mu(x) \tag{1}$$

where $T\#\mu$ is the pushforward distribution of μ by T, defined by $(T\#\mu)(A) = \mu(T^{-1}(A))$ for any measurable set $A \subset \mathcal{Y}$. This formulation calls for a deterministic mapping from $\mathcal{X}$ to $\mathcal{Y}$, which is not always desirable or feasible under general assumptions. Kantorovich introduced a relaxed OT problem that aims at finding a **transport plan** $\pi^* \in \Pi(\mu, \nu)$ in the set of joint distributions on $\mathcal{X} \times \mathcal{Y}$ with marginals μ and ν such that:

$$\pi^* \in \underset{\pi\in\Pi(\mu,\nu)}{\text{Argmin}} \int_{\mathcal{X}\times\mathcal{Y}} c(x, y) d\pi(x, y) \qquad \text{Cost}(\mu, \nu) = \int_{\mathcal{X}\times\mathcal{Y}} c(x, y) d\pi^*(x, y) \tag{2}$$

In general the solution to the Kantorovich problem is stochastic, but it may be deterministic in which case it is also a solution to the Monge problem. Building on this idea of stochasticity in the solution, weak OT was introduced as a generalization of the Kantorovich problem, where the cost function is of the form $C : \mathcal{X} \times \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$. In this case the weak OT problem writes:

$$\pi^* \in \underset{\pi\in\Pi(\mu,\nu)}{\text{Argmin}} \int_{\mathcal{X}} C(x, \pi(\cdot|x)) d\pi(x) \qquad \text{Cost}(\mu, \nu) = \int_{\mathcal{X}} C(x, \pi^*(\cdot|x)) d\pi^*(x) \tag{3}$$

where $\pi(\cdot|x)$ is the conditional distribution of π given x and $\pi(x)$ is the marginal distribution of π on $\mathcal{X}$ (which is actually μ).

2.2 Weak OT duality

There are strong duality results for the Kantorovich problem, which we have seen extensively in class. Here we focus on duality results for the weak OT problem. For weak OT cost C, and f defined on $\mathcal{Y}$ sufficiently regular, the weak C-transform of f is defined on $\mathcal{X}$ as:

$$f^C(x) = \inf_{\rho\in\mathcal{P}(\mathcal{Y})} \left\{ C(x, \rho) - \int_{\mathcal{Y}} f(y) d\rho(y) \right\} \tag{4}$$

where $\mathcal{P}(\mathcal{Y})$ denotes the set of probability distributions on $\mathcal{Y}$. The dual form of the weak OT problem is then:

$$f^* \in \underset{f}{\text{Argmax}} \int_{\mathcal{X}} f^C(x) d\mu(x) + \int_{\mathcal{Y}} f(y) d\nu(y) \qquad \text{Cost}(\mu, \nu) = \int_{\mathcal{X}} f^{*C}(x) d\mu(x) + \int_{\mathcal{Y}} f^*(y) d\nu(y) \tag{5}$$

The transport cost $\text{Cost}(\mu, \nu)$ in the dual form is equal to the cost of the primal form, which is a generalization of the Kantorovich duality to the weak OT problem.

3 Neural Optimal Transport

In (Korotin, 2023) [1], the authors aim to solve the weak OT problem with a neural network. First they reformulate the weak dual problem with noise outsourcing, then they introduce an algorithm to solve this problem.

3.1 Weak dual OT reformulation

From now on we consider $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^m$. We introduce $\mathcal{Z} \subset \mathbb{R}^d$ a latent space with a distribution $\rho \in \mathcal{P}(\mathcal{Z})$. The following result holds true under some basic assumptions on ρ:

$$f^C(x) = \inf_t \left\{ C(x, T\#\rho) - \int_{\mathcal{Z}} f(t(z)) d\rho(z) \right\} \tag{6}$$

This result can be integrated against μ to replace the dual weak OT problem by a maximin problem:

$$\begin{aligned} \text{Cost}(\mu, \nu) &= \sup_f \inf_T \int_{\mathcal{Y}} f(y) d\nu(y) + \int_{\mathcal{X}} \left(C(x, T(x, \cdot)\#\rho) - \int_{\mathcal{Z}} f(T(x, z)) d\rho(z) \right) d\mu(x) \\ &= \sup_f \inf_T \mathcal{L}(f, T) \end{aligned} \tag{7}$$

where $T : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$. The proofs of these results are in the paper [1]. This reformulation can be interpreted in the following way: the solution to the weak OT problem is a **stochastic map** $T : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ that is split into two parts, a deterministic part $T(x, \cdot)$ and a stochastic part $z \mapsto T(x, z)$. $\mathcal{Z}$ is a latent space that models the randomness in the transport.

