HAL open science

Learning optimal transport between two empirical distributions with normalizing flows

Florentin Coeurdoux, Nicolas Dobigeon, Pierre Chainais

To cite this version:

Florentin Coeurdoux, Nicolas Dobigeon, Pierre Chainais. Learning optimal transport between two empirical distributions with normalizing flows. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2022), Sep 2022, Grenoble, France. 10.48550/arXiv.2207.01246 . hal-03713840

HAL Id: hal-03713840

https://hal.science/hal-03713840

Submitted on 5 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning Optimal Transport Between two Empirical Distributions with Normalizing Flows*

Florentin Coeurdoux[1[0000−0002−2100−1438]]✉, Nicolas Dobigeon[1[0000−0001−8127−350X]], and Pierre Chainais[2[0000−0003−4377−7584]]

[1] University of Toulouse, IRIT/INP-ENSEEIHT, F-31071 Toulouse, France
{Florentin.Coeurdoux, Nicolas.Dobigeon}@irit.fr
[2] Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
pierre.chainais@centralelille.fr

Abstract. Optimal transport (OT) provides effective tools for comparing and mapping probability measures. We propose to leverage the flexibility of neural networks to learn an approximate optimal transport map. More precisely, we present a new and original method to address the problem of transporting a finite set of samples associated with a first underlying unknown distribution towards another finite set of samples drawn from another unknown distribution. We show that a particular instance of invertible neural networks, namely the normalizing flows, can be used to approximate the solution of this OT problem between a pair of empirical distributions. To this aim, we propose to relax the Monge formulation of OT by replacing the equality constraint on the push-forward measure by the minimization of the corresponding Wasserstein distance. The push-forward operator to be retrieved is then restricted to be a normalizing flow which is trained by optimizing the resulting cost function. This approach allows the transport map to be discretized as a composition of functions. Each of these functions is associated to one sub-flow of the network, whose output provides intermediate steps of the transport between the original and target measures. This discretization yields also a set of intermediate barycenters between the two measures of interest. Experiments conducted on toy examples as well as a challenging task of unsupervised translation demonstrate the interest of the proposed method. Finally, some experiments show that the proposed approach leads to a good approximation of the true OT.

Keywords: Normalizing flows · Optimal transport · Generative Model.

1 Introduction

The optimal transport (OT) problem was initially formulated by the French mathematician Gaspard Monge. In his seminal paper published in 1781 [18], he

* This work was supported by the Artificial Natural Intelligence Toulouse Institute (ANITI, ANR-19-PI3A-0004), the AI Sherlock Chair (ANR-20-CHIA-0031-01), the ULNE national future investment programme (ANR-16-IDEX-0004) and the Hauts-de-France Region.

raised the following question: how to move a pile of sand to a target location with the least possible effort or cost? The objective was to find the best way to minimize this cost by a transport plan, without having to list all the possible matches between the starting and ending points. More recently, thanks to recent advances related to computational issues [22], OT has founded notable successes with respect to applications ranging from image processing and computer vision [21] to machine learning [2] and domain adaptation [7].

Normalizing flows (NFs) have also attracted a lot of interest in the machine learning community, motivated in particular by their ability to model high dimensional data [19,15]. These deep generative models are characterized by an invertible operator that associates any input data distribution with a target distribution that is usually chosen to be Gaussian. They have the great advantage of leading to tractable distributions, which eases direct sampling and density estimation. Applications of these generative models include image generation with real-valued non-volume preserving transformations (RealNVP) [10] or generative flows using an invertible 1x1 convolution (GLOW) [14].

Motivated by the similarities between the problem of OT and the training of NF, this paper proposes a neural architecture and a corresponding training strategy that permits to learn an approximate Monge map between any two empirical distributions. The proposed framework is based on a relaxation of the Monge formulation of OT. To adapt the training loss to the flow-based structure of the network, this loss function is supplemented with a Sobolev regularisation to promote minimal efforts achieved by each flow. Numerical simulations show that this regularisation results in a smoother and more efficient trajectory. Interestingly, the discretization inherent to the flow-based structure of the network implicitly provides intermediate transports and, at the same time, Wasserstein barycenters [1]. To the best of our knowledge, this is the first time that NFs are considered to address OT and Wasserstein barycenter computation, up to interesting dimensions.

Contributions. Our contributions are twofold: i) Section 2 recalls the Monge formulation of OT and proposes a relaxation in the case of a transport between two empirical distributions. ii) Section 3 presents the generic framework based on NFs and describes a particular instance to solve the OT problem. Section 4 presents some experimental results illustrating the performance of the proposed method. Section 6 concludes this paper.

2 Relaxation of the optimal transport problem

Let μ and ν be two probability measures with finite second order moments. More general measures, for example on $\mathcal{X} = \mathbb{R}^d$ (where $d \in \mathbb{N}^*$ is the dimension), can have a density $\mathrm{d}\mu(x) = p_X(x)dx$ with respect to the Lebesgue measure, often noted $p_X = \frac{\mathrm{d}\mu}{\mathrm{d}x}$, which means that

$$\forall h \in \mathcal{C}\left(\mathbb{R}^d\right), \quad \int_{\mathbb{R}^d} h(x)\mathrm{d}\mu(x) = \int_{\mathbb{R}^d} h(x)p_X(x)\mathrm{d}x \tag{1}$$

where $\mathcal{C}(\cdot)$ is the class of continuous functions. In the remainder of this paper, $\mathrm{d}\mu(x)$ and $p_X(x)\mathrm{d}x$ will be used interchangeably.

