Optimal Transport for the colorization of old pictures: Project Report

Marwen Khelifa
marwen.khelifa@ensae.fr

Enzo Lounes
enzo.lounes@ensae.fr

Abstract

Optimal transport has emerged as a powerful framework for analyzing and processing probability distributions in machine learning. By incorporating the geometric properties of distributional spaces, OT enables meaningful comparisons and interpolations that are crucial in tasks ranging from image processing to generative modeling. This report introduces a new approach to histogram regression using Wasserstein Barycentric Coordinates (WBC). It builds upon the concept of Wasserstein Barycenters, enabling geometry-aware interpolation between probability distributions based on dynamic weights derived from the input data. By respecting the underlying geometry of the Wasserstein space, WBC provides a principled and interpretable framework for regression tasks where the outputs are histograms or distributions. We discuss the theoretical foundations of WBC, its computational challenges, and its benefits over traditional methods. Our approach is then tested on synthetic and real-world datasets.

1 Introduction

The field of optimal transport (OT) has become increasingly prominent in machine learning, providing a powerful framework for comparing and interpolating probability distributions. Rooted in the 18th-century problem posed by Gaspard Monge and later expanded by Leonid Kantorovich, optimal transport aims to determine the most efficient way to transform one distribution into another while minimizing a cost function. This concept has found various applications in machine learning thanks to its ability to handle complex data.

In many machine learning tasks, data is represented as probability distributions or histograms, especially in applications involving natural language processing, computer vision, and generative modeling. For instance, in image processing, the pixel intensity distributions of two images can be compared using OT metrics such as the Wasserstein distance. Unlike traditional divergence measures, which often assume overlap in distributional support or rely on parametric forms, OT metrics explicitly account for the geometry of the underlying space. This makes them particularly suitable for high-dimensional, structured, or sparse data.

One of the primary advantages of OT in machine learning lies in its flexibility and interpretability. The Wasserstein distance, provides a meaningful notion of closeness between distributions by considering the cost of transporting probability mass. This capability has fueled its adoption in a variety of applications, such as domain adaptation, generative adversarial networks (GANs), and clustering. For example, Wasserstein GANs make use of the Wasserstein distance to stabilize training and provide better gradients compared to classical GANs.

Despite its theoretical appeal, the computational complexity of OT has long been an issue. Classical OT formulations involve solving a linear program with $O(N^3)$ complexity, making it impractical for large-scale datasets. Recent advances, such as entropy-regularized OT, have significantly reduced computational overhead by relaxing the original problem. Techniques such as the Sinkhorn algorithm approximate OT distances efficiently, enabling their application in high-dimensional machine learning

problems. However, these approximations often trade off exactness for speed and may struggle with certain properties of the data.

The notion of Wasserstein Barycenters, central points that minimize the average Wasserstein distance to given distributions, has emerged as a critical tool for regression and interpolation tasks in Euclidian or probability distribution spaces. Wasserstein Barycenters allow for meaningful interpolation between distributions and have been applied in areas like style transfer, distributional clustering, and statistical summarization.

After a presentation of the methods we will be using for this project, we will present in this report two applications of barycentric coordinate regression: one on synthetic data and one on real-life data.

2 Wasserstein Barycentric Coordinates: Histogram Regression Using Optimal Transport

2.1 Motivation

Traditional regression methods struggle when the target space consists of probability distributions. Techniques that treat histograms as simple vectors ignore the inherent geometric structure of the space, potentially leading to predictions that violate basic probabilistic principles (e.g., non-negativity, normalization). Moreover, Euclidean metrics, such as L^2 distances, fail to capture meaningful differences between distributions, especially when their supports differ. This can result in poor model interpretability and degraded performance in applications where the transport cost or spatial relationships between bins are crucial.

Optimal transport addresses these issues by providing a geometry-aware framework for comparing and interpolating distributions. Wasserstein Barycenters, central elements in OT, generalize the concept of "averages" to the space of distributions. They minimize the Wasserstein distance, a metric that considers the cost of transporting mass, between a given set of distributions. Extending this idea, Wasserstein Barycentric Coordinates define a convex combination of distributions in the Wasserstein space. This formulation allows for a more nuanced representation of histogram data, enabling interpolation and regression tasks that respect the underlying geometry.

