Computational Optimization and Applications
https://doi.org/10.1007/s10589-023-00458-3

# Simple approximative algorithms for free-support Wasserstein barycenters

Johannes von Lindheim[1]

Received: 9 October 2022 / Accepted: 19 January 2023

## Abstract

Computing Wasserstein barycenters of discrete measures has recently attracted considerable attention due to its wide variety of applications in data science. In general, this problem is NP-hard, calling for practical approximative algorithms. In this paper, we analyze a well-known simple framework for approximating Wasserstein-$p$ barycenters, where we mainly consider the most common case $p = 2$ and $p = 1$, which is not as well discussed. The framework produces sparse support solutions and shows good numerical results in the free-support setting. Depending on the desired level of accuracy, this requires only $N - 1$ or $N(N - 1)/2$ standard two-marginal optimal transport (OT) computations between the $N$ input measures, respectively, which is fast, memory-efficient and easy to implement using any OT solver as a black box. What is more, these methods yield a relative error of at most $N$ and $2$, respectively, for both $p = 1, 2$. We show that these bounds are practically sharp. In light of the hardness of the problem, it is not surprising that such guarantees cannot be close to optimality in general. Nevertheless, these error bounds usually turn out to be drastically lower for a given particular problem in practice and can be evaluated with almost no computational overhead, in particular without knowledge of the optimal solution. In our numerical experiments, this guaranteed errors of at most a few percent.

**Keywords** Optimal transport · Discrete Wasserstein barycenter · Approximative algorithm · Error analysis · Sparsity

**Mathematics Subject Classification** 65D18 · 68U10 · 90B80

✉ Johannes von Lindheim
vonlindheim@tu-berlin.de

[1] Institute of Mathematics, Technische Universität Berlin, Strasse des 17. Juni 136, 10587 Berlin, Germany

Published online: 01 March 2023

## 1 Introduction

Wasserstein barycenters are an increasingly popular application of optimal transport in data science [1, 2]. They have nice mathematical properties, since they are the Fréchet means with respect to the Wasserstein distance [3–5]. Their applications range from mixing textures [6, 7], stippling patterns and bidirectional reflectance distribution functions [8], or color distributions and shapes [9] over averaging of sensor data [10] to Bayesian statistics [11], just to name a few. For further reading, we refer to the surveys [12, 13].

Unfortunately, Wasserstein barycenters are in general hard to compute [14]. Many algorithms restrict the support of the solution to a fixed set and minimize only over the weights. Such methods include projected subgradient [15], iterative Bregman projections [16], (proximal) algorithms based on the latter [17], interior point methods [18], Gauss-Seidel based alternating direction of multipliers [19], multi-marginal Sinkhorn algorithms and its accelerated variants [20], debiased Sinkhorn barycenter algorithms [21], methods using the Wasserstein distance on a tree [22], accelerated Bregman projections [23] and methods based on mirror proximal maps or on a dual extrapolation scheme [24], among others. While iterative Bregman projections are a standard benchmark that are hard to beat in terms of simplicity and speed, fixed-support methods applied on a grid suffer from the curse of dimensionality.

On the other hand, barycenters without such restriction are called free-support barycenters. This approach can overcome the curse of dimensionality, since the optimal solution is sparse. Free-support barycenters can be computed directly from the solution of the closely related multi-marginal optimal transport (MOT) problem. The latter was originally introduced in [25] in the continuous setting for squared Euclidean costs and further generalized in various ways, e.g., to entropy regularized [26, 27] and unbalanced variants with non-exact marginal constraints [28]. The solution to MOT can be obtained by solving a linear program (LP) that unfortunately scales exponentially in $N$, however [29]. Although there are exact polynomial-time methods for measures on $\mathbb{R}^d$ for fixed $d$ [30], see also LP-based methods in [29, 31, 32], these are not necessarily fast in practice and rather involved to implement. A remedy is to resort to approximative approaches, which include so far a Newton-approach that iteratively alternates between optimizing over the weights and supports [15], another LP-based method [33], an inexact proximal alternating minimization method [34], an iterative stochastic algorithm [35] and the iterative swapping algorithm [2]. A free-support barycenter method based on the Frank–Wolfe algorithm is given in [36]. Another method in [37] computes continuous barycenters using another way of parameterizing them. For approaches for MOT similar to this paper, see [38]. Further speedups can be obtained by subsampling the given measures [39] or dimensionality reduction of the support point clouds [40].

Despite the plethora of literature, many algorithms with low theoretical computational complexity or high accuracy solutions are rather involved to implement. This impedes its actual usage and further research in practice. To the best of our knowledge, there does not exist an algorithm that fulfills the following list of desiderata in the free-support setting:

- simple to implement,
- sharp theoretical error bounds,
- sparse solutions, and
- good numerical results in practice.

The purpose of this paper is to show that all of these points can be achieved using one iteration of a simple well-known fixed-point algorithm, which only requires some off-the-shelve two-marginal OT solver as ingredient to its otherwise easy implementation. Here we consider the cases $p = 2$ and $p = 1$, where the latter has received less attention in the literature so far. One such fixed-point iteration consists in computing optimal transport plans from a given measure to the input measures and pushing each atom to the $p$-barycenter of its target locations. For the cost of $N - 1$ OT plans, this yields a relative error bound of $N$, or a 2-approximation, respectively, when averaging over these results, which requires to solve $N(N-1)/2$ OT problems. The key to these theoretical bounds is based on the observation that the they are already fulfilled for the input measures or their mixture, respectively, which we choose as initialization. On the other hand, we show that the aforementioned fixed-point iteration guarantees to at least retain the current approximation quality, but improves it considerably in practice in the first step.

Note that other algorithms with an upper error bound of 2 have been proposed in [33] for $p = 2$. The basic algorithm produces a barycenter with support $\cup_{i=1}^{N} \operatorname{supp}(\mu^i)$ by solving an LP over its weights. However, while this support choice leads to bad approximations in practice (consider, e.g., two distinct Dirac measures as input), for a merely theoretical 2-approximation, no computation is necessary as mentioned above. On the other hand, the implementation and proofs of the other algorithms in that paper with better results in practice are rather involved.

In view of the hardness of the Wasserstein barycenter problem [14], it is clear that the derived relative error bounds cannot be close to 1 for every set of inputs, unless P = NP. However, the improvement made by one iteration is straightforward to evaluate in the proposed algorithms, such that it can output relative error bounds specific to the given problem without knowing the optimal solution. We observe these resulting improved bounds to be close to 1 in the numerical experiments.

This paper is organized as follows: We introduce the Wasserstein barycenter problem and our notation in Sect. 2. In Sect. 3, we state the algorithms considered in this paper. In Sect. 4, we analyze their worst-case relative error. In Sect. 5, we provide a comparison with other algorithms on a synthetic data set, a numerical exploration of Wasserstein-1 barycenters, and two applications of the discussed framework. Concluding remarks are given in Sect. 6.

## 2 Wasserstein barycenter problem

In the following, we denote by $\|\cdot\|$ the Euclidean norm on $\mathbb{R}^d$ and by $\mathcal{P}(\mathbb{R}^d)$ the space of probability measures on $\mathbb{R}^d$. Let $1 \le p < \infty$. For two discrete measures

$$\mu^1 = \sum_{k=1}^{n_1} \mu_k^1 \delta(x_k^1), \quad \mu^2 = \sum_{l=1}^{n_2} \mu_l^2 \delta(x_l^2),$$

the *Wasserstein-$p$ distance* is defined by

$$\mathcal{W}_p^p(\mu^1, \mu^2) = \min_{\pi \in \Pi(\mu^1, \mu^2)} \langle c_p, \pi \rangle,$$

where $\langle c_p, \pi \rangle = \int_{\mathbb{R}^d \times \mathbb{R}^d} c_p \, \mathrm{d}\pi$ with $c_p(x, y) := \|x - y\|^p$ and $\Pi(\mu^1, \mu^2)$ denotes the set of probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals $\mu^1$ and $\mu^2$. The above optimization problem is convex, but can have multiple minimizers $\pi$.

In this paper, we are given $N$ discrete probability measures $\mu^i \in \mathcal{P}(\mathbb{R}^d)$ supported at $\operatorname{supp}(\mu^i) = \{x_1^i, \dots, x_{n_i}^i\}$, where the $x_l^i$ are pairwise different for every $i$, i.e.,

$$\mu^i = \sum_{l=1}^{n_i} \mu_l^i \delta(x_l^i), \quad i = 1, \dots, N.$$

Let $\Delta_N := \{\lambda \in (0,1)^N : \sum_{i=1}^N \lambda_i = 1\}$ denote the open probability simplex. For given weights $\lambda = (\lambda_1, \dots, \lambda_N) \in \Delta_N$, we are interested in the computation of Wasserstein barycenters, which are the solutions to the optimization problem

$$\min_{\nu \in \mathcal{P}(\mathbb{R}^d)} \Psi_p(\nu), \qquad \Psi_p(\nu) := \sum_{i=1}^N \lambda_i \mathcal{W}_p^p(\nu, \mu^i). \tag{2.1}$$

The following theorem restates important results from [41, Prop. 3], which connects barycenter problems with what is nowadays known as multi-marginal optimal transport, as well as [29, Prop. 1, Thm. 2] and [42, Thm. 1] in our notation.