2.1 Background on optimal transport

Let consider $\mathcal{X}$ and $\mathcal{Y}$ two separable metric spaces. Any measurable application $T : \mathcal{X} \to \mathcal{Y}$ can be extended to the so-called push-forward operator $T_\sharp$ which moves a probability measure on $\mathcal{X}$ to a new probability measure on $\mathcal{Y}$. For any measure μ on $\mathcal{X}$, one defines the image measure $\nu = T_\sharp \mu$ on $\mathcal{Y}$ such that

$$\forall h \in \mathcal{C}(\mathcal{Y}), \quad \int_{\mathcal{Y}} h(y)\mathrm{d}\nu(y) = \int_{\mathcal{X}} h(T(x))\mathrm{d}\mu(x). \tag{2}$$

Intuitively, the application $T : \mathcal{X} \to \mathcal{Y}$ can be interpreted as a function moving a single point from one measurable space to another [22]. The operator $T_\sharp$ pushes each elementary mass of a measure μ on $\mathcal{X}$ by applying the function T to obtain an elementary mass in $\mathcal{Y}$. The problem of OT as formulated by Monge is now stated in a general framework. For a given cost function $c : \mathcal{X} \times \mathcal{Y} \to [0, +\infty]$, the measurable application $T : \mathcal{X} \to \mathcal{Y}$ is called the OT map from a measure μ to the image measure $\nu = T_\# \mu$ if it reaches the infimum

$$\inf_T \left\{ \int_{\mathcal{X}} c(x, T(x))\mathrm{d}\mu(x) : T_\sharp \mu = \nu \right\}. \tag{3}$$

Alternatively the Kantorovitch formulation of OT results from a convex relaxation of the Monge problem (3). By defining Π as the set of all probabilistic couplings with marginals μ and ν, it yields the optimal π that reaches

$$\min_{\pi \in \Pi} \int_{\mathcal{X} \times \mathcal{Y}} c\left(\mathbf{x}, \mathbf{y}\right) d\pi\left(\mathbf{x}, \mathbf{y}\right) \tag{4}$$

Under this formulation, the optimal π, which is a joint probability measure with marginals μ and ν, can be interpreted as the optimal transportation map. It allows the Wasserstein distance of order p between μ and ν to be defined as

$$W_p\left(\mu, \nu\right) \stackrel{\text{def}}{=} \inf_{\pi \in \Pi} \left\{ \left(\mathop{\mathbb{E}}_{\substack{\mathbf{x} \sim \mu \\ \mathbf{y} \sim \nu}} d\left(\mathbf{x}, \mathbf{y}\right)^p \right)^{\frac{1}{p}} \right\} \tag{5}$$

where $d(\cdot, \cdot)$ is a distance defining the cost function $c\left(\mathbf{x}, \mathbf{y}\right) = d\left(\mathbf{x}, \mathbf{y}\right)^p$. The Wasserstein distance is also known as the Earth mover's distance. It defines a metric over the space of square integrable probability measures.

2.2 Proposed relaxation of OT

OT boils down to a variational problem, i.e., it requires the minimization of an integral criterion in a class of admissible functions. Given two probability

measures μ and ν, the existence and uniqueness of an operator T that belongs to the class of bijective, continuous and differentiable functions such that $T_\sharp\mu = \nu$ is not guaranteed. The difficulty lies in the class defining these admissible functions. Indeed, even when μ and ν are regular densities on regular subsets of $\mathbb{R}^d$, the search for a transport map such that $T_\sharp\mu = \nu$ makes the problem (3) difficult in a general case. To overcome the difficulty of solving this equation on $T_\sharp$, we propose to reformulate the Monge's OT statement by relaxing the equality on the operator defining the image measure.

More precisely, the equality between the image measure $T_\sharp\mu$ and the target measure ν is replaced by the minimization of their statistical distance $d(T_\sharp\mu, \nu)$. The choice of the distance $d(\cdot,\cdot)$ is crucial because it determines the quality of the approximation of the image measure by the transport map T. In this work, we propose to choose $d(\cdot,\cdot)$ as the Wasserstein distance $W_p(\cdot,\cdot)$. This choice will be motivated by the fact that this distance can be easily approximated numerically without explicit knowledge of the probability distributions μ and ν, in particular when they are empirically described by samples only. The relaxation of the Monge problem (3) can then be written as

$$\inf_T \left\{ W_p(T_\sharp\mu, \nu) + \lambda \int_{\mathcal{X}} c(x, T(x)) \mathrm{d}\mu(x) \right\} \tag{6}$$

where the cost function defined in (3) is interpreted here as a regularisation term adjusted by the hyperparameter λ.

Remark 1. The relaxed formulation (6) relies on the Wasserstein distance between the target measure ν and the image measure $T_\sharp\mu$. This term should not be confused with the Wasserstein distance $W_p(\mu, \nu)$ which is the infimum reached by the solution of the Kantorovitch's formulation of OT (4).

2.3 Discrete formulation

In a machine learning context, the underlying continuous measures are conventionally approximated by empirical point measures thanks to available data samples. Therefore, in this paper, we are interested in discrete measures and the empirical formulation of the OT problem. Within this framework, we will consider μ and ν two discrete measures described by the respective samples $\mathbf{x} = \{x_n\}_{n=1}^N$ and $\mathbf{y} = \{y_n\}_{n=1}^N$ such that $\mu = \frac{1}{N}\sum_{n=1}^N \delta_{x_n}$ and $\nu = \frac{1}{N}\sum_{n=1}^N \delta_{y_n}$. In the following, an empirical version of the criterion (6) is proposed in the case of discrete measures.

The formulation (6) requires the evaluation of a Wasserstein distance whose computation is not trivial in its original form, especially in high dimension. An alternative consists in considering its rewriting in the form of the *sliced-Wasserstein* (SW) distance. The idea underlying the SW distance is to represent a distribution defined in high dimension thanks to a set of projected one-dimensional distributions for which the computation of the Wasserstein distance is closed-form. Let p_X and p_Y denote the probability distributions of the random

variables X and Y. For any vector on the unit sphere $u \in \mathbb{S}^{d-1}$, the projection operator $S_u : \mathbb{R}^d \to \mathbb{R}$ is defined as $S_u(x) \triangleq \langle u, x \rangle$. The SW distance of order $p \in [1, \infty)$ between p_X and p_Y can be written [4]

$$SW_p\left(p_X, p_Y\right) = \left(\int_{\mathbb{S}^{d-1}} W_p\left(S_{u\sharp} p_X, S_{u\sharp} p_Y\right)^p du\right)^{\frac{1}{p}} \tag{7}$$