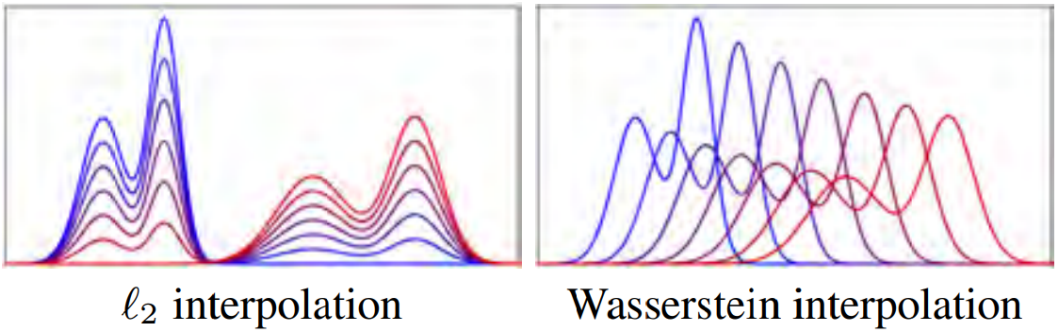

Figure 1: Difference between Euclidean and OT barycenters between two densities

2.2 Wasserstein Barycenters

Wasserstein Barycentric Coordinates are built on the concept of Wasserstein Barycenters. Given a set of target distributions $\{p_1, p_2, \ldots, p_s\} \subset \mathbb{R}^N$ and corresponding weights $\{\lambda_1, \lambda_2, \ldots, \lambda_s\} \subset \mathbb{R}$ which satisfy the constraints $\sum_{i=1}^{s} \lambda_i = 1$ and $\lambda_i \geq 0$, the Wasserstein Barycenter p_B is defined as the distribution that minimizes the weighted sum of Wasserstein distances:

$$\tilde{P}(\lambda) = \arg\min_{p} \sum_{i=1}^{s} \lambda_i \tilde{W}(p, p_i),$$

where $\tilde{W}$ denotes the Wasserstein distance:

$$\tilde{W}(p,q) = \min_{T \in \mathbb{R}_+^{NxN}} \{\langle T, C\rangle \ ; \ T\mathbf{1} = p, \ T^\top \mathbf{1} = q\}.$$

Here, $\langle T, C\rangle = tr(T^\top C)$ is the usual inner-product for two matrices of the same size, $\mathbf{1}$ is the vector with all coefficients equal to 1 (its size depending on the context), and C $\in \mathbb{R}^{N\times N}$ is the cost matrix, where the coefficient $C_{i,j}$ quantifies the cost of transporting mass from the i-th histogram bin to the j-th histogram bin.

Since this optimization problem is computationally too expensive, we instead define the entropy-regularized problem:

$$P(\lambda) = \arg\min_p \sum_{i=1}^{s} \lambda_i W(p, p_i),$$

with:

$$W(p,q) = \min_{T \in \mathbb{R}_+^{NxN}} \{\langle T, C\rangle + \gamma H(T) \ ; \ T\mathbf{1} = p, \ T^\top \mathbf{1} = q\},$$

where $\gamma > 0$ is the regularization parameter and $H(T) = \sum_{i,j} T_{i,j} log(T_{i,j})$ is the negative entropy of a matrix with the convention 0log(0) = 0. This regularization makes the optimal transport much easier to solve, as it allows us to use the Sinkhorn algorithm.

Algorithm 1 Sinkhorn Algorithm for Entropy-Regularized Optimal Transport

Require: Cost matrix $C \in \mathbb{R}^{N\times N}$, marginal distributions $p \in \mathbb{R}^N$, $q \in \mathbb{R}^N$, regularization parameter $\gamma > 0$, tolerance $\delta > 0$.
Ensure: Optimal transport plan T.
1: Compute the kernel: $K \leftarrow \exp(-C/\gamma)$.
2: Initialize scaling vectors: $u \leftarrow \mathbf{1}_n$, $v \leftarrow \mathbf{1}_n$.
3: **repeat**
4: Update $v \leftarrow q/(K^\top u)$.
5: Update $u \leftarrow p/(Kv)$.
6: **until** convergence: $\|u \cdot (Kv) - p\|_1 < \delta$ and $\|v \cdot (K^\top u) - q\|_1 < \delta$.
7: Compute transport plan: $T \leftarrow \text{diag}(u)K\text{diag}(v)$.