**Theorem 2.1** *The barycenter problem (2.1) has at least one optimal solution $\hat{\nu}$. Every optimal solution $\hat{\nu}$ fulfills*

$$\operatorname{supp}(\hat{\nu}) \subseteq \left\{ \sum_{i=1}^N \lambda_i x^i : x^i \in \operatorname{supp}(\mu^i),\ i = 1, \dots, N \right\}. \tag{2.2}$$

*Moreover, there exists an optimal solution $\hat{\nu}$, such that*

$$\#\operatorname{supp}(\hat{\nu}) \le \sum_{i=1}^N n_i - N + 1. \tag{2.3}$$

***Proof*** Note that (2.2) is straightforward to obtain from the relation to multi-marginal optimal transport [41, Prop. 3]. In the special case $p = 2$, the results from [29], in particular (2.3), can readily be generalized to arbitrary $\lambda \in \Delta_N$. For general $p \geq 1$ and barycenter problems with even more general cost functions, this follows from sparsity of multi-marginal optimal transport recently shown in [42, Thm. 1] in combination with [41, Prop. 3]. □

In particular, the theorem says that finding optimal Wasserstein barycenters is a discrete optimization problem over the weights of its finite support, which is contained in the convex hull of the supports of the $\mu^i$. However, the number of possible support points scales exponentially in $N$.

## 3 Algorithms for barycenter approximation

In this section, after motivating the main framework considered in this paper in Sect. 3.1, we discuss two more concrete configurations of it in Sects. 3.2 and 3.3.

### 3.1 Motivation

In its core, the algorithms in this paper approximate barycenters by "averaging optimal transport plans" from a particular reference measure to the input measures in some sense. This approach is well-known and comes in various flavors in the literature. For example, it can be viewed through the lens of generalized geodesics in Wasserstein spaces [43] and recent literature on linear optimal transport and relatives [44–48]. On the other hand, in [15], one of the first papers on the numerical approximation of Wasserstein barycenters, the idea is presented as a Newton iteration. The same iteration is analyzed in the continuous setting in [49], and it can be used as a characterization of Wasserstein barycenters in terms of fixed points of this procedure, even for uncountably many input measures [50]. See also [51] for this algorithm in the context of weak optimal transport.

Let us define the averaging of transport plans more precisely.

**Definition 3.1** Given a discrete measure $\nu = \sum_{k=1}^{n_\nu} \nu_k \delta(y_k) \in \mathcal{P}(\mathbb{R}^d)$ and transport plans

$$\pi^i := \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \delta(y_k, x^i_l) \in \Pi(\nu, \mu^i), \quad i = 1, \dots, N,$$

set $\pi = (\pi^1, \dots, \pi^N)$ and let for $k = 1, \dots, n_\nu$, $p \geq 1$, the barycentric map $M^p_{\lambda,\pi} : \operatorname{supp}(\nu) \to \mathbb{R}^d$ be defined as

$$m_k = M^p_{\lambda,\pi}(y_k) := \operatorname{argmin}_{m \in \mathbb{R}^d} \sum_{i=1}^{N} \lambda_i \sum_{l=1}^{n_i} \frac{\pi^i_{k,l}}{\nu_k} \|m - x^i_l\|^p.$$

Furthermore, we define the mapping

$$G^p_{\lambda,\pi}(\nu) := \sum_{k=1}^{n_\nu} \nu_k \delta(m_k). \tag{3.1}$$

That is, each atom $y_k$ in the measure $\nu$ is pushed to the weighted barycenter $m_k$ of its target locations $x^i_l$, where the weights are given by the $\lambda_i$ and the weights of the source locations as given by the transport plans $\pi^i$, relative to the corresponding transported mass $\nu_k$.

Note that for $p = 2$, the map $M^p_{\lambda,\pi}$ is the classical mean, whereas for $p = 1$, it is called geometric median. It is uniquely defined, whenever the points are not collinear, see [52]. Otherwise, in case of ambiguity, the set of minimizers is a one-dimensional line segment, of which we choose the midpoint. However, unlike in the case $p = 2$, there is no explicit formula or exact algorithm involving only arithmetic operations and $k$-th roots to compute $M^p_{\lambda,\pi}$, see [53]. Nevertheless, the geometric median can be approximated using Weiszfeld's algorithm, which consists mainly in the fixed point iteration

$$m^{(k+1)} = \Big( \sum_{i=1}^{N} \frac{\lambda_i}{\|x_i - m^{(k)}\|} \Big)^{-1} \Big( \sum_{i=1}^{N} \frac{\lambda_i x_i}{\|x_i - m^{(k)}\|} \Big),$$

with a particular choice of the starting point $m^{(0)}$ that guarantees $m^{(k)} \neq x_i$ for all $i = 1, \dots, N$ and $k \geq 0$. This method is a gradient descent method and accelerated methods are also available. For more details, we refer to the survey [52].

Next, we comment on the relation of $G^p_{\lambda,\pi}$ to Wasserstein barycenters. In the most important case $p = 2$, formula (3.1) simplifies when the transport plans are non-mass-splitting, that is, for every $i = 1, \dots, N$, each $\pi^i$ is supported on the graph of some transport maps $T^i : \mathrm{supp}(\nu) \to \mathrm{supp}(\mu^i)$ with $T^i_\# \nu = \mu^i$. In that case, $G^p_{\lambda,\pi}$ pushes $\nu$ forward by the average of the transport maps,

$$G^p_{\lambda,\pi} = \Big( \sum_{i=1}^{N} \lambda_i T^i \Big)_\#.$$

This is called McCann interpolation for $N = 2$. In the nondiscrete setting, if $\nu$ is absolutely continuous, then optimal transport maps $T^i$ exist by Brenier's theorem, see e.g. [54, Thm. 1.22]. In fact, [49] discusses the following fixed-point iteration for approximate barycenter computation:

1. Compute the optimal transport maps $T^i$ from $\nu$ to $\mu^i$, $i = 1, \dots, N$
2. Update $\nu \leftarrow \Big( \sum_{i=1}^{N} \lambda_i T^i \Big)_\# \nu$, repeat.

It is shown that if there is a unique fixed point, then this is the optimal barycenter and the iteration converges, which is the case for, e.g., Gaussian measures. The convergence is numerically observed to be very fast, and in certain special cases, it is reached already in one iteration. Taking the geometric structure of the Wasserstein space into

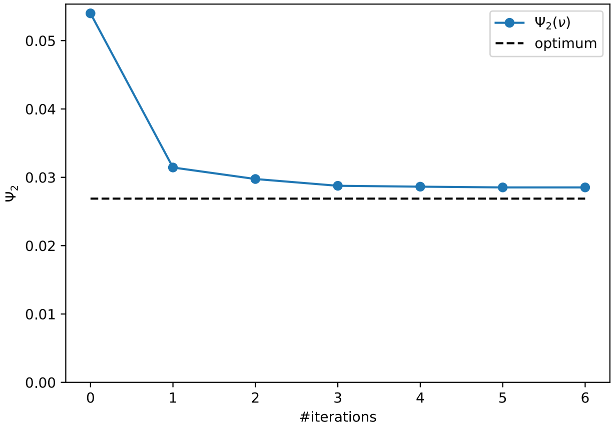


**Fig. 1** Barycenter cost $\Psi_2(\nu^{(k)})$ over the number of iterations $k$ in blue. The black dashed line depicts the optimal cost $\Psi_2(\hat{\nu})$

account, see, e.g., [43], the fixed-point procedure above is the the typical algorithm for computing Fréchet means on manifolds [3–5].

This motivates the algorithms presented in this paper, which consist in deliberately performing only the first iteration of the fixed-point procedure above. More precisely, the approximate barycenters are of the form $\tilde{\nu} = G^p_{\lambda,\pi}(\nu)$ for certain plans $\pi^i$ and initial measures $\nu$. We found that this yields the best tradeoff between speed and accuracy in practice, since the error improvement of further iterations is typically rather small.