where the distance $W_p(\cdot, \cdot)$ defining the integrand is now one-dimensional, leading to an explicit computation by inversion of the cumulative distribution functions. In the case where the distributions p_X and p_Y are represented by the respective samples $\mathbf{x}$ and $\mathbf{y}$, a numerical Monte Carlo approximation of the SW distance is

$$\widehat{SW}_p(\mathbf{x}, \mathbf{y}) = \frac{1}{J} \sum_{j=1}^{J} W_p\left(\frac{1}{N} \sum_{n=1}^{N} \delta_{S_{u_j}(x_n)}, \frac{1}{N} \sum_{n=1}^{N} \delta_{S_{u_j}(y_n)}\right) \tag{8}$$

where $u_1, \ldots, u_J$ are drawn uniformly on the sphere $\mathbb{S}^{d-1}$. The empirical form of the relaxation of the Monge problem (6) is then written as

$$\min_T \left\{\widehat{SW}_p(T(\mathbf{x}), \mathbf{y}) + \lambda \sum_{n=1}^{N} c\left(x_n, T\left(x_n\right)\right)\right\} \tag{9}$$

where, with a slight abuse of notations, $T(\mathbf{x}) \triangleq \{T(x_n)\}_{n=1}^N$.

3 Normalizing flows to approximate OT

This section proposes to solve the problem (9) by restricting the class of the operator T to a class of invertible deep networks referred to as normalisation flows. The structure and the main properties of these networks are detailed in paragraph 3.1. The strategy proposed to train these networks to solve the problem (9) is then detailed in paragraph 3.2.

3.1 Normalizing flows

Normalization flows are a flexible class of deep generative networks that intend to learn a change of variable between two probability distributions p_X and p_Y through an invertible transformation $T_{\boldsymbol{\Theta}} : X \mapsto Y = T_{\boldsymbol{\Theta}}(X)$ parametrized by $\boldsymbol{\Theta}$. In general, the distribution p_X is only known through samples $\mathbf{x} = \{x_n\}_{n=1}^N$ and, for tractability purpose, the distribution p_Y is chosen as a centered normal distribution with unit variance. The parameters $\boldsymbol{\Theta}$ defining the operator $T_{\boldsymbol{\Theta}}$ are then adjusted by maximizing the likelihood associated with the observations $\mathbf{x}$ according to the change of variable formula

$$p_X(x) = p_Y\left(T_{\boldsymbol{\theta}}(x)\right) \left|\det J_{T_{\boldsymbol{\theta}}^{-1}}\right| \tag{10}$$

with $J_{T_{\boldsymbol{\Theta}}^{-1}} = \frac{\partial T_{\boldsymbol{\theta}}^{-1}}{\partial x}$. NF networks obey a cell-like structure, explicitly defining the operator $T_{\boldsymbol{\Theta}}(\cdot)$ as the composition of M functions $T_{\boldsymbol{\theta}_m}^{(m)}$, usually referred to as *flows*, i.e.,

$$T_{\boldsymbol{\Theta}}(\cdot) = T_{\boldsymbol{\theta}_M}^{(M)} \circ T_{\boldsymbol{\theta}_{M-1}}^{(M-1)} \circ \ldots \circ T_{\boldsymbol{\theta}_1}^{(1)}(\cdot) \tag{11}$$

with $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$. In the following, to lighten notations, each sub-function contributing to the flow will be denoted by $T_m = T_{\boldsymbol{\theta}_m}^{(m)}$. In the present work, these functions are chosen as coupling layers as implemented by flows like RealNVP [10] and nonlinear independent component estimation (NICE) [9]. These coupling layers ensure an invertible transformation and an explicit expression of the Jacobian required in the change of variables (10). The input and output of the mth layer are related as $(y_{\text{id}}, y_{\text{ch}}) = T_m(x_{\text{id}}, x_{\text{ch}})$ with

$$\begin{cases} y_{\text{id}} = x_{\text{id}} \\ y_{\text{ch}} = (x_{\text{ch}} + \mathsf{D}_m(x_{\text{id}})) \odot \exp(\mathsf{E}_m(x_{\text{id}})) \end{cases} \tag{12}$$

where x_{id} and x_{ch} (resp. y_{id} and y_{ch}) are disjoint subsets of components of the input vector x (resp. the output vector y). The splitting of the input x into x_{id} and x_{ch} is achieved by a masking process such that $x_{\text{ch}} = \text{mask}(x)$ is transformed into a function of the unchanged part x_{id}. The scale function $\mathsf{E}_m(\cdot)$ and the offset function $\mathsf{D}_m(\cdot)$ are then described by neural networks whose parameters $\boldsymbol{\theta}_m$ need to be adjusted during the training. It is worth noting that imposing the flow-based architecture detailed in (11) will lead to an explicit discretization scheme of the transport map $T_{\boldsymbol{\Theta}}(\cdot)$ into a sequence of elementary transport functions $T_m(\cdot)$. As it will be shown in Section 3.3, this discretization has the great advantage of providing Wasserstein barycenters associated with the two measures μ and ν. Note that the proposed method is not limited to NFs composed of coupling layers such as RealNVP [10], NICE [9] or GLOW [14]. It can be generalized to other types of NFs, including free-form Jacobian of reversible dynamics (FFJORD) [12] and masked autoregressive flows (MAF) [20].

3.2 Loss function

As mentioned before, the objective of this work is to learn a bijective operator relating any two distributions p_X and p_Y described by samples $\mathbf{x}$ and $\mathbf{y}$. The search for this operator is restricted to the class of invertible deep networks $T_{\boldsymbol{\Theta}}$ described in paragraph 3.1. The conventional strategy to train the network would be to maximize the likelihood defined by (10). However this approach cannot be implemented in the context of interest here since the base distribution p_Y is no longer explicitly given: it is only available through the knowledge of the set of samples $\mathbf{y}$. As a consequence, to adjust the weights of the network, the proposed alternative interprets the underlying learning task as the search for a transport map. Then a first idea would be to adjust these weights by directly solving the problem (9). However, to take advantage of the flow-based architecture of the operator $T_{\boldsymbol{\Theta}}(\cdot)$, it seems legitimate to equally distribute the transport efforts

provided by each flow. Thus, the regularization in (9) will be instantiated for each elementary transformation $T_m(\cdot)$ associated to each flow of the network.