This reformulation is known as **noise outsourcing** [2], a trick which is justified by a strong theoretical result that we recall here:

Theorem 1. *If X and Y are random variables in suitable spaces $\mathcal{X}$ and $\mathcal{Y}$, then there exists $\eta \sim \mathcal{U}([0, 1])$ with $\eta \perp\!\!\!\perp X$ and a function $h : [0, 1] \times \mathcal{X} \longrightarrow \mathcal{Y}$ such that $(X, Y) = (X, h(\eta, X))$ almost surely.*

Solving (7) does not necessarily yield a solution T^* to the weak OT problem. However under sufficient convexity conditions on C the solution to the maximin problem is a solution to the weak OT problem. With that in mind the authors introduce an algorithm to solve this problem and restrict their attention to such cost functions.

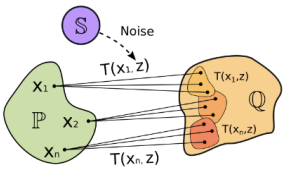


Figure 1: Transportation map with outsourced noise z [1]

3.2 SGAD

The authors prove that the space of neural networks is rich enough to approximate stochastic maps T which motivates the use of neural networks to solve (7). f and T in (7) are parametrized by neural networks f_ω and T_θ respectively. The authors use a stochastic gradient ascent descent (SGAD) algorithm to solve the problem. The algorithm is presented in Figure 2.

Algorithm 1 Stochastic Gradient Ascent Descent (SGAD) algorithm for Neural Optimal Transport

Input: distributions μ, ν, ρ accessible by samples, mapping network $T_\theta : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^d$, potential network $f_\omega : \mathbb{R}^n \to \mathbb{R}$, number of inner iterations K_T, (weak) cost $C : \mathcal{X} \times \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$, empirical estimator $\widehat{C}(x, T(x, Z))$ for the cost
Output: learned stochastic OT map T_θ representing an OT plan between distributions μ, ν
repeat
  Sample batches $Y \sim \nu$, $X \sim \mu$, for each $x \in X$ sample batch $Z_x \sim \rho$
  $\mathcal{L}_f \leftarrow \frac{1}{|X|} \sum_{x \in X} \frac{1}{|Z_x|} \sum_{z \in Z_x} f_\omega(T_\theta(x, z)) - \frac{1}{|Y|} \sum_{y \in Y} f_\omega(y)$
  Update ω by using $\frac{\partial \mathcal{L}_f}{\partial \theta}$
  for $k_T = 1, 2, \ldots, K_T$ **do**
    Sample batch $X \sim \mu$, for each $x \in X$ sample batch $Z_x \sim \rho$
    $\mathcal{L}_T \leftarrow \frac{1}{|X|} \sum_{x \in X} [\widehat{C}(x, T_\theta(x, Z_x)) - \frac{1}{|Z_x|} \sum_{z \in Z_x} f_\omega(T_\theta(x, z))]$
    Update θ by using $\frac{\partial \mathcal{L}_\theta}{\partial \theta}$
until converged

Figure 2: Neural Optimal Transport algorithm [1]

This approach is similar to the one used in GANs, with a generator network T_θ and a discriminator network f_ω. The algorithm alternates between updating the generator and the discriminator, with the generator trying to minimize the cost function and the discriminator trying to maximize it. Yet this approach is different from GANs in that the learned T_θ is a solution to the weak OT problem.

3.3 Application to generative modeling

Weak optimal transport can be used for generative modeling by learning a stochastic map between two (data) distributions. This stochastic map is learned by a neural network, which is trained to solve the weak optimal transport problem. One

appealing aspect of weak optimal transport is that it allows for stochasticity in the solution, which is desirable in many generative modeling tasks such as image generation (for instance generating multiple plausible images from a single input image).