We illustrate this claim by the following numerical example. We create $N = 10$ discrete measures $\mu^i = \sum_{l=1}^n \frac{1}{n}\delta(x^i_l)$, $i = 1, \dots, N$, with $n = 50$ points each, which we sample uniformly from the unit disk and center to have mean zero. We initialize with $\nu^{(0)}:=\mu^1$ and perform the iteration above until convergence after 5 iterations, that is, $\nu^{(6)} = \nu^{(5)}$. Optimal transport maps $T^i$ always exist here, since we have empirical measures with the same number of atoms. In Fig. 1, we show the cost $\Psi_2(\nu^{(k)})$ with respect to $k$ and compare to the cost $\Psi_2(\hat{\nu})$ of an optimal barycenter $\hat{\nu}$, that is, a solution of (2.1). While the error $\Psi_2(\nu^{(k)}) - \Psi_2(\hat{\nu})$ is decreased in the first step by 83.2%, the improvement in the second iteration is only 37% of the remaining error and decreases even further until convergence to a suboptimal solution. Moreover, the absolute cost decrease $\Psi_2(\nu^{(2)}) - \Psi_2(\nu^{(1)})$ in the second iteration is only 7.5% of the decrease $\Psi_2(\nu^{(1)}) - \Psi_2(\nu^{(0)})$ of the first iteration. This also makes sense intuitively, since it seems reasonable that the largest improvement is gained by pushing every support point from some rather arbitrary initialization to the barycenter of several other reasonably chosen support points of the $\mu^i$.

Furthermore, from a theoretical standpoint, there are simple examples with convergence after one iteration for both presented algorithms below, such that we cannot expect in general to gain any improvements using more than one iteration either. In particular, as in the numerical example above, there is no way to guarantee convergence to the optimum of this iterative procedure in general, which is the case for any algorithm due to the NP-hardness of the problem [14].

### 3.2 Reference algorithm

In this section, we choose $\nu = \mu^j$ as initialization. For simplicity of notation, reorder the measures such that $j = 1$. That is, we compute $N-1$ optimal transport plans

$$\pi^i = \sum_{k=1}^{n_1}\sum_{l=1}^{n_i} \pi^i_{k,l}\delta(x^1_k, x^i_l) \in \operatorname{argmin}_{\pi\in\Pi(\mu^1,\mu^i)}\langle c_p, \pi\rangle, \quad i = 2,\dots,N$$

and consider the approximate barycenter defined by

$$\tilde{\nu} = \sum_{k=1}^{n_1} \mu^1_k \delta(M^p_{\lambda,\pi}(x^1_k)). \tag{3.2}$$

Note that the support of $\tilde{\nu}$ given by (3.2) is very sparse, since it contains only $n_1$ elements, which is an interesting feature from a computational point of view.

For $p = 2$, if the input measures are given in terms of matrices $X^i \in \mathbb{R}^{n_i\times d}$, where the rows are the support points, and the corresponding mass weights are the vectors $\mu^i \in \mathbb{R}^{n_i}$ for all $i = 1,\dots,N$, then computing the support matrix $Y \in \mathbb{R}^{n_1\times d}$ of (3.2) can be written as an average of $N$ matrix products as outlined in Algorithm 1.

**Algorithm 1** Reference algorithm, $p = 2$

```
**Input:** Support points $X^i \in \mathbb{R}^{n_i\times d}$, masses $\mu^i \in \mathbb{R}^{n_i}$, $i = 1,\dots,N$, weights $0 < \lambda \in \mathbb{R}^N$
$\pi^1 := \operatorname{diag}(\mu^1)$
**for** $i = 2,\dots,N$ **do**
    Compute $\pi^i \in \operatorname{argmin}_{\pi\in\Pi(\mu^1,\mu^i)}\langle c, \pi\rangle \in \mathbb{R}^{n_1\times n_i}$
**end for**
$Y := \operatorname{diag}(\mu^1)^{-1}\sum_{i=1}^N \lambda_i \pi^i \cdot X^i$
**Output:** support $Y \in \mathbb{R}^{n_1\times d}$, masses $\mu^1 \in \mathbb{R}^{n_1}$
```

In the case $p = 1$, since there is no closed form for $M^p_{\lambda,\pi}(x^1_k)$, we have to make slight modifications to the algorithm in that case.

***Remark 3.2*** Let $f_{x,\lambda}(m) := \sum_{i=1}^N \lambda_i\|x_i - m\|$ and $\hat{m} = \operatorname{argmin}_{m\in\mathbb{R}^d} f(m)$. We show that Weiszfeld's algorithm is guaranteed to approximate $f(\hat{m})$ up to a multiplicative factor of $(1+\varepsilon)$ for a certain minimal number of iterations that is explicitly computable. In [52, Thm. 8.2] it is shown for the Weiszfeld iterates $m^{(k)}$ that

$$f(m^{(k)}) - f(\hat{m}) \le \frac{M}{k}\|m^{(0)} - \hat{m}\|^2,$$

with an explicit formula for $M$, depending only on the $x_i$ and $m^{(0)}$. Since $f$ is convex, from $\nabla f = 0$, a simple calculation shows that $\hat{m}$ must lie in the convex hull of the $x_i$. Thus

$$\|m^{(0)} - \hat{m}\|^2 \leq \max_{i=1,\dots,N} \|m^{(0)} - x_i\|^2.$$

Moreover, by (4.3), it holds

$$\sum_{i<j}^{N} \lambda_i \lambda_j \|x_i - x_j\| \leq \sum_{i=1}^{N} \lambda_i \|x_i - \hat{m}\| = f(\hat{m}),$$

such that for any given $\varepsilon > 0$, choosing

$$k \geq \frac{M \cdot \max_{i=1,\dots,N} \|m^{(0)} - x_i\|^2}{\varepsilon \sum_{i<j}^{N} \lambda_i \lambda_j \|x_i - x_j\|}$$

guarantees that

$$\begin{aligned} f(m^{(k)}) &= (f(m^{(k)}) - f(\hat{m})) + f(\hat{m}) \leq \frac{M}{k} \|m^{(0)} - \hat{m}\|^2 + f(\hat{m}) \\ &\leq \frac{M\varepsilon \sum_{i<j}^{N} \lambda_i \lambda_j \|x_i - x_j\|}{M \cdot \max_{i=1,\dots,N} \|m^{(0)} - x_i\|^2} \|m^{(0)} - \hat{m}\|^2 + f(\hat{m}) \leq (1+\varepsilon) f(\hat{m}). \end{aligned}$$

This prepares us to state the reference algorithm for the case $p = 1$, see Algorithm 2.

**Algorithm 2** Reference algorithm, $p = 1$

```
Input: Measures $\mu^i = \sum_{l=1}^{n_i} \mu_l^i \delta(x_l^i)$, $i = 1, \dots, N$, weights $\lambda \in \Delta_N$, Weiszfeld accuracy $\varepsilon$
$\pi^1 := \mathrm{diag}(\mu^1)$
for $i = 2, \dots, N$ do
    Compute $\pi^i \in \mathrm{argmin}_{\pi \in \Pi(\mu^1, \mu^i)} \langle c, \pi \rangle$
end for
for $k = 1, \dots, n_1$ do
    $m_k := \mathrm{argmin}_{m \in \mathbb{R}^d} \sum_{i=1}^{N} \lambda_i \sum_{l=1}^{n_i} \frac{\pi_{k,l}^i}{\mu_k^1} \|m - x_l^i\|$ up to factor $(1+\varepsilon)$
end for
Output: approximate barycenter $\tilde{\nu} := \sum_{k=1}^{n_1} \mu_k^1 \delta(m_k)$
```

### 3.3 Pairwise algorithm

We will see that in order to achieve better results than the reference algorithm, it is beneficial to "average out" the asymmetry introduced by choosing $\mu^1$ as the reference measure in (3.2). Therefore, we choose

$$\nu = \sum_{i=1}^{N} \lambda_i \mu^i \tag{3.3}$$

as initial measure in this section. However, instead of computing optimal plans from $\nu$ to each $\mu^i$, we solve

$$\pi^{ij} \in \operatorname{argmin}_{\pi \in \Pi(\mu^i, \mu^j)} \langle c, \pi \rangle$$

pairwise for every $1 \leq i < j \leq N$ and use the transport plans

$$\pi^i = \sum_{j=1}^{N} \lambda_j \pi^{ji} \in \Pi(\nu, \mu^i), \tag{3.4}$$

in (3.1), so that our approximate barycenter $\tilde{\nu}$ with (3.3) and (3.4) reads as

$$\tilde{\nu} = G^p_{\lambda,\pi}(\nu) = \sum_{i=1}^{N} \lambda_i \sum_{k=1}^{n_i} \mu^i_k \delta(M_{\lambda,(\pi^{i1},\ldots,\pi^{iN})}(x^i_k)). \tag{3.5}$$

Splitting up the OT computations like this scales better in terms of computational complexity and seems to yield better numerical results in practice. Clearly, we will have

$$\#\mathrm{supp}(\tilde{\nu}) \leq n_1 + \cdots + n_N,$$

that is, $\tilde{\nu}$ meets practically the same sparsity bound as an optimal solution $\hat{\nu}$, see (2.3).