Moreover, when fitting deep learning-based models a major challenge arises from the stochastic nature of the optimization procedure, which imposes to use partial information (e.g., as mini-batches) to infer the whole structure of the optimization landscape. On top of that, the cost function to be optimized is not numerically constant since the approximation $\widehat{SW}$ of the SW distance in (9) depends on the precise set of random vectors $\{u_j\}_{j=1}^{J}$ drawn over the unit sphere. To alleviate these optimization difficulties, we propose to further regularize the objective function by penalizing the energy $|J_{T_m}(\cdot)|^2$ of the Jacobians associated with the transformations $T_m(\cdot)$, $m = 1, \ldots, M$. These Sobolev-like penalties promote regular operators $T_m(\cdot)$, promoting an overall operator $T_{\boldsymbol{\theta}}(\cdot)$ regular itself [13]. In the context of optimal transport, this regularization has already been studied in depth in [17]. In that work, the author focused on the penalization of the Monge's formulation of OT by the ℓ_2-norm of the Jacobian. It stated the existence of an optimal transport map T solving the minimization problem

$$\inf_{T} \left\{ \int_{\mathcal{X}} \left(|T(x) - x|^2 + \gamma |J_T|^2 \right) T(x) \mathrm{d}x : T_{\#}\mu = \nu \right\} \tag{13}$$

This formulation of OT imposes the transport map T to be regular rather than deducing its regularity from its optimal properties. Finally, the training of the NF is carried out by minimizing the loss function

$$\underbrace{\widehat{SW}_p(\mathbf{x}, \mathbf{y})}_{\text{SW}} + \underbrace{\sum_{n=1}^{N} \sum_{m=1}^{M} \left[\lambda c(T_{m-1}(x_n), T_m(x_n)) + \gamma \left| J_{T_m}(x_n) \right|^2 \right]}_{\text{Reg}} \tag{14}$$

with $T_0(x_n) = x_n$. The proposed network, whose general architecture is depicted in Fig. 1, will be referred to as SWOT-Flow in what follows.

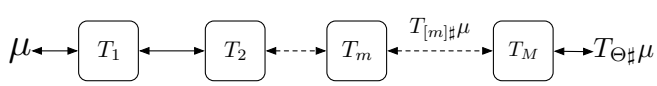


Fig. 1: Architecture of the proposed SWOT-Flow.

3.3 Intermediate transports and Wasserstein barycenters

As a consequence of the multiple-flow architecture (11) of the NF, the transport map operated by the proposed SWOT-Flow is a composition of the M individual flows $T_m(\cdot)$ ($m = 1, \ldots, M$). Thus each flow implements an elementary transport and the composition of the first m flows defined as

$$T_{[m]}(\cdot) \triangleq T_m \circ \ldots \circ T_1(\cdot) \tag{15}$$

can be interpreted as an intermediate step of the transport map from the input measure μ towards the target measure ν, with $T_{[M]}(\cdot) \triangleq T_{\boldsymbol{\Theta}}(\cdot)$. Interestingly, these intermediate transports can be related to Wasserstein barycenters between μ and ν defined by [1]

$$\inf_{\beta} \left\{ \alpha W_p(\mu, \beta) + (1-\alpha) W_p(\beta, \nu) \right\}. \tag{16}$$

Indeed, the next section dedicated to numerical experiments will empirically show that $T_{[m]\sharp}\mu$ approaches the solution of the problem (16) for the specific choice of the weight $\alpha = \frac{m}{M}$. In other words, the image measures provided by each intermediate transport operated by SWOT-Flow, i.e., as the outputs of each of the M flows, can legitimately be interpreted as Wasserstein barycenters.

4 Numerical experiments

This section assesses the versatility and the accuracy of SWOT-Flow through two sets of numerical experiments. First, several toy experiments are presented to provide some insights about key ingredients of the proposed approach. Then the performance of SWOT-Flow is illustrated through the more realistic and challenging task of unsupervised alignment of word embeddings in natural language processing. The source code is publicly available on GitHub [3].

4.1 Toy examples

In these experiments, the proposed framework SWOT-Flow is implemented and tested with synthetic data. In all experiments, the input distributions are described by the respective samples $\mathbf{x} = \{x_n\}_{n=1}^N$ and $\mathbf{y} = \{y_n\}_{n=1}^N$ such that $\mu = \frac{1}{N}\sum_{n=1}^N \delta_{x_n}$ and $\nu = \frac{1}{N}\sum_{n=1}^N \delta_{y_n}$ with $N = 20000$. The cost function $c(\cdot,\cdot)$ is chosen as the squared Euclidean distance, i.e., $c(x,y) = \|x-y\|_2^2$. However, it is worth noting that the proposed method is not limited to this Euclidean distance and can handle other costs defined on $\mathbb{R}^d$ or even on curved domains.

Implementation details. The stochastic gradient descent used to solve (14) is implemented in Pytorch. We use Adam optimizer with learning rate 10^{-4} and a batch size of 4096 or 8192 samples. The NF implementing $T_{\boldsymbol{\Theta}}(\cdot)$ is a RealNVP [10] for the example of Fig. 2 and an ActNorm type architecture network [14] for Fig. 3 and Fig. 4. It is composed of $M = 4$ flows, each composed of two four-layer neural networks corresponding to $\mathsf{D}_m(\cdot)$ and $\mathsf{E}_m(\cdot)$ ($d \to 8 \to 8 \to d$) using hyperbolic tangent activation function. During training, the number J of slices drawn to approximate the SW distance in (8) has been progressively increased, starting from $J = 500$ to $J = 2000$ by step of 50 slices. At each epoch, new slices are uniformly drawn over the unit sphere and 100 epochs are carried out for each number of slices. The training procedure consist in 1) defining the loss function

[3] FlorentinCDX/SWOT-Flow

as the sole SW term in (14) from $J = 500$ to 1500 slices and then 2) incorporating the regularization term denoted as Reg in (14) where hyperparameters λ and γ are increased by a factor of 5% every step of 100 slices.