Benamou et al [2015] have demonstrated that $P(\lambda)$ can be estimated through the following Sinkhorn fixed-point iteration algorithm:

$$\textbf{Define for all } s \leq S, a_s^{(0)} = \mathbf{1} \textbf{ and then recursively for } l > 0, s \leq S:$$

$$P^{(l)}(\lambda) = \Pi_s(K^\top a_s^{(l)})^(\lambda_s), \qquad b_s^{(l+1)} = \frac{P^{(l)}(\lambda)}{K^\top a_s^{(l)}}, \qquad a_s^{(l+1)} = \frac{p_s}{K^\top b_s^{(l+1)}}$$

2.3 Barycentric Coordinate Regression

The core objective of our work is to estimate for a family of reference histograms $(p)_{s=1}^S$ and a given histogram q the vector of weights λ which minimizes a loss function $\mathcal{L}$ between $P(\lambda)$ and q, so that we can obtain from $(p)_{s=1}^S$ a histogram such that $P(\lambda) \approx q$. Formally, we want to solve this optimization problem:

$$\underset{\lambda \in \Sigma_s}{\textbf{argmin}}\ \mathcal{E}(\lambda) \qquad where \quad \mathcal{E}(\lambda) = \mathcal{L}(P(\lambda), p)$$

Using the chain rule, the gradient of $\mathcal{E}$ with respect to λ writes as follow:

$$\nabla\mathcal{E}(\lambda) = [\delta P(\lambda)]^T \nabla\mathcal{L}(P(\lambda, q))$$

In the case of a simple and convex loss function (we use the quadratic loss in our implementations), $\nabla\mathcal{L}(P(\lambda), q)$ is quite easy to compute, so the main difficulty here is to compute the transpose of the Jacobian $[\delta P(\lambda)]^\top$. The article [Bonneel, Peyré, Cuturi, 2016] presents a method that replaces the barycenter $P(\lambda)$ by an iterative estimate $p^L(\lambda)$ obtained after L iterations, which is far more practicable.

Algorithm 2 Sinkhorn-Differentiate Algorithm

Require: $(p_s)_{s=1}^S$, q,λ
Ensure: $P^{(L)}(\lambda)$, $\nabla\mathcal{E}^{(L)}(\lambda)$
Compute the kernel: $K \leftarrow \exp(-C/\gamma)$.
$\forall s, b_s^{(0)} \leftarrow \mathbf{1}$
$(w, r) \leftarrow (0^S, 0^{S\mathbf{x}N})$
for $l = 1, 2, ..., L$ **do**
  $\forall s, \phi_s^{(l)} \leftarrow K^\top \frac{p_s}{Kb_s^{(l-1)}}$
  $p \leftarrow \Pi_s(\phi_s^{(l)})^{\lambda_s}$
  $\forall s, b_s^{(l)} \leftarrow \frac{p_s}{\phi_s^{(l)}}$
end for
$g \leftarrow \nabla\mathcal{L}(p, q) \cdot p$
for $l = 1, 2, ..., L$ **do**
  $\forall s, w_s \leftarrow w_s + < log\phi_s^{(l)}, g >$
  $\forall s, r_s \leftarrow -K^\top(K(\frac{\lambda_s g - r_s}{\phi_s^{(l)}} \cdot \frac{p_s}{(Kb_s^{(l-1)})^2}) \cdot b_s^{(l-1)}$
  $g \leftarrow \Sigma_s r_s$
end for
$P^{(L)}(\lambda) \leftarrow p, \nabla\mathcal{E}^{(L)}(\lambda) \leftarrow w$

2.4 Benefits of Wasserstein Barycentric Coordinates

- **Geometry-Aware Interpolation:** WBC respects the geometry of the Wasserstein space, ensuring that interpolated distributions are meaningful.
- **Interpretability:** The weights λ_i provide a quick and easy way to check how each anchor distribution contributes to the predicted distribution, making the regression process more transparent.
- **Flexibility:** By varying the weights dynamically with input features, WBC enables regression models to adapt to complex relationships between inputs and target distributions.
- **Robustness:** WBC can handle distributions with non-overlapping supports or sparse representations, a common scenario in real-world histogram data.