***Remark 3.3*** Note that the inner sum in (3.5) is of the form (3.2). If we denote by $\tilde{\nu}^i$ the barycenter obtained from the reference algorithm, when $\mu^i$ was the reference measure, i.e., permuted to the first position, our approximation (3.5) is simply

$$\tilde{\nu} = \sum_{i=1}^{N} \lambda_i \tilde{\nu}^i.$$

However, since we can choose $\pi^{ji} = (\pi^{ij})^{\mathrm{T}}$, we save half of the necessary OT computations compared to executing the reference algorithm $N$ times.

Algorithm 3 summarizes this approach for $p = 2$ using matrix-vector notation, where $\odot$ denotes element-wise multiplication and $\mathbb{1}_d$ denotes a $d$-dimensional vector of ones. Note that $\eta$ denotes an upper bound of the relative error for the particular given problem, i.e., it holds that $\Psi_2(\tilde{\nu})/\Psi_2(\hat{\nu}) \leq \eta$. This is proven in Sect. 4.

Again, these matrix-vector computations will not work in the case $p = 1$. Instead, Algorithm 4 outlines the computation of (3.5) using Weiszfeld's algorithm.

**Algorithm 3** Pairwise algorithm, $p = 2$

```
Input: Support points $X^i \in \mathbb{R}^{n_i \times d}$, masses $\mu^i \in \mathbb{R}^{n_i}$, $i = 1, \dots, N$, weights $0 < \lambda \in \mathbb{R}^N$
Set $M = n_1 + \cdots + n_N$
for $i = 1, \dots, N$ do
    $\pi^{ii} := \mathrm{diag}(\mu^i)$
    for $j = i + 1, \dots, N$ do
        Compute $\pi^{ij} \in \mathrm{argmin}_{\pi \in \Pi(\mu^i, \mu^j)} \langle c, \pi \rangle \in \mathbb{R}^{n_i \times n_j}$
        $\pi^{ji} := (\pi^{ij})^{\mathrm{T}}$
    end for
end for
```

$$P := \begin{bmatrix} \pi^{11} & \dots & \pi^{1N} \\ \vdots & \ddots & \vdots \\ \pi^{N1} & \dots & \pi^{NN} \end{bmatrix} \in \mathbb{R}^{M \times M}$$

$$X := \begin{bmatrix} X^1 \\ \vdots \\ X^N \end{bmatrix} \in \mathbb{R}^{M \times d}$$

$$\Lambda := \Big[\underbrace{\lambda_1, \dots, \lambda_1}_{n_1 \text{ times}}, \dots, \underbrace{\lambda_N, \dots, \lambda_N}_{n_N \text{ times}}\Big]^{\mathrm{T}} \in \mathbb{R}^M$$

$$\mu := \left[\mu^1_1 \dots \mu^1_{n_1} \dots \mu^N_1 \dots \mu^N_{n_N}\right]^{\mathrm{T}} \in \mathbb{R}^M$$

$$Y := \mathrm{diag}(\mu)^{-1} \cdot P \cdot \mathrm{diag}(\Lambda) \cdot X \in \mathbb{R}^{M \times d}$$

$$\nu := \Lambda \odot \mu \in \mathbb{R}^M$$

$$\eta := 2 - \nu^{\mathrm{T}} \cdot ((Y - X) \odot (Y - X)) \cdot \mathbb{1}_d / \sum_{i<j}^{N} \lambda_i \lambda_j \langle c_2, \pi^{ij} \rangle \in \mathbb{R}$$

**Output:** support $Y \in \mathbb{R}^{M \times d}$, masses $\nu \in \mathbb{R}^M$, error bound $\eta \in [1, 2]$

## 4 Analysis

In this section, we give worst case bounds for the relative error $\Psi_p(\tilde{\nu}) / \Psi_p(\hat{\nu})$, where $\tilde{\nu}$ is an approximate barycenter computed by one of the algorithms above, $\hat{\nu}$ is an optimal barycenter, and $\Psi_p$ is the objective defined in (2.1). In the proofs, we will use the following basic identities.

**Algorithm 4** Pairwise algorithm, $p = 1$

```
Input: Measures $\mu^i = \sum_{k=1}^{n_i} \mu^i_k \delta(x^i_k)$, $i = 1, \dots, N$, weights $\lambda \in \Delta_N$, Weiszfeld accuracy $\varepsilon$
for $i = 1, \dots, N$ do
    $\pi^{ii} := \mathrm{diag}(\mu^i)$
    for $j = i + 1, \dots, N$ do
        Compute $\pi^{ij} \in \mathrm{argmin}_{\pi \in \Pi(\mu^i, \mu^j)} \langle c_1, \pi \rangle \in \mathbb{R}^{n_i \times n_j}$
        $\pi^{ji} := (\pi^{ij})^{\mathrm{T}}$
    end for
end for
for $i = 1, \dots, N$ do
    for $k = 1, \dots, n_i$ do
        $m^i_k := \mathrm{argmin}_{m \in \mathbb{R}^d} \sum_{j=1}^N \lambda_j \sum_{l=1}^{n_j} \frac{\pi^{ij}_{k,l}}{\mu^i_k} \|m - x^j_l\|$ up to factor $(1 + \varepsilon)$
    end for
end for
$\eta := \sum_{i=1}^N \lambda_i \sum_{j=1}^N \lambda_j \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^{ij}_{k,l} \|m^i_k - x^j_l\| / \sum_{i<j}^N \lambda_i \lambda_j \langle c_1, \pi^{ij} \rangle$
Output: barycenter $\tilde{\nu} := \sum_{i=1}^N \lambda_i \sum_{k=1}^{n_i} \mu^i_k \delta(m^i_k)$, error bound $\eta \in [1, 2]$
```

**Lemma 4.1** *For any points $x_1, \dots, x_N, y \in \mathbb{R}^d$, $\lambda \in \Delta_N$ and $m := \sum_{i=1}^N \lambda_i x_i$, we have the following identities:*

$$\sum_{i=1}^N \lambda_i \|x_i - y\|^2 = \|m - y\|^2 + \sum_{i=1}^N \lambda_i \|x_i - m\|^2, \tag{4.1}$$

$$\sum_{i=1}^N \lambda_i \|x_i - m\|^2 = \sum_{i<j}^N \lambda_i \lambda_j \|x_i - x_j\|^2, \tag{4.2}$$

$$\sum_{i=1}^N \lambda_i \|x_i - y\| \geq \sum_{i<j}^N \lambda_i \lambda_j \|x_i - x_j\|. \tag{4.3}$$

***Proof*** For (4.1), we set $z := m - y$ to obtain

$$\begin{aligned}\sum_{i=1}^N \lambda_i \|x_i - y\|^2 &= \sum_{i=1}^N \lambda_i \|x_i - m + z\|_2^2 = \sum_{i=1}^N \lambda_i \left( \|z\|^2 + \|x_i - m\|^2 - 2\langle x_i - m, z\rangle \right) \\ &= \|m - y\|^2 + \sum_{i=1}^N \lambda_i \|x_i - m\|^2.\end{aligned}$$

For (4.2), plugging $y = x_j$ into (4.1), we get

$$\sum_{i=1}^{N} \lambda_i \|x_i - x_j\|^2 = \|m - x_j\|^2 + \sum_{i=1}^{N} \lambda_i \|x_i - m\|^2.$$

Weighting this equality with $\lambda_j$ and summing over $j = 1, \ldots, N$, we get

$$\begin{aligned}\sum_{i,j=1}^{N} \lambda_i \lambda_j \|x_i - x_j\|^2 &= \sum_{j=1}^{N} \lambda_j \|x_j - m\|^2 + \sum_{j=1}^{N} \lambda_j \sum_{i=1}^{N} \lambda_i \|x_i - m\|^2 \\ &= 2 \sum_{i=1}^{N} \lambda_i \|x_i - m\|^2.\end{aligned}$$

Dividing by 2 yields (4.2). For (4.3), note that by the triangle inequality,

$$\begin{aligned}\sum_{i<j}^{N} \lambda_i \lambda_j \|x_i - x_j\| &= \frac{1}{2} \sum_{i=1}^{N} \lambda_i \sum_{j=1}^{N} \lambda_j \|x_i - x_j\| \\ &\leq \frac{1}{2} \sum_{i=1}^{N} \lambda_i \sum_{j=1}^{N} \lambda_j (\|x_i - y\| + \|y - x_j\|) \\ &= \frac{1}{2} \Big( \sum_{i=1}^{N} \lambda_i \|x_i - y\| + \sum_{j=1}^{N} \lambda_j \|y - x_j\| \Big) = \sum_{i=1}^{N} \lambda_i \|x_i - y\|.\end{aligned}$$

□

In order to upper bound $\Psi_p(\tilde{\nu})/\Psi_p(\hat{\nu})$, we require a lower bound on $\Psi_p(\hat{\nu})$.