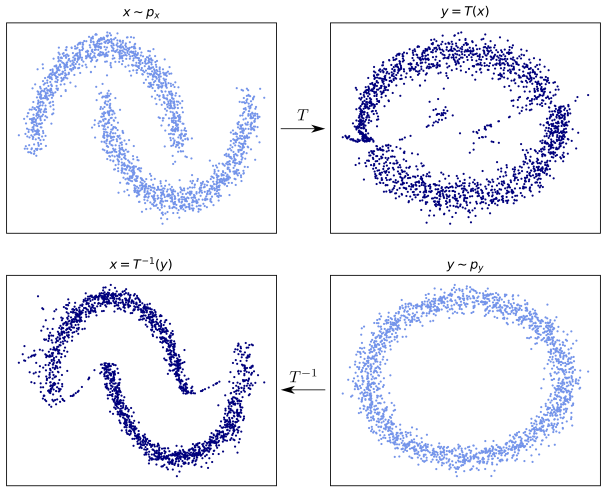


Fig. 2: Operator T learnt by SWOT-Flow when the base distribution p_X is a double-moon (top left) and the target distribution p_Y is a circle (bottom right).

Qualitative results. As a first illustration of the flexibility of the proposed approach, Fig. 2 shows the results obtained after learning an operator T that transports a double moon-shaped distribution p_X (top left) to a circle-shaped distribution (bottom right). The empirical image measures $T_\sharp p_X$ (top right) and $T_\sharp^{-1} p_Y$ (bottom left) are obtained by applying the estimated $T(\cdot)$ operator or its inverse $T^{-1}(\cdot)$. It is worth noting that the difficulty inherent to this experiment lies in the respective disjoint and non-disjoint supports of the two distributions. Despite the regularity of the trained NF, a very good approximation of the OT is learnt, even in presence of this topological change.

Fig. 3 aims at illustrating the relevance of the Sobolev-like regularization (i.e., the ℓ_2-norm of the Jacobian) included into the loss function (14) defined to train the NF. The first simulation protocol considers circle-shaped distributions while the second case considers rectangle-shaped distributions. In what follows, these two cases will be referred to as $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. In this experiment, the objective is to learn the transport map from an initial distribution p_X (light blue) to a target distribution p_Y (dark blue) which is translated for $\mathcal{P}_1$ and both translated and stretched for $\mathcal{P}_2$. The color gradient shows the outputs of the M successive flows of the network, i.e. the image measures $T_{[m]\sharp} p_X$ for $m = 1, \ldots, M$. In the absence of regularization (left), the successive elementary

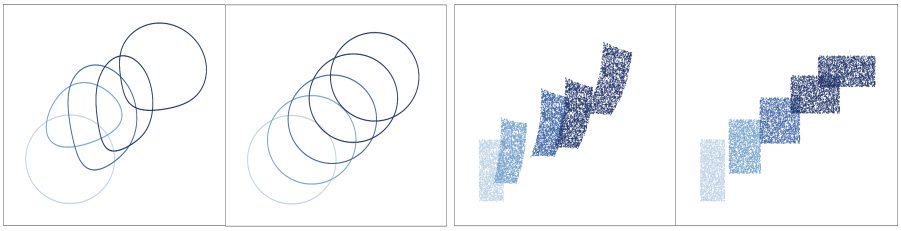


Fig. 3: Elementary transports achieved by the proposed NF when trained without (1st and 3rd panels) or with (2nd and 4th panels) the regularization for protocols $\mathcal{P}_1$ (left panels) and $\mathcal{P}_2$ (right panels).

transports clearly suffer from multiple unexpected deformations (superfluous translations and dilations). In contrast, when the loss is complemented with the proposed Sobolev-type penalty (right), the learnt operator T is decomposed as a sequence of much more regular elementary transports. The resulting transport appears to be very close to optimal. In case $\mathcal{P}_1$, the expected translation is recovered, as well as the combined translation and stretching in case $\mathcal{P}_2$.

Table 1: Overall cost $\bar{C}$ and elementary costs $\bar{c}_m$ required by each flow $T_m(\cdot)$ of the NF trained with or without (w/o) regularization for protocols $\mathcal{P}_1$ (circle-shaped distributions) and $\mathcal{P}_2$ (rectangle-shaped distributions).

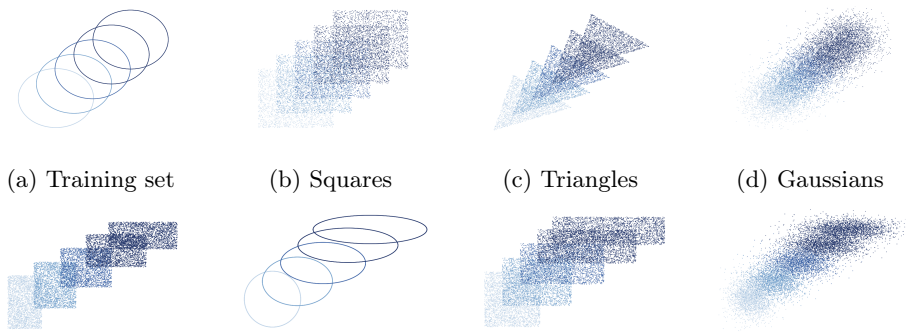
		$\bar{c}_1$	$\bar{c}_2$	$\bar{c}_3$	$\bar{c}_4$	$\bar{C}$
$\mathcal{P}_1$	w/o regularization	150.13	110.94	108.41	151.65	521.12
	with regularization	90.20	90.70	90.71	90.22	361.22
$\mathcal{P}_2$	w/o regularization	154.99	98.67	52.49	101.21	407.38
	with regularization	88.77	89.42	89.43	89.38	357.0