2.5 Limitations and Challenges

- **Computational Complexity:** Computing Wasserstein Barycenters and distances is computationally expensive, particularly for high-dimensional data or large numbers of anchor distributions. While entropy-regularized OT methods (e.g., Sinkhorn iterations) somewhat fix this issue, they introduce approximation errors that may affect model accuracy.
- **Choice of Anchor Distributions:** The effectiveness of WBC depends on the selection of anchor distributions $\{p_s\}$. Poorly chosen anchors may fail to capture the diversity or structure of the target distributions, limiting the model.
- **Scalability:** Scaling WBC to very large datasets or real-time applications remains a challenge due to the complexity of OT-based operations.
- **Hyperparameter Sensitivity:** The regularization parameter γ and the number of anchor distributions can significantly impact performance.

3 First tests on synthetic data

3.1 Objective

Before starting our main application for the colorization of old pictures, we wanted to run simpler tests first on monochrome black and white pictures. Our main objective here is to replicate the Wasserstein symplex presented on the figure 3 of the article [Bonneel, Peyré, Cuturi, 2016].

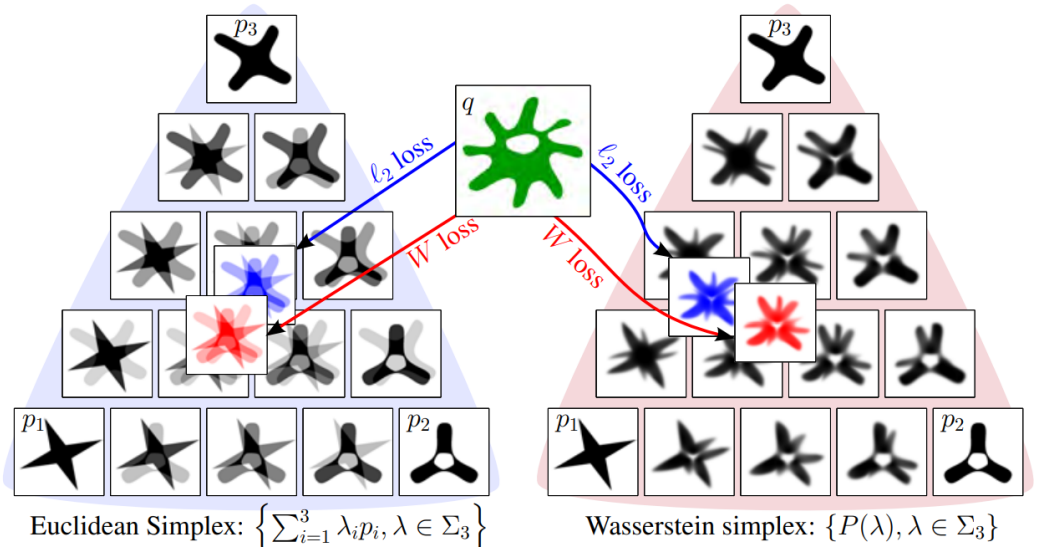

Figure 2: Euclidean symplex (left) and Wasserstein Symplex (right) for three monochrome images

We aim to form a similar symplex than the one on the right, with three 100x100 monochrome pictures of shapes (square, triangle and circle) generating the symplex and another shape (pentagon) as the target distribution which we want to project on the Wasserstein symplex. The 4 pictures were made with Paint:

3.2 Implementation

The algorithm for computing the Wasserstein barycenter and the one for computing the gradient of the loss between the barycenter and the target distribution q can be implemented in a fairly simple manner in Python. Here, each 100x100 monochrome image is represented by a histogram of size $100^2 = 10000$ and normalized. For this application and the following one with real data, we will be using the L^2 cost matrix:

$$C_{i,j} = ||x_i - x_j||_2^2 = (x_i(1) - x_j(1))^2 + (x_i(2) - x_j(2))^2$$

where x(i) for $x \in \mathbb{R}^2$ and $i \in \{1, 2\}$ is the i-th coefficient of the vector x. For two histograms p and q, we will also be using the L^2 loss function:

$$\mathcal{L}(p, q) = \frac{1}{2}||p - q||_2^2$$

Finally, we set the regularization parameter γ at 8 for most applications. While the implementation of the 2 algorithms is mostly straight-forward, there is one point that needs a bit of attention: how to apply the kernel K to vectors in the algorithm. As a reminder:

$$K = e^{-\frac{C}{\gamma}}$$

where the exponential is applied entry-wise.
This matrix is very large: for N_grid the size of the 2D grid we would like to solve an OT problem on