**Proposition 4.2** *For any discrete $\nu \in \mathcal{P}(\mathbb{R}^d)$ and $p = 1, 2$, it holds that*

$$\Psi_p(\nu) \geq \sum_{i<j}^{N} \lambda_i \lambda_j \mathcal{W}_p^p(\mu^i, \mu^j). \tag{4.4}$$

***Proof*** Let $p \in \{1, 2\}$ and $\nu = \sum_{k=1}^{n_\nu} \nu_k \delta(y_k)$ be arbitrary. Take $\pi^i \in \operatorname{argmin}_{\pi \in \Pi(\nu, \mu^i)} \langle c_p, \pi \rangle$, then by definition,

$$\Psi_p(\nu) = \sum_{i=1}^{N} \lambda_i \mathcal{W}_p^p(\nu, \mu^i) = \sum_{i=1}^{N} \lambda_i \sum_{k=1}^{n_\nu} \sum_{l_i=1}^{n_i} \pi_{k,l_i}^i \|y_k - x_{l_i}^i\|^p.$$

Since it holds for any $i = 1, \ldots, N$ and $k = 1, \ldots, n_\nu$ that

$$\sum_{l_1,\ldots,l_{i-1},l_{i+1},\ldots,l_N} \frac{\pi_{k,l_1}^1 \ldots \pi_{k,l_{i-1}}^{i-1} \pi_{k,l_{i+1}}^{i+1} \ldots \pi_{k,l_N}^N}{\nu_k^{N-1}} = 1,$$

we get

$$\Psi_p(\nu) = \sum_{i=1}^N \lambda_i \sum_{k=1}^{n_\nu} \sum_{l_1,\dots,l_N} \frac{\pi^1_{k,l_1} \dots \pi^N_{k,l_N}}{\nu_k^{N-1}} \|y_k - x^i_{l_i}\|^p$$
$$= \sum_{k=1}^{n_\nu} \sum_{l_1,\dots,l_N} \frac{\pi^1_{k,l_1} \dots \pi^N_{k,l_N}}{\nu_k^{N-1}} \sum_{i=1}^N \lambda_i \|y_k - x^i_{l_i}\|^p$$

and by (4.2) and (4.3), this yields

$$\Psi_p(\nu) \geq \sum_{k=1}^{n_\nu} \sum_{l_1,\dots,l_N} \frac{\pi^1_{k,l_1} \dots \pi^N_{k,l_N}}{\nu_k^{N-1}} \sum_{i<j}^N \lambda_i \lambda_j \|x^i_{l_i} - x^j_{l_j}\|^p$$
$$= \sum_{i<j}^N \lambda_i \lambda_j \sum_{k=1}^{n_\nu} \sum_{l_i,l_j} \frac{\pi^i_{k,l_i} \pi^j_{k,l_j}}{\nu_k} \|x^i_{l_i} - x^j_{l_j}\|^p.$$

It is straightforward to check that

$$\sum_{k=1}^{n_\nu} \sum_{l_i,l_j} \frac{\pi^i_{k,l_i} \pi^j_{k,l_j}}{\nu_k} \delta(x^i_{l_i}, x^j_{l_j}) \in \Pi(\mu^i, \mu^j),$$

and so we get

$$\Psi_p(\nu) \geq \sum_{i<j}^N \lambda_i \lambda_j \mathcal{W}_p^p(\mu^i, \mu^j).$$

□

Equipped with (4.4), we can see that already the simple choices $\nu = \mu^j$ and $\nu = \sum_{i=1}^N \lambda_i \mu^i$ for the initial measure approximate the optimal barycenter to some extent.

**Proposition 4.3** *Let $p \in \{1, 2\}$ and $\hat{\nu}$ be an optimal barycenter in (2.1).*

*(i) For $\nu := \mu^j$, it holds that*

$$\frac{\Psi_p(\nu)}{\Psi_p(\hat{\nu})} \leq \frac{1}{\lambda_j}.$$

*Note that in particular, if $j \in \operatorname{argmax}_{i=1}^N \lambda_i$, then $\Psi_p(\nu)/\Psi_p(\hat{\nu}) \leq N$.*

*(ii) Let $\nu := \sum_{i=1}^N \lambda_i \mu^i$, then*

$$\frac{\Psi_p(\nu)}{\Psi_p(\hat{\nu})} \leq 2.$$

*(iii) If $\nu$ is chosen randomly as one of the $\mu^i$ with probabilities $\lambda_i$, then also*

$$\frac{\mathbb{E}[\Psi_p(\nu)]}{\Psi_p(\hat{\nu})} \leq 2.$$

***Proof*** (i) Let $\nu := \mu^j$, then we see that

$$\Psi_p(\nu) = \Psi_p(\mu^j) = \sum_{i=1}^N \lambda_i \mathcal{W}_p^p(\mu^i, \mu^j).$$

By (4.4),

$$\Psi_p(\hat{\nu}) \geq \sum_{i<j}^N \lambda_i \lambda_j \mathcal{W}_p^p(\mu^i, \mu^j) \geq \lambda_j \sum_{i=1}^N \lambda_i \mathcal{W}_p^p(\mu^i, \mu^j),$$

such that

$$\frac{\Psi_p(\nu)}{\Psi_p(\hat{\nu})} \leq \frac{1}{\lambda_j}.$$

(ii) For the choice $\nu := \sum_{i=1}^N \lambda_i \mu^i$, taking $\pi^{ij} \in \operatorname{argmin}_{\pi \in \Pi(\mu^i, \mu^j)} \langle c_p, \pi \rangle$, we note that

$$\sum_{j=1}^N \lambda_j \pi^{ji} \in \Pi(\nu, \mu^i).$$

Hence,

$$\begin{aligned} \Psi_p(\nu) &= \sum_{i=1}^N \lambda_i \mathcal{W}_p^p\Big(\sum_{j=1}^N \lambda_j \mu^j, \mu_i\Big) \leq \sum_{i=1}^N \lambda_i \langle c_p, \sum_{j=1}^N \lambda_j \pi^{ji} \rangle \\ &= 2 \sum_{i<j}^N \lambda_i \lambda_j \mathcal{W}_p^p(\mu^i, \mu^j), \end{aligned}$$

such that

$$\frac{\Psi_p(\nu)}{\Psi_p(\hat{\nu})} \leq 2.$$

(iii) This follows similarly as (ii) does by linearity of expectation.

□

In general, there is no polynomial-time algorithm that will achieve an error arbitrarily close to 1 with high probability, see [14]. In light of this result, it is interesting to see that it is possible to obtain a relative error bound of 2 as in [33], but without performing any computations. However, note that merely using a mixture of the inputs yields rather useless barycenter approximations in practice; consider, e.g., two distinct Dirac measures.

Although we will see that the bounds above are still more or less sharp for Algorithms 1–4, these algorithms perform a lot better in practice. Moreover, these bounds are typically drastically improved as soon as a specific problem is given, see Remark 4.6 and Sect. 5.

Using one of the mentioned trivial choices as initial measures, all algorithms above aim to improve the approximation quality using the mapping $G^p_{\lambda,\pi}$. Next, we show that given any approximate barycenter $\nu$, executing $G^p_{\lambda,\pi}$ on $\nu$ never makes the approximation worse, if we choose the OT plans $\pi^i \in \Pi(\nu, \mu^i)$ to be optimal.

**Proposition 4.4** *Given a discrete* $\nu = \sum_{k=1}^{n_\nu} \nu_k \delta(y_k) \in \mathcal{P}(\mathbb{R}^d)$, $p \geq 1$, *let*

$$\pi^i := \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \delta(y_k, x^i_l) \in \mathit{argmin}_{\pi \in \Pi(\nu,\mu^i)} \langle c_p, \pi \rangle$$

*be optimal transport plans. Then it holds*

$$\Psi_p(G^p_{\lambda,\pi}(\nu)) \leq \Psi_p(\nu).$$

***Proof*** By definition of $\pi^i$, we have for all $i = 1, \dots, N$ that

$$\mathcal{W}^p_p(\nu, \mu^i) = \langle c_p, \pi^i \rangle = \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \|y_k - x^i_l\|^p.$$

Set

$$\tilde{\pi}^i := \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \delta(m_k, x^i_l) \in \Pi(G^p_{\lambda,\pi}(\nu), \mu^i),$$

where $m_k = M^p_{\lambda,\pi}(y_k)$. Then it holds that

$$\begin{aligned}
\Psi_p(G^p_{\lambda,\pi}(\nu)) &= \sum_{i=1}^{N} \lambda_i \mathcal{W}^p_p(G^p_{\lambda,\pi}(\nu), \mu^i) \leq \sum_{i=1}^{N} \lambda_i \langle c_p, \tilde{\pi}^i \rangle \\
&= \sum_{i=1}^{N} \lambda_i \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \|m_k - x^i_l\|^p = \sum_{k=1}^{n_\nu} \nu_k \sum_{i=1}^{N} \lambda_i \sum_{l=1}^{n_i} \frac{\pi^i_{k,l}}{\nu_k} \|m_k - x^i_l\|^p \\
&= \sum_{k=1}^{n_\nu} \nu_k \min_{m \in \mathbb{R}^d} \sum_{i=1}^{N} \lambda_i \sum_{l=1}^{n_i} \frac{\pi^i_{k,l}}{\nu_k} \|m - x^i_l\|^p
\end{aligned}$$

$$\leq \sum_{k=1}^{n_\nu} \nu_k \sum_{i=1}^{N} \lambda_i \sum_{l=1}^{n_i} \frac{\pi^i_{k,l}}{\nu_k} \|y_k - x^i_l\|^p = \sum_{i=1}^{N} \lambda_i \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \|y_k - x^i_l\|^p$$
$$= \sum_{i=1}^{N} \lambda_i \mathcal{W}_p^p(\nu, \mu^i) = \Psi_p(\nu).$$

□

Combining the results above, we immediately get the following error bounds for the algorithms introduced in Sect. 3.