To be more precise quantitatively, Table 1 compares some metrics obtained when the NF has been trained using the regularization-free or regularized loss function, as defined in (14). For the two aforementioned simulation protocols, it reports the elementary costs

$$\bar{c}_m = \frac{1}{N} \sum_{n=1}^{N} \|T_{m-1}(x_n) - T_m(x_n)\|_2^2 \tag{17}$$

spent by each of the M flows $T_1(\cdot), \ldots, T_M(\cdot)$ to achieve the transport maps retrieved by SWOT-Flow. This table (last column) also reports the overall cost $\bar{C} = \sum_{m=1}^{M} \bar{c}_m$. For the two simulation protocols $\mathcal{P}_1$ and $\mathcal{P}_2$, these results clearly show cheaper transports when using the proposed regularization. For instance,

for the simulation protocol $\mathcal{P}_1$, the overall cost is $\bar{C} = 360$ with the regularization, compared to $\bar{C} = 520$ when it is omitted. Moreover, when using the regularized loss function, this cost is distributed homogeneously over the successive flows, with a variation of at most $\pm 1\%$ from one flow to another, against $\pm 20\%$ otherwise.



(a) Training set (b) Squares (c) Triangles (d) Gaussians

(e) Training set (f) Circles (g) Squares (h) Gaussians

Fig. 4: Examples of transported data sets for protocols $\mathcal{P}_1$ (top) and $\mathcal{P}_1$ (bottom).

Fig. 4 aims at illustrating the capacity of generalization of the transport map learnt by SWOT-Flow. In this experiment, SWOT-Flow has been trained following the simulation protocols $\mathcal{P}_1$ (Fig. 4a) or $\mathcal{P}_2$ (Fig. 4e). Once trained on the data set associated with each protocol, the NFs are fed with differently shaped data and the elementary transports are monitored as above. Fig 4b-4d and 4g-4h show the results when using square-, triangle-, Gaussian-shaped data sets for both protocols, respectively. As expected, all initial distributions are either simply translated in case $\mathcal{P}_1$ or translated and stretched in case $\mathcal{P}_2$. The intermediate distributions correspond to the expected barycenters as well. Fig. 4 clearly demonstrates the generalization capacity of the proposed approach.

Multivariate Gaussians with varying dimensions. When the source and target distributions μ and ν of a transportation problem are multivariate Gaussians, the Wasserstein barycenters defined by (16) are also multivariate Gaussian distributions. In this case, an efficient fixed-point algorithm can be used to estimate its mean vector $\mathbf{a}$ and covariance matrix $\boldsymbol{\Sigma}$ [11]. This experiment capitalizes on this finding to assess the ability of SWOT-Flow to approximate Wasserstein barycenters, as stated in Section 3.3. To this end, the algorithm designed in [11] is implemented to estimate the actual barycenter associated with two prescribed multivariate Gaussian distributions for $\alpha = 1 - \alpha = \frac{1}{2}$. This barycenter is compared to the image measure $T_{[m]\sharp}\mu$ estimated by SWOT-Flow with $m = \frac{M}{2}$. More precisely, the mean vector and the covariance matrix of the barycenter

are compared to their maximum likelihood estimates $\hat{\mathbf{a}}$ and $\hat{\boldsymbol{\Sigma}}$ computed from the samples $\{T_{[m]}(x_n)\}_{n=1}^N$ transported by the first m flows. The resulting mean square errors (MSEs)

$$\mathrm{MSE}(\mathbf{a}) = \|\mathbf{a} - \hat{\mathbf{a}}\|_2^2 \quad \text{and} \quad \mathrm{MSE}(\boldsymbol{\Sigma}) = \|\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}\|_{\mathrm{F}}^2 \tag{18}$$

are reported in Table 2 for varying dimensions ranging from 2 to 8. This table also reports the MSEs reached by other state-of-the-art free-support methods [8,5,16]. For the methods [8] and [5], $n = 5000$ and $n = 100$ support points have been used, respectively, since these are the maximum numbers allowed for the algorithms to terminate in reasonable computational times. SWOT-Flow compares favorably to state-of-the-art methods since reported MSEs in Table 2 appear to be most often the smallest. These observation may call for a more general study, but remains noticeable since SWOT-Flow has not been specifically designed to compute the Wasserstein barycenters, contrary to alternate methods.

Table 2: Performance of the estimation of the median barycenters. Reported scores result from the average over 5 Monte Carlo runs.

			[8]	[5]	[16]	SWOT-Flow
Dimension	2	MSE($\mathbf{a}$)	$9.99 \cdot 10^{-5}$	$3.14 \cdot 10^{-4}$	$1.17 \cdot 10^{-4}$	$\mathbf{8.09 \cdot 10^{-5}}$
		MSE($\boldsymbol{\Sigma}$)	$7.28 \cdot 10^{-4}$	$2.39 \cdot 10^{-3}$	$1.98 \cdot 10^{-3}$	$\mathbf{1.44 \cdot 10^{-4}}$
	4	MSE($\mathbf{a}$)	$1.73 \cdot 10^{-3}$	$1.68 \cdot 10^{-3}$	$1.44 \cdot 10^{-3}$	$\mathbf{1.44 \cdot 10^{-4}}$
		MSE($\boldsymbol{\Sigma}$)	$1.35 \cdot 10^{-2}$	$2.50 \cdot 10^{-2}$	$1.22 \cdot 10^{-2}$	$\mathbf{3.61 \cdot 10^{-4}}$
	6	MSE($\mathbf{a}$)	$2.04 \cdot 10^{-3}$	$\mathbf{2.58 \cdot 10^{-3}}$	$3.24 \cdot 10^{-3}$	$1.23 \cdot 10^{-2}$
		MSE($\boldsymbol{\Sigma}$)	$4.38 \cdot 10^{-2}$	$8.86 \cdot 10^{-2}$	$2.37 \cdot 10^{-2}$	$\mathbf{5.29 \cdot 10^{-4}}$
	8	MSE($\mathbf{a}$)	$\mathbf{1.23 \cdot 10^{-3}}$	$1.48 \cdot 10^{-3}$	$3.14 \cdot 10^{-3}$	$1.29 \cdot 10^{-2}$
		MSE($\boldsymbol{\Sigma}$)	$8.31 \cdot 10^{-2}$	$1.64 \cdot 10^{-1}$	$4.23 \cdot 10^{-2}$	$\mathbf{2.22 \cdot 10^{-3}}$

4.2 Unsupervised word translation

In a second set of experiments, the performance of SWOT-Flow has been assessed on the task of unsupervised word translation. Given word embeddings trained on two monolingual corpora, the goal is to infer a bilingual dictionary by aligning the corresponding word vectors.