However, learning with weak optimal transport also has its own set of challenges. For instance, the choice of the cost function can be difficult, for instance when the distributions are high-dimensional. Furthermore, the cost function needs to satisfy strict convexity conditions in the context of this paper to ensure that the solution to the maximin problem is a solution to the weak optimal transport problem. The training of the neural network can also be challenging, requiring careful tuning of hyperparameters. It is also subject to the usual challenges of generative modeling, such as mode collapse.

From a theoretical standpoint, weak OT is also challenging because of the non-uniqueness of the solution, and the difficulty to derive theoretical solutions even in simple cases. The authors of the paper consider the weak quadratic cost with $\gamma = 1$ for which a unique *class* of solutions exists (i.e. all solutions verify some gradient property), but this is not the case in general.

4 EXPERIMENTS

For this project, we conducted experiments using the code provided by the authors. You can find our scripts at https://github.com/bastienlc/NOT. We did a first experiment with generated synthetic data and a second one with a more realistic use-case using a large dataset.

4.1 Synthetic data

First, we conducted experiments using synthetic data on a simple 2D grid. Our spaces $\mathcal{X}$ and $\mathcal{Y}$ are $\mathbb{R}^2$ and we adopt the same γ-weak quadratic cost function as in the paper (8) (with $\gamma = 1$), thus solving the weak OT problem.

$$C(x,\mu) = \int_{\mathcal{Y}} \frac{1}{2}\|x-y\|^2 d\mu(y) - \frac{1}{2}\mathrm{Var}(\mu) = \frac{1}{2}\|x - \int_{\mathcal{Y}} y\, d\mu(y)\|^2 \tag{8}$$

The input distribution is a $2D$ standard normal distribution and the target distribution is a mixture of two semicircles ("moon distribution"). This is not too different from the toy examples in the paper which were either a spiral or a mixture of gaussians.

The networks are simple feedforward neural networks with 2 hidden layers of 100 hidden units each and ReLU activations. We used similar hyperparameters to the ones in the paper (batch size of 64, 10 iterations for the optimization of T_θ, Adam optimizer, 10000 steps...). The training is straightforward with these parameters, but while experimenting with different ones we found that the model is quite sensitive to the batch size and the number of iterations for the optimization of T_θ.

The main results are shown in Figure 3. The results are similar to the ones in the paper with a good mapping of the input distribution to the target distribution. The resulting distribution is a bit more noisy than the target distribution but the general shape is well captured.

In figure 4, we show the inner workings of the model. The left plot shows the average of the learned transport for different points in the input distribution. On average, points in the center will not move (they will be mapped equally in different directions), while points on the border will be moved towards the center because the input distribution has a bigger support than the target distribution. The middle plot shows the transport for a batch of points; we can see that there is a lot of movement to obtain the target distribution, and also that the same point can sometimes be moved in very different directions. The right plot shows the optimal transport plan obtained with the POT[1] library. The learned transport is very close to the optimal transport plan.

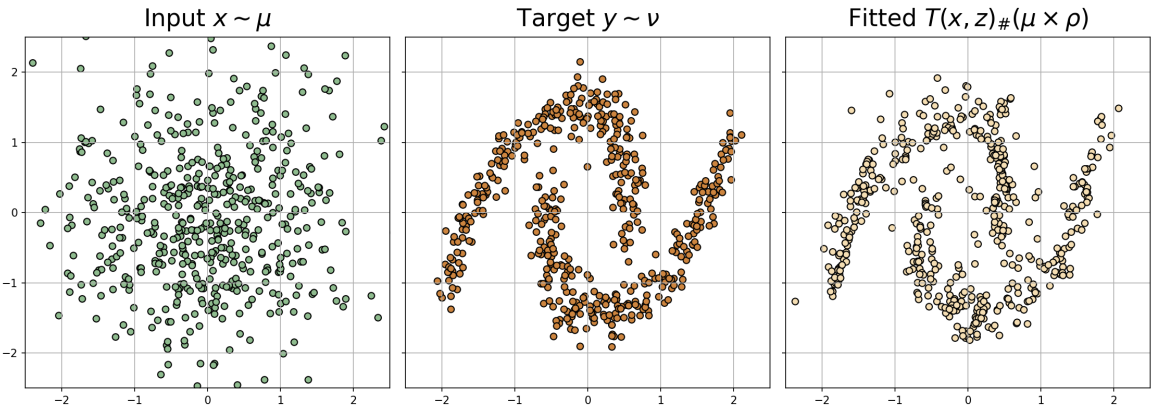


Figure 3: Synthetic data experiment. Left: input gaussian distribution. Middle: target distribution. Right: learned transport of the input distribution to the target distribution.