**Corollary 4.5** *Let $p \in \{1, 2\}$ and let $\hat{\nu}$ be an optimal barycenter.*

*(i) If $\tilde{\nu}$ is obtained by Algorithm 1 (case $p = 2$) or Algorithm 2 (case $p = 1$), then it holds that*

$$\frac{\Psi_p(\tilde{\nu})}{\Psi_p(\hat{\nu})} \leq \frac{1}{\lambda_1} \quad or \quad \frac{\Psi_p(\tilde{\nu})}{\Psi_p(\hat{\nu})} \leq \frac{1+\varepsilon}{\lambda_1},$$

*respectively. Moreover, if instead the reference measure is chosen randomly with probabilities equal to the corresponding $\lambda_i$, then*

$$\frac{\mathbb{E}[\Psi_p(\tilde{\nu})]}{\Psi_p(\hat{\nu})} \leq 2 \quad or \quad \frac{\mathbb{E}[\Psi_p(\tilde{\nu})]}{\Psi_p(\hat{\nu})} \leq 2(1+\varepsilon).$$

*(ii) If $\tilde{\nu}$ is obtained by Algorithm 3 (case $p = 2$) or Algorithm 4 (case $p = 1$), then it holds that*

$$\frac{\Psi_p(\tilde{\nu})}{\Psi_p(\hat{\nu})} \leq 2 \quad or \quad \frac{\Psi_p(\tilde{\nu})}{\Psi_p(\hat{\nu})} \leq 2(1+\varepsilon),$$

*respectively.*

***Proof*** This follows immediately by combining Propositions 4.3 and 4.4, and the fact that

$$\sum_{i=1}^{N} \lambda_i \sum_{l=1}^{n_i} \frac{\pi^i_{k,l}}{\nu_k} \|m - x^i_l\|$$

is only optimized by $m_k$ up to a factor $(1+\varepsilon)$ for every $k = 1, \ldots, n_\nu$ in the case $p = 1$. □

***Remark 4.6*** Next, we show how to improve on the 2-approximation bound for a specific given problem. We assume that we are given optimal or close to optimal transport plans

$$\pi^i = \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \delta(y_k, x^i_l) \in \Pi(\nu, \mu^i), \quad i = 1, \ldots, N.$$

In case of the pairwise algorithm (Algorithms 3 and 4), we use

$$\pi^i = \sum_{j=1}^N \lambda_j \pi^{ji} \in \Pi(\nu, \mu^i), \quad \text{where} \quad \pi^{ji} \in \operatorname{argmin}_{\pi \in \Pi(\mu^j, \mu^i)} \langle c_p, \pi \rangle.$$

Given our approximate barycenter

$$\tilde{\nu} = \sum_{k=1}^{n_\nu} \nu_k \delta(m_k), \quad m_k = M^p_{\lambda,\pi}(y_k),$$

consider again

$$\tilde{\pi}^i := \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \delta(m_k, x^i_l) \in \Pi(\tilde{\nu}, \mu^i).$$

Then

$$\Psi_p(\tilde{\nu}) = \sum_{i=1}^N \lambda_i \mathcal{W}^p_p(\tilde{\nu}, \mu^i) \leq \sum_{i=1}^N \lambda_i \langle c_p, \tilde{\pi}^i \rangle = \sum_{i=1}^N \lambda_i \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \|m_k - x^i_l\|^p.$$

Together with (4.4), this gives

$$\frac{\Psi_p(\tilde{\nu})}{\Psi_p(\hat{\nu})} \leq \frac{\sum_{i=1}^N \lambda_i \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \|m_k - x^i_l\|^p}{\sum_{i<j}^N \lambda_i \lambda_j \mathcal{W}^p_p(\mu^i, \mu^j)}. \tag{4.5}$$

In the case $p = 2$, since

$$m_k = M^p_{\lambda,\pi}(y_k) = \sum_{i=1}^N \lambda_i \sum_{l=1}^{n_i} \frac{\pi^i_{k,l}}{\nu_k} x^i_l \quad \text{with} \quad \sum_{i=1}^N \lambda_i \sum_{l=1}^{n_i} \frac{\pi^i_{k,l}}{\nu_k} = 1,$$

by incorporating (4.1), the denominator in (4.5) simplifies to

$$\begin{aligned}
\Psi_2(\tilde{\nu}) &\leq \sum_{k=1}^{n_\nu} \nu_k \sum_{i=1}^N \lambda_i \sum_{l=1}^{n_i} \frac{\pi^i_{k,l}}{\nu_k} \|m_k - x^i_l\|^2 \\
&= \sum_{k=1}^{n_\nu} \nu_k \Big( \sum_{i=1}^N \lambda_i \sum_{l=1}^{n_i} \frac{\pi^i_{k,l}}{\nu_k} \|y_k - x^i_l\|^2 - \|m_k - y_k\|^2 \Big) \\
&= \sum_{i=1}^N \lambda_i \sum_{k=1}^{n_\nu} \sum_{l=1}^{n_i} \pi^i_{k,l} \|m_k - x^i_l\|^2 - \sum_{k=1}^{n_\nu} \nu_k \|m_k - y_k\|^2
\end{aligned}$$

$$= \Psi_2(\nu) - \sum_{k=1}^{n_\nu} \nu_k \|m_k - y_k\|^2,$$

such that by Proposition 4.3 (ii), we get

$$\frac{\Psi_2(\tilde{\nu})}{\Psi_2(\hat{\nu})} \leq 2 - \frac{\sum_{k=1}^{n_\nu} \nu_k \|m_k - y_k\|^2}{\sum_{i<j}^{N} \lambda_i \lambda_j \mathcal{W}_2^2(\mu^i, \mu^j)}. \tag{4.6}$$

Either way, for both $p = 1, 2$, the right-hand sides of (4.5) and (4.6) can be evaluated with almost no computational overhead after the execution of Algorithms Algorithms 3 and 4, since the optimal transport plans $\pi^{ij}$ between $\mu^i$ and $\mu^j$ have already been computed. This usually gives bounds much closer to one than the worst-case guarantees in Corollary 4.5.

Finally, we discuss the sharpness of the bounds in Corollary 4.5.

**Proposition 4.7** *Let $N \geq 2$ and consider the case with $\lambda = (\frac{1}{N}, \dots, \frac{1}{N}) \in \Delta_N$. There exist measures $\mu^1, \mu^2 = \mu^3 = \cdots = \mu^N$, such that if $\hat{\nu}$ is an optimal barycenter, the following hold true:*

*(i) Let $\tilde{\nu}$ be computed with Algorithm 1, then*

$$\frac{\Psi_2(\tilde{\nu})}{\Psi_2(\hat{\nu})} = N = \frac{1}{\lambda_1}.$$

*If the reference measure is chosen uniformly at random, then*

$$\frac{\mathbb{E}[\Psi_2(\tilde{\nu})]}{\Psi_2(\hat{\nu})} = 2 - \frac{1}{N} \overset{N\to\infty}{\longrightarrow} 2.$$

*(ii) Let $\tilde{\nu}$ be computed with Algorithm 2, then*

$$\frac{\Psi_1(\tilde{\nu})}{\Psi_1(\hat{\nu})} = N - 1 = \frac{1}{\lambda_1} - 1.$$

*If the reference measure is chosen uniformly at random, then*

$$\frac{\mathbb{E}[\Psi_1(\tilde{\nu})]}{\Psi_1(\hat{\nu})} = 2\Big(1 - \frac{1}{N}\Big) \overset{N\to\infty}{\longrightarrow} 2.$$

*(iii) Let $\tilde{\nu}$ be computed with Algorithm 3, then*

$$\frac{\Psi_2(\tilde{\nu})}{\Psi_2(\hat{\nu})} \geq \frac{N-1}{N}\Big(1 + \frac{N-1}{N}\Big) \overset{N\to\infty}{\longrightarrow} 2.$$

*(iv) Let $\tilde{\nu}$ be computed with Algorithm 4, then*

$$\frac{\Psi_1(\tilde{\nu})}{\Psi_1(\hat{\nu})} = 2 - \frac{1}{N} \overset{N\to\infty}{\longrightarrow} 2.$$