Experiment description. This experiment considers the task of aligning two sets of points in high dimension. More precisely, it aims at inferring a bilingual lexicon, without supervision, by aligning word embeddings trained on monolingual data. FastText [3] has been implemented to learn the word vectors used for representation. It provides monolingual embeddings of dimension 300 trained on Wikipedia corpora. Words are lower-cased, and those that appear less than 5 times are discarded for training. As a post-processing step, only the first 50k most frequent words are selected in the reported experiments.

Architecture. The proposed SWOT-Flow method has been implemented using a RealNVP architecture. The scale function $\mathsf{E}_m(\cdot)$ and the offset function $\mathsf{D}_m(\cdot)$ are multilayer neural networks with two hidden layers of size 512 and hyperbolic tangent activation function. Adam has been used as an optimizer with a learning rate of $1 \cdot 10^{-3}$. The number of slices involved in the Monte Carlo approximation of the SW distance in (8) has been progressively increased from $J = 500$ slices to $J = 3000$ by steps of 50. For each number of slices, 100 epochs have been performed. The hyperparameters λ and γ adjusting the weights of the composite regularization have been increased by a factor of 5% every steps of 500 slices.

Table 3: Comparison of accuracies obtained by SWOT-Flow and adv-net [6] for unsupervised word translation ('en' is English, 'fr' is French, 'de' is German, 'ru' is Russian).

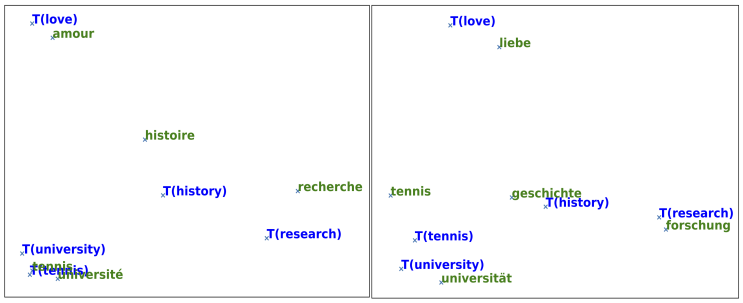
Method		en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en
SWOT-Flow	20-NN	37.4	24.2	46.6	34.1	44.4	27.6	14.4	3.8
	10-NN	**33.5**	**22.5**	**42.5**	32.5	39.5	26.8	**10.2**	2.1
adv-net [6]	10-NN	31.4	21.2	39.6	**35.1**	**40.1**	**27.1**	7.1	**2.3**

Main results. To quantitatively measure the quality of SWOT-Flow, the problem of bilingual lexicon induction is addressed, with the same setting as in [6]. The same evaluation data sets and codes, as well as the same word vectors have been used. Given an input word embedding ($n = 1, \ldots, N$ with $N_{\text{test}} = 1000$) in a given language, the objective is to assess if its counterpart $T(x_n)$ transported by SWOT-Flow belongs to the close neighborhood of the output word embedding y_n in the target language. The neighborhood $\mathcal{V}(y_n)$ is defined as the set of K-nearest neighbors computed in a cosine similarity sense with $K = 10$ or 20 in dimension 300. The overall accuracy is computed as the percentage of correctly transported input samples. Denoting by $\mathbf{1}_{\mathsf{A}}$ the indicator function, i.e., $\mathbf{1}_{\mathsf{A}} = 1$ if the assertion A is true and $\mathbf{1}_{\mathsf{A}} = 0$ otherwise,

$$\text{accuracy} = \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \mathbf{1}_{\{T(x_n) \in \mathcal{V}(y_n)\}} \times 100 \ (\%) \tag{19}$$

Table 3 reports the accuracy scores for several pairs of languages. Although SWOT-Flow has not been specifically designed to perform word translation, these results show that its overall performance is on par with the adversarial network (adv-net) proposed specifically for this task in [6]. In particular, SWOT-Flow seems to perform well for translation between languages with close origins.

Fig. 5 qualitatively illustrates this good performance by showing how close a set of translated words $T(x_n)$ are to their true translation y_n. This representation is obtained by a classical projection on the 2 first PCA components of the target



(a) english → french (b) english → german

Fig. 5: 2D PCA representation of the target word embedding space: the targeted translated (in green) and the transported source (in blue) embedded words.

embedded space. The translation of 5 specific words from English to French or German fall in the close vicinity of their true counterparts.

5 Discussion

Cycle consistency. Cycle consistency, as proposed in CycleGAN [23], aims at learning meaningful cross-domain mappings such that the data translated from the domain $\mathcal{X}$ to the domain $\mathcal{Y}$ via $T_{\mathcal{X}\to\mathcal{Y}}$ can be mapped back to the original data points in $\mathcal{X}$ via $T_{\mathcal{Y}\to\mathcal{X}}$. That is, $T_{\mathcal{Y}\to\mathcal{X}} \circ T_{\mathcal{X}\to\mathcal{Y}}(x) \approx x$ for all $x \in \mathcal{X}$. For CycleGan, and many other domain transfer models such as [2], this key property should be enforced by including a cycle consistency term into the loss function. Conversely, since NF-based generative models learn bijective mappings, the proposed SWOT-Flow inherits the cycle consistency property by construction.