[1] https://pythonot.github.io/

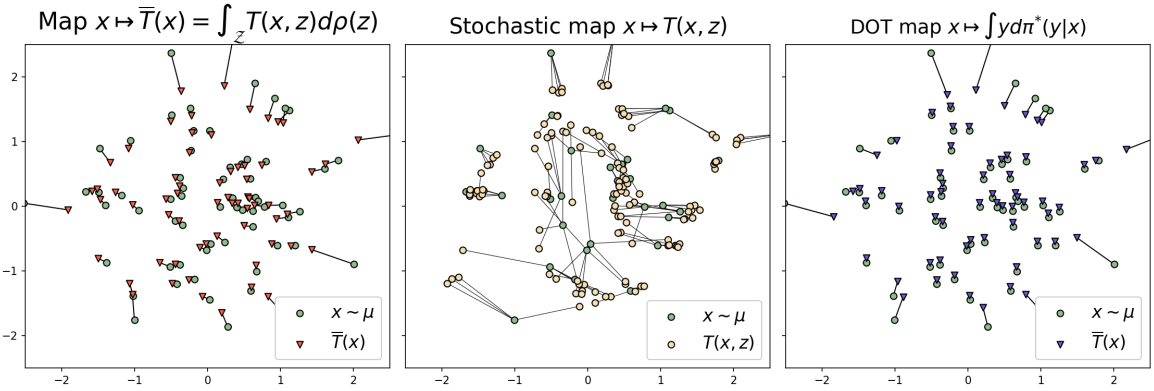


Figure 4: Synthetic data experiment. Left: average of the learned transport for different points. Middle: learned transport for a batch of points. Right: optimal transport plan obtained with the POT library.

4.2 Real dataset

In their paper the authors made experiments using faces from animes and real persons which we found amusing, so we decided to make a similar experiment with a different dataset. More precisely, we used the CelebA dataset [3] which comprises 200k images of celebrity faces with attribute annotations. For the second dataset, we chose CartoonSet100K, a dataset introduced in [4]. It's a large-scale dataset containing 100k images of 2D generated cartoon avatar images.

Figure 5: Images samples from CartoonSet100K

For this large-scale experiment we used a cloud instance with a Nvidia A100 GPU with 40GB of memory. We increased the batch size per dataset from 64 to 128 and we used 128×128 images in place of 64×64 to get generated images with a higher resolution. Increasing the batch size led to better memory utilisation and quicker convergence as we'll see in the next paragraphs.

On the figure below, you can see how the loss function evolves throughout the training. Unusually, the loss function here is fluctuating around the constant value 0 which is expected: the algorithm outlined above optimises the difference of two antagonist loss terms so as one is diminishing the other is getting larger.