***Proof*** We consider

$$\mu^1 := \delta(0), \qquad \mu^2 = \ldots = \mu^N := \frac{1}{2}(\delta(-1) + \delta(1)).$$

(i) For $\pi^i$ defined as in Algorithm 1, it holds

$$\pi^i = \frac{1}{2}(\delta(0,-1) + \delta(0,1)), \qquad i = 2, \ldots, N.$$

and thus

$$\tilde{\nu} = \delta\Big(\frac{1}{2}(-1+1)\Big) = \delta(0) = \mu^1.$$

Thus,

$$\Psi_2(\tilde{\nu}) = \sum_{i=1}^N \lambda_i \mathcal{W}_2^2(\tilde{\nu}, \mu^i) = \frac{N-1}{N}.$$

On the other hand, consider

$$\nu = \frac{1}{2}\Big(\delta\Big(-\frac{N-1}{N}\Big) + \delta\Big(\frac{N-1}{N}\Big)\Big),$$

then

$$\Psi_2(\nu) = \sum_{i=1}^N \lambda_i \mathcal{W}_2^2(\nu, \mu^i) = \frac{1}{N}\Big(\Big(\frac{N-1}{N}\Big)^2 + (N-1)\Big(\frac{1}{N}\Big)^2\Big) = \frac{N-1}{N^2},$$

such that

$$\frac{\Psi_2(\tilde{\nu})}{\Psi_2(\hat{\nu})} \geq \frac{\Psi_2(\tilde{\nu})}{\Psi_2(\nu)} = N = \frac{1}{\lambda_1}.$$

(ii) We only need to compute the following medians:

$$\operatorname{argmin}_{m\in\mathbb{R}^d} \frac{1}{N}\|0-m\| + \frac{1}{2}\sum_{i=2}^N \frac{1}{N}(\|-1-m\| + \|1-m\|) = 0,$$

$$\operatorname{argmin}_{m\in\mathbb{R}^d} \frac{1}{N}\|0-m\| + \sum_{i=2}^N \frac{1}{N}(\|-1-m\|) = -1, \quad \text{and}$$

$$\operatorname{argmin}_{m \in \mathbb{R}^d} \frac{1}{N}\|0 - m\| + \sum_{i=2}^{N} \frac{1}{N}(\|1 - m\|) = 1.$$

Then we see that $\tilde{\nu} = \mu^1$, such that

$$\Psi_1(\tilde{\nu}) = \sum_{i=1}^{N} \lambda_i \mathcal{W}_1(\mu^1, \mu^i) = \frac{1}{N} \cdot 0 + \Big(1 - \frac{1}{N}\Big) \cdot 1 = 1 - \frac{1}{N},$$

and for any $j \in \{2, \ldots, N\}$,

$$\Psi_1(\mu^j) = \frac{1}{N} \cdot 1 + \Big(1 - \frac{1}{N}\Big) \cdot 0 = \frac{1}{N},$$

which leads to

$$\frac{\Psi_1(\tilde{\nu})}{\Psi_1(\hat{\nu})} \geq \frac{\Psi_1(\tilde{\nu})}{\Psi_1(\mu^j)} = N - 1 = \frac{1}{\lambda_1} - 1.$$

For the randomized case, we get

$$\begin{aligned}\mathbb{E}[\Psi_1(\tilde{\nu})] &= \frac{1}{N}\Psi_1(\mu^1) + \Big(1 - \frac{1}{N}\Big)\Psi_1(\mu^j) = \frac{1}{N} \cdot \Big(1 - \frac{1}{N}\Big) + \Big(1 - \frac{1}{N}\Big) \cdot \frac{1}{N} \\ &= 2\frac{1}{N}\Big(1 - \frac{1}{N}\Big),\end{aligned}$$

such that

$$\frac{\mathbb{E}[\Psi_1(\tilde{\nu})]}{\Psi_1(\hat{\nu})} \geq \frac{2\frac{1}{N}(1 - \frac{1}{N})}{\frac{1}{N}} = 2\Big(1 - \frac{1}{N}\Big).$$

(iii) We get for $i = 2, \ldots, N$ that

$$\pi^{ij} = \begin{cases} \frac{1}{2}(\delta(-1, 0) + \delta(1, 0)), & j = 1, \\ \frac{1}{2}(\delta(-1, -1) + \delta(1, 1)), & j = 2, \ldots, N, \end{cases}$$

and hence

$$\begin{aligned}\tilde{\nu}^i &= \frac{1}{2}\Big(\delta\Big(\frac{N-1}{N} \cdot (-1) + \frac{1}{N} \cdot 0\Big) + \Big(\delta\Big(\frac{N-1}{N} \cdot 1 + \frac{1}{N} \cdot 0\Big)\Big) \\ &= \frac{1}{2}\Big(\delta\Big(-\frac{N-1}{N}\Big) + \delta\Big(\frac{N-1}{N}\Big)\Big).\end{aligned}$$

Thus,

$$\tilde{\nu} = \frac{1}{N}\delta(0) + \frac{N-1}{2N}\Big(\delta\Big(-\frac{N-1}{N}\Big) + \delta\Big(\frac{N-1}{N}\Big)\Big).$$

Hence, it is easy to compute that

$$\mathcal{W}_2^2(\tilde{\nu}, \mu^i) = \begin{cases} \frac{N-1}{N}(\frac{N-1}{N})^2 = (\frac{N-1}{N})^3, & i = 1 \\ \frac{1}{N}(\frac{N-1}{N})^2 + \frac{N-1}{N}(\frac{1}{N})^2 = \frac{1}{N^3}N(N-1) = \frac{N-1}{N^2}, & i = 2, \dots, N, \end{cases}$$

such that

$$\Psi_2(\tilde{\nu}) = \frac{1}{N}\Big(\Big(\frac{N-1}{N}\Big)^3 + (N-1)\Big(\frac{N-1}{N^2}\Big)\Big) = \frac{N-1}{N^2}\Big(\Big(\frac{N-1}{N}\Big)^2 + \frac{N-1}{N}\Big).$$

Finally, considering

$$\nu = \frac{1}{2}\Big(\delta\Big(-\frac{N-1}{N}\Big) + \delta\Big(\frac{N-1}{N}\Big)\Big),$$

we get

$$\frac{\Psi_2(\tilde{\nu})}{\Psi_2(\hat{\nu})} \geq \frac{\Psi_2(\tilde{\nu})}{\Psi_2(\nu)} = \Big(\frac{N-1}{N}\Big)^2 + \frac{N-1}{N} = \frac{N-1}{N}\Big(1 + \frac{N-1}{N}\Big) \overset{N\to\infty}{\longrightarrow} 2.$$

(iv) In this case, we get

$$\tilde{\nu} = \frac{1}{N}\delta(1) + \frac{N-1}{2N}\Big(\delta(-1) + \delta(1)\Big).$$

Compute

$$\Psi_1(\tilde{\nu}) = \sum_{i=1}^{N} \lambda_i \mathcal{W}_1(\tilde{\nu}, \mu^i) = \frac{1}{N} \cdot \frac{N-1}{N} + \frac{N-1}{N} \cdot \frac{1}{N} = \frac{2(N-1)}{N^2}.$$

On the other hand, for any $j \in \{2, \dots, N\}$,

$$\Psi_1(\nu^j) = \frac{1}{N}\mathcal{W}_1(\nu^j, \nu^1) = \frac{1}{N},$$

such that

$$\frac{\Psi_1(\tilde{\nu})}{\Psi_1(\hat{\nu})} \geq \frac{\Psi_1(\tilde{\nu})}{\Psi_1(\nu^j)} = 2\frac{\frac{N-1}{N^2}}{\frac{1}{N}} = 2\Big(1 - \frac{1}{N}\Big) \overset{N\to\infty}{\longrightarrow} 2.$$

□

***Remark 4.8*** Intuitively, the example used in the proof of Proposition 4.7 is based on the fact that the analyzed algorithms can not split $\mu^1 = \delta(0)$ into two Dirac measures with weight $1/2$, in which case the approximations would be optimal. We chose the example in the proof for simplicity of exposition. However, it is also possible to show the same sharpness results using measures $\mu^1, \dots, \mu^N$ that all have two support points. To this end, for $N$ odd and some small $\varepsilon > 0$, consider

$$\begin{aligned} \mu^1 &:= \frac{1}{2}(\delta(0, -\varepsilon) + \delta(0, \varepsilon)), \\ \mu^2 = \mu^4 = \cdots = \mu^{N-1} &:= \frac{1}{2}(\delta(-1, -\varepsilon) + \delta(1, \varepsilon)) \\ \mu^3 = \mu^5 = \cdots = \mu^N &:= \frac{1}{2}(\delta(-1, \varepsilon) + \delta(1, -\varepsilon)). \end{aligned}$$

## 5 Numerical results

We present a numerical comparison of different Wasserstein-2 barycenter algorithms, the computation of a Wasserstein-1 barycenter, and, as applications, an interpolation between measures and textures, respectively. To compute the exact two-marginal transport plans of the presented algorithms, we used the `emd` function of the Python-OT (POT 0.7.0) package [55], which is a wrapper of the network simplex solver[1] from [8], which, in turn, is based on an implementation in the LEMON C++ library.[2]

### 5.1 Numerical comparison

In this section, we compare different Wasserstein-2 barycenter algorithms in terms of accuracy and runtime. We would like to include popular algorithms as iterative Bregman projections into the comparison. However, many of these algorithms operate in a fixed-support setting, that is, they only optimize over the weights of some a priori chosen support grid. On the other hand, free-support methods are the ideal candidate for sparse and possibly high-dimensional point cloud data, i.e., if such a grid structure is not present. An approximation of such data with a coarse grid decreases the accuracy of the solution, but a fine grid increases the runtime of the fixed-support methods. Hence, the fair choice of a comparison data set is challenging.