Semi-discrete formulation. The proposed SWOT-Flow framework has been explicitly derived to approximate OT between two discrete empirical distributions. It can be instanciated to perform semi-discrete OT, i.e., to handle the case where one of distribution is not described by data points but rather given as an explicit continuous probability measure. Instead of relaxing the Monge formulation (3) as in (6), it would consist in replacing the SW distance with a log-likelihood term $\log f_\nu(\cdot)$ associated with the target continuous measure. The loss function in (14) would be replaced by

$$-\sum_{n=1}^{N} \log f_\nu(T_{\boldsymbol{\Theta}}(x_n)) + \sum_{n=1}^{N}\sum_{m=1}^{M} \left[\lambda c(T_{m-1}(x_n), T_m(x_n)) + \gamma \left|J_{T_m}(x_n)\right|^2\right] \quad (20)$$

where the log-likelihood term is evaluated at the data points $\{T_{\boldsymbol{\Theta}}(x_n)\}_{n=1}^{N}$ transported by the NF.

NF to approximate barycenters. As discussed in Section 3.3 and experimentally illustrated in Section 4.1, the flow-based architecture of the SWOT-Flow network leads to intermediate transports, that can be related to Wasserstein barycenters. On the toy Gaussian example considered in Section 4.1, SWOT-Flow provides good approximation of the barycenters, although it has not been specifically designed to perform this task. If one is interested in devising a NF approximating these barycenters, the definition (16) would lead to the optimization problem

$$\inf_T \left\{ \sum_{m=1}^{M} \alpha_m W_p(\mu, T_{[m]\sharp}\mu) + (1-\alpha_m) W_p(T_{[m]\sharp}\mu, \nu) \right\} \tag{21}$$

with $\alpha_m = \frac{m}{M}$. When handling empirical measures described by samples, the subsequent discretization would require to replace both terms with Monte Carlo approximations (8) of the SW distances. However, this would lead to a computationally demanding training procedure.

6 Conclusion

We propose a new method to learn the optimal transport map between two empirical distributions from sets of available samples. To this aim, we write a relaxed and penalized formulation of the Monge problem. This formulation is used to build a loss function that balances between the cost of the transport and the proximity in Wasserstein distance between the transported base distribution and the target one. The proposed approach relies on normalizing flows, a family of invertible neural networks. Up to our knowledge, this is the first method that is able to learn such a generalizable transport operator. As a side benefit, the multiple flow architecture of the proposed network interestingly yields intermediate transports and Wasserstein barycenters. The proposed method is illustrated by numerical experiments on toy examples as well as an unsupervised word translation task. Future work will aim at extending these results to high dimensional applications.

References

1. Agueh, M., Carlier, G.: Barycenters in the Wasserstein space. SIAM J. Mathematical Analysis **43**(2), 904–924 (2011)
2. de Bézenac, E., Ayed, I., Gallinari, P.: Cyclegan through the lens of (dynamical) optimal transport. In: Joint Eur. Conf. Machine Learning and Knowledge Discovery in Databases (ECML-PKDD) (2021)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics (2017)
4. Bonneel, N., Rabin, J., Peyré, G., Pfister, H.: Sliced and radon Wasserstein barycenters of measures. J. Math. Imag. Vision **51**(1), 22–45 (2015)

5. Claici, S., Chien, E., Solomon, J.M.: Stochastic Wasserstein barycenters. In: Proc. Int. Conf. Machine Learning (ICML) (2018)
6. Conneau, A., Lample, G., Ranzato, M., Denoyer, L., Jégou, H.: Word translation without parallel data. In: Proc. IEEE Int. Conf. Learn. Represent. (ICLR) (2018)
7. Courty, N., Flamary, R., Tuia, D.: Domain adaptation with regularized optimal transport. In: Joint Eur. Conf. Machine Learning and Knowledge Discovery in Databases (ECML-PKDD) (2014)
8. Cuturi, M., Doucet, A.: Fast computation of wasserstein barycenters. In: Xing, E.P., Jebara, T. (eds.) Proc. Int. Conf. Machine Learning (ICML) (2014)
9. Dinh, L., Krueger, D., Bengio, Y.: NICE: non-linear independent components estimation. In: Bengio, Y., LeCun, Y. (eds.) Proc. IEEE Int. Conf. Learn. Represent. (ICLR) (2015)
10. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real NVP. In: Proc. IEEE Int. Conf. Learn. Represent. (ICLR) (2017)
11. Álvarez Esteban, P.C., del Barrio, E., Cuesta-Albertos, J., Matrán, C.: A fixed-point approach to barycenters in wasserstein space. Journal of Mathematical Analysis and Applications (2016)
12. Grathwohl, W., Chen, R.T.Q., Bettencourt, J., Sutskever, I., Duvenaud, D.: Ffjord: Free-form continuous dynamics for scalable reversible generative models. Proc. IEEE Int. Conf. Learn. Represent. (ICLR) (2019)
13. Hoffman, J., Roberts, D.A., Yaida, S.: Robust learning with jacobian regularization. arXiv (2020)
14. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: Adv. in Neural Information Process. Systems (NeurIPS) (2018)
15. Kobyzev, I., Prince, S.J., Brubaker, M.A.: Normalizing flows: An introduction and review of current methods. IEEE Trans. Patt. Anal. Mach. Intell. **43**(11), 3964–3979 (2020)
16. Korotin, A., Li, L., Solomon, J., Burnaev, E.: Continuous wasserstein-2 barycenter estimation without minimax optimization. In: Proc. IEEE Int. Conf. Learn. Represent. (ICLR) (2021)
17. Louet, J.: Problèmes de transport optimal avec pénalisation en gradient. Ph.D. thesis, Université Paris-Sud, France (2014)
18. Monge, G.: Mémoire sur la théorie des déblais et des remblais. Histoire de l'Académie Royale des Sciences de Paris (1781)
19. Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. J. Mach. Learning Research **22**(57), 1–64 (2021)
20. Papamakarios, G., Pavlakou, T., Murray, I.: Masked autoregressive flow for density estimation. In: Adv. in Neural Information Process. Systems (NeurIPS) (2017)
21. Paulin, L., Bonneel, N., Coeurjolly, D., Iehl, J.C., Webanck, A., Desbrun, M., Ostromoukhov, V.: Sliced optimal transport sampling. ACM Trans. Graphics (Proc. SIGGRAPH) (2020)
22. Peyré, G., Cuturi, M.: Computational optimal transport: with applications to data science. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019)
23. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proc. IEEE Int. Conf. Computer Vision (ICCV) (2017)