(here, N_grid = 100), our histograms would have a size of N_grid², which then means that the matrix K has a size of N_grid²xN_grid², or N_grid^4 coefficients. We solve this problem by implementing a separate function *applyK*(v, N_grid, Gamma) which applies the matrix K with a regularization parameter $\gamma = Gamma$ and to a vector v in a more efficient manner thanks to the special structure of K. Since we chose a L² cost for the grid, C can be written as a block matrix of N_grid² matrices $B_{I,J}$ of size N_gridxN_grid such that:

$$B_{I,J} = B^0 + (I - J)^2 * \mathbf{1}$$

where $I, J \in \{1, 2, ..., N\_grid\}$, $\mathbf{1}$ is the matrix of size N_gridxN_grid such that each coefficient is equal to 1, and B_0 is a Toeplitz matrix where:

$$\forall I, J \in \{1, 2, ..., N\_grid\}, B^0_{i,1} = (i-1)^2; B^0_{1,j} = (j-1)^2$$

Thus, the function *applyK*(v, N_grid, Gamma) makes computations per block so the matrices K and C never have to be formed explicitly. We also tried to use the *matmul_toeplitz* method from scipy for this method by constructing N_grid blocks that were all Toeplitz matrixes, but we had problems with it. With this function implemented, we can replicate the Wasserstein symplex shown above. Most functions which are used to analyze and process pictures in this application and the one after were obtained online or from ChatGPT in order to not take too much time on these.

3.3 Results

We can now make our own Wasserstein symplex:

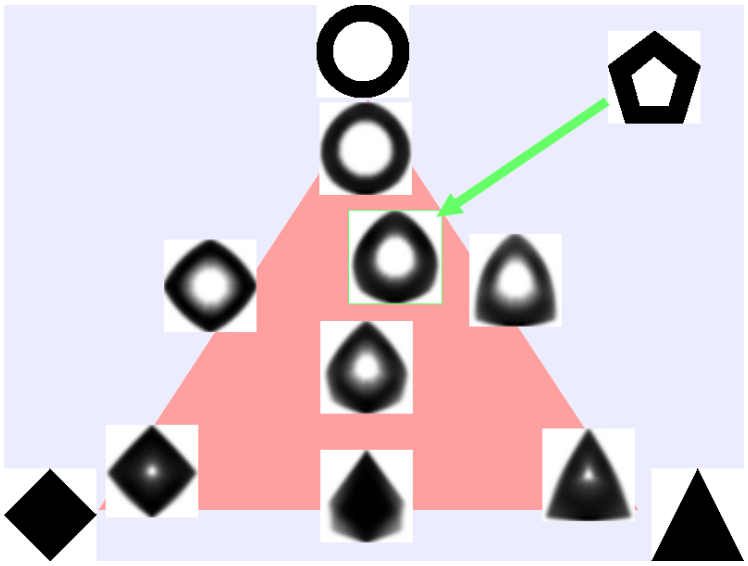

Figure 3: Wasserstein Symplex of our three shapes, and projection of a pentagon on the symplex with L² loss

The optimal λ for the projection of the pentagon was (0.57390262, 0.23109489, 0.19500249), with the first coefficient corresponding to the circle, the second to the triangle and the third to the square.

4 Application on real data

4.1 Objective

Now that we have tested our algorithms on synthetic data, we can now try to apply our results for image color palette manipulation.
Since we do not need lots of data, we handpicked pictures on Google Images. We have three samples

Figure 4: Target picture for the Alps

Figure 5: First sample picture for the Alps

Figure 6: Second sample picture for the Alps

Figure 7: Third sample picture for the Alps

of pictures of landscapes: one from the Alps, one from the Grand Canyon and one from the Hoggar Mountains. In each sample, we have:

- Three somewhat recent pictures of the landscape. They will be referred to as the sample pictures.
- One older picture of the landscape, which is slightly modified to only have a size of around 200x100 pixels. It will be referred to as the target picture.