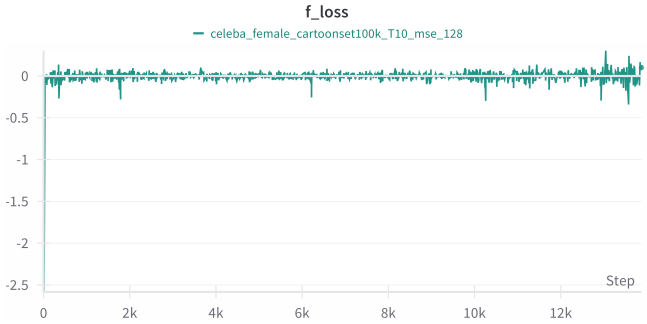


Figure 6: $\mathcal{L}_f$ during training

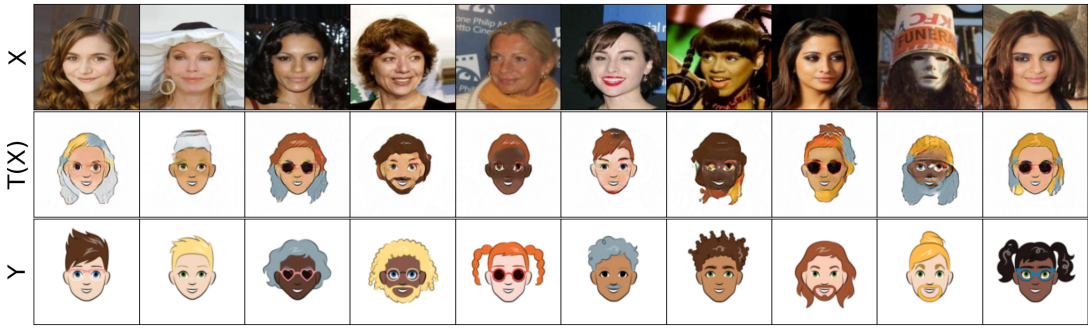
Those experiments are quite long to run (about a day for ~15k iterations with the parameters given above) so we chose to focus on the strong OT problem in this case, while we used the weak OT problem in the synthetic data experiment. The results for a fixed sample of the test set are given in 7.

4.2.1 Observations

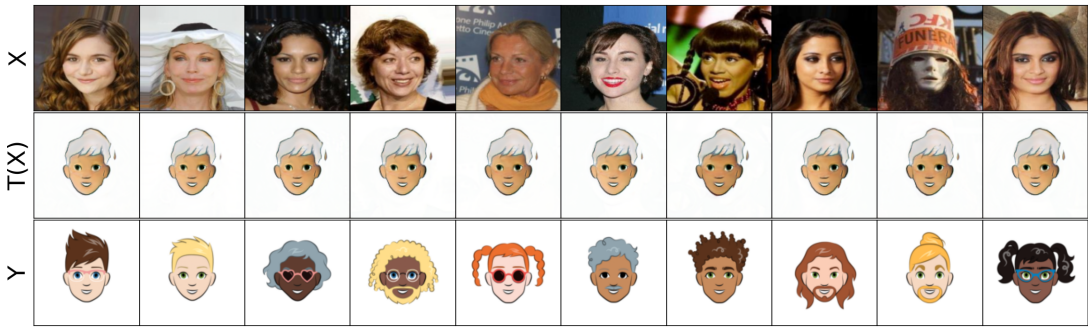
Figure 7a depicts the initial state of the model.

Around 5000 iterations, we witness realistic mappings of the real faces onto the cartoon avatar distribution (figure 7b). This is far less than what the authors observed in their experiments, this number was closer to 40k iterations for convergence.

(a) Iteration 1



(b) Iteration 5000



(c) Iteration 13800

Figure 7: Test images

However, our images have an empty background so it might be easier in comparison with anime face images which contain a complex surrounding as they are extracted from animes. One can see that the color of the skin and the hair map well. However, the model struggles when there are glasses on the target image. It translates glasses into dark black holes. Still the model is able to generate new avatars that are not in the training set but that are plausible, which is an encouraging result.

At this point, we thought we were far from the end and we decided to continue the training for a night. Unfortunately, after a few thousands more iterations, the model collapsed on a single mode as shown in figure 7c. Surprisingly, the constant predicted face is changing with at each training step but the model remains stuck in this mode. This is a common issue in generative modeling and it is not specific to the weak OT problem.

5 CONCLUSION

In this project, we studied the paper "Neural Optimal Transport" (Korotin, 2023) [1] which introduces an algorithm to train a neural network to learn the weak optimal transport plan between two distributions. We conducted experiments using synthetic data and a real dataset, and obtained results similar to the ones in the paper. We also observed that the model is quite sensitive to hyperparameters such as the batch size and the number of iterations for the optimization of T_θ. In the end we encountered mode collapse, which was disappointing but not unexpected. Overall, we found that the proposed algorithm is effective, and that weak optimal transport is a promising framework for generative modeling.

REFERENCES

[1] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. *arXiv (Cornell University)*, 1 2022. doi: 10.48550/arxiv.2201.12220. URL https://arxiv.org/abs/2201.12220.

[2] Olav Kallenberg. *Foundations of modern probability*. Probability and its Applications (New York). Springer-Verlag, New York, second edition, 2002. ISBN 0-387-95313-2. doi: 10.1007/978-1-4757-4015-8. URL http://dx.doi.org/10.1007/978-1-4757-4015-8.

[3] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[4] Amelie Royer, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole, and Kevin Murphy. XGAN: unsupervised image-to-image translation for many-to-many mappings. *CoRR*, abs/1711.05139, 2017. URL http://arxiv.org/abs/1711.05139.