4.2 Implementation

In order to alter the color palette of a picture, we use the method detailed in the article *Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains* by Solomon et al. (2015).
We first convert our sample pictures and target picture in chrominance-luminance histograms in the format YCbCr. Since the parameter Y only affects the luminosity of each pixel, we do not modify it. We consider CbCr histograms of 64 bins for Cb and Cr, giving us histograms of size $64^2 = 4096$. We again consider a L^2 loss function and L^2 ground cost in the 64x64 CbCr grid. The CbCr histograms of the sample pictures will be referred to as p_s for $s \in \{1, 2, 3\}$ and the CbCr histogram of the target picture as q. We then compute each optimal barycenter $P(\lambda^*)$ of p_s for q.
To modify the target picture, we first compute the optimal transport map T between q and $P(\lambda^*)$. Then, the color of each pixel in the target picture is mapped to the average of the 4 colors which have the biggest weights for receiving mass from this color in the transport map T.

4.3 Results

Unfortunately, we were not able to obtain good results in this part. The fact that the batches of colors in the pictures align well with the distribution of color in the original picture leads us to believe that the optimal transport worked fine. We found that the YCbCr triplets we obtained after processing seemed coherent, but we struggled to convert triplets of this format to RGB triplets. We would have liked to re-do all the code of this part but using a 3D grid with RGB triplets instead of a 2D grid with CbCr couples, to not have to deal with conversions between YCbCr, but we ran out of time.

Figure 8: Target picture for the Alps

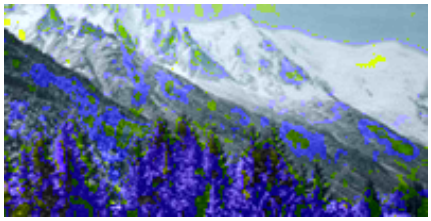

Figure 9: Modified picture of the Alps

Figure 10: Target picture for the Grand Canyon

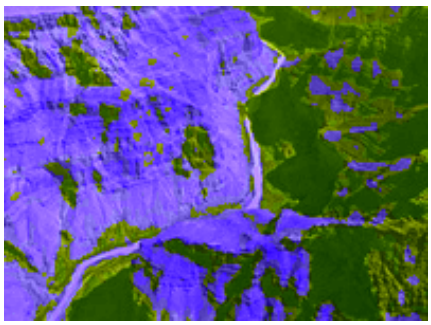

Figure 11: Modified picture of the Granc Canyon

Figure 12: Target picture for the Hoggar Mountains

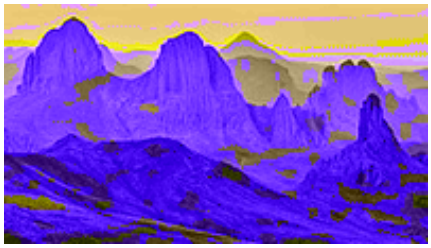

Figure 13: Modified picture of the Hoggar Mountains

5 Code

The Jupyter Notebook written for this project can be found on the Github repository of this project: https://github.com/mkhelifa2002/ProjetML/

6 Conclusion

We have illustrated in this project how Wasserstein barycenter are a powerful tool for image processing. While our results on real-world datasets are flawed, we have implemented a promising algorithm and successfully applied it to a synthetic dataset.
By building on the principles of optimal transport, Wasserstein Barycenters allow for the preservation of geometric relationships within data, making them especially effective in applications where traditional methods may fail to capture the underlying structure. This project shows the versatility of this approach and its potential across various domains.

7 References

AGUEH, M., AND CARLIER, G. 2011. Barycenters in the Wasserstein space. SIAM J. on *Mathematical Analysis* 43, 2

BENAMOU, J.-D., CARLIER, G., CUTURI, M., NENNA, M., AND PEYRE´, G. 2015. Iterative Bregman projections for regularized transportation problems. *SIAM J. on Sci. Computing* 2, 37

BONEEL, PEYRE, CUTURI, Wasserstein Barycentric Coordinates: Histogram Regression Using Optimal Transport, *Siggraph* 2016.

CUTURI, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Adv. in Neural Information Proc. Sys*

SCHMIDT, M. W., VAN DEN BERG, E., FRIEDLANDER, M. P., AND MURPHY, K. P. 2009. Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. In *International Conference on Artificial Intelligence and Statistics* (AISTATS), vol. 5

SOLOMON, J., DE GOES, F., PEYRE´, G., CUTURI, M., BUTSCHER, A., NGUYEN, A., DU, T., AND GUIBAS, L. 2015. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.* (SIGGRAPH) 34, 4