We attempt to solve this problem by choosing a grid data set with relatively few nonzero mass weights, that has nevertheless been commonly used as a benchmark example in the literature, also for fixed-support algorithms. It originates from [15] and consists of $N = 10$ ellipses shown in Fig. 2, given as images of $60 \times 60$ pixels. We take $\lambda \equiv 1/N$.

First, we compute approximate barycenters $\tilde{\nu}$ using the presented algorithms in the case $p = 2$, which we call "Reference" and "Pairwise" below.[3] Furthermore, we compute the barycenter using publicly available implementations for the methods [18, 21, 36], called "Debiased", "IBP", "Product", "MAAIPM" and "Frank–Wolfe"

[1] https://perso.liris.cnrs.fr/nicolas.bonneel/FastTransport/.

[2] http://lemon.cs.elte.hu/pub/doc/latest-svn/index.html.

[3] https://github.com/jvlindheim/free-support-barycenters.

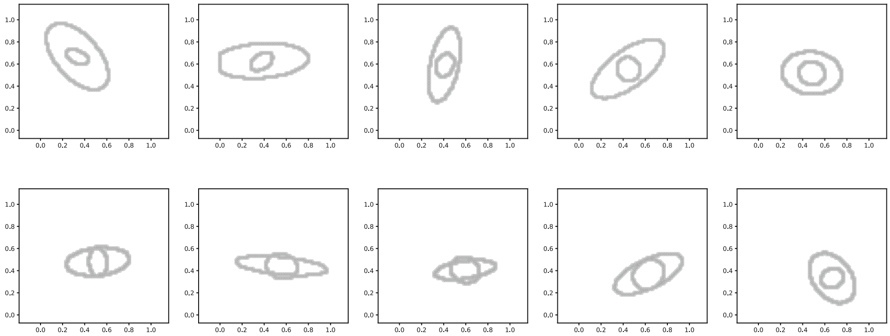


**Fig. 2** Data set of 10 nested ellipses

below,[4] the exact barycenter method from [30] called “Exact” below,[5] and the method from [23] called “FastIBP” below.[6] We also tried the BADMM[7] method from [56], but since it did not converge properly, we do not consider it further.

While the fixed-support methods receive the input measures supported on $\{0, \dots, 59/60\} \times \{0, \dots, 59/60\}$ as gray-valued $60 \times 60$ images, the free-support methods get the measures as a list of support positions and corresponding weights. Clearly, the sparse support of the data is an advantage for the free-support methods. As a means to facilitate the comparison, we execute the reference and pairwise algorithms also as fixed-support versions. Instead of computing optimal solutions in Algorithms 1 and 3 and, we approximate the optimal transport plans $\pi^{ij}$ using the Sinkhorn algorithm on the full grid. We call these algorithms “Reference full” and “Pairwise full” below. Note that, as do the implementations of “IBP”, “Debiased” and “Product”, we exploit the fact that the Sinkhorn kernel $K = \exp(-c/\varepsilon)$ is separable, such that the corresponding convolution can be performed separately in $x$- and $y$-direction, see, e.g., [13, Rem. 4.17]. This also reduces memory consumption, since it is not necessary to compute a distance matrix in $\mathbb{R}^{3600\times 3600}$. We remark that the runtime of the Sinkhorn algorithms crucially depends on the desired accuracy. In analogy to “IBP”, “Debiased” and “Product” that terminate, once the barycenter measure has a maximum change of $10^{-5}$ in any iteration, we terminate once this tolerance is reached in the first marginal of $\pi^{ij}$. We check for this criterion only every 10-th iteration, since it produces computational overhead (contrary to the aforementioned methods).

For all Sinkhorn methods, we used a parameter of $\varepsilon = 0.002$ and otherwise chose the default parameters. For the reference algorithm, we have chosen the reference measure to be the upper left measure shown in Fig. 2. To compare the runtimes, we executed all codes on the same laptop with Intel i7-8550U CPU and 8GB memory. The Matlab codes were run in Matlab R2020a. The runtimes of the Python codes are averages over several runs, as obtained by Python’s `timeit` function. The results are shown in Fig. 3 and Table 1.

[4] https://github.com/hichamjanati/debiased-ot-barycenters.

[5] https://github.com/eboix/high_precision_barycenters.

[6] https://github.com/tyDLin/FS-WBP.

[7] https://github.com/bobye/WBC_Matlab.

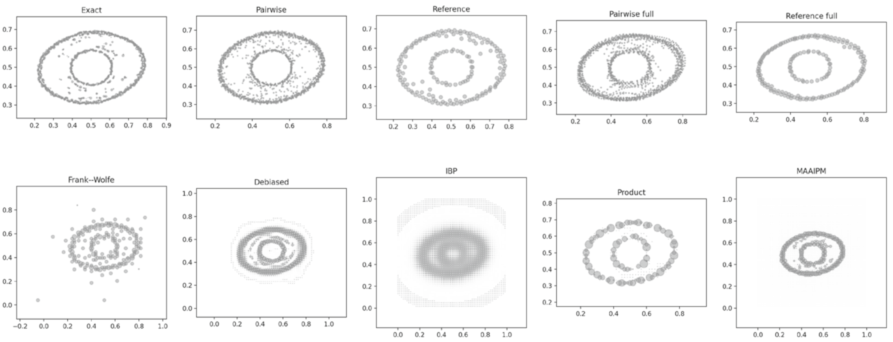


**Fig. 3** Barycenters for data set in Fig. 2 computed by different methods. The weight of a support point is indicated by its area in the plot

**Table 1** Numerical results for the ellipse barycenter problem. The runtime is measured in seconds. The ranking is the sum of the standard scores of the logarithm of the relative error and the runtime, respectively. The best values of all approximative algorithms are highlighted in bold

| | $\Psi(\tilde{\nu})$ | $\Psi(\tilde{\nu})/\Psi(\hat{\nu})$ | Runtime | Ranking | Free support |
|---|---|---|---|---|---|
| Reference | 0.02683 | 1.0061 | **0.0501** | **−1.65** | ✓ |
| Pairwise | 0.02669 | **1.0012** | 0.3095 | −1.41 | ✓ |
| Pairwise full | 0.02678 | 1.0042 | 0.5092 | −1.14 | ✓ |
| Debiased | 0.02675 | 1.0033 | 1.5061 | −0.90 | × |
| Reference full | 0.02716 | 1.0186 | 0.1128 | −0.84 | ✓ |
| IBP | 0.02723 | 1.0214 | 0.0914 | −0.76 | × |
| Product | 0.02688 | 1.0082 | 21.2982 | 0.02 | × |
| MAAIPM | 0.02672 | 1.0020 | 158.5085 | 0.24 | × |
| Exact | 0.02666 | 1.0000 | 18187.6740 | 1.38 | ✓ |
| FastIBP | 0.02753 | 1.0323 | 111.0340 | 1.59 | × |
| Frank–Wolfe | 0.02870 | 1.0763 | 68.2480 | 3.48 | ✓ |

While the exact method has a very high runtime, no approximative method achieves a perfect relative error of $\Psi_2(\tilde{\nu})/\Psi_2(\hat{\nu}) = 1$. However, the error is well below 2 for all methods, which is a lot better than the worst case bounds shown above. In fact, using the problem-adapted bounds as outlined in Remark 4.6, without knowledge of $\hat{\nu}$, the pairwise algorithm already guarantees a relative error of at most 1.64%. Whereas the pairwise algorithm achieves the lowest error of all approximative algorithms with around 0.12%, the reference algorithm achieves the lowest runtime of 0.05 seconds. Notably, the FastIBP method is a lot slower than IBP whilst producing a more blurry result, which might indicate an implementation issue. While the Frank–Wolfe method suffers from outliers, the support of most fixed-support methods is more extended than exact barycenter's support, since Sinkhorn-barycenters have dense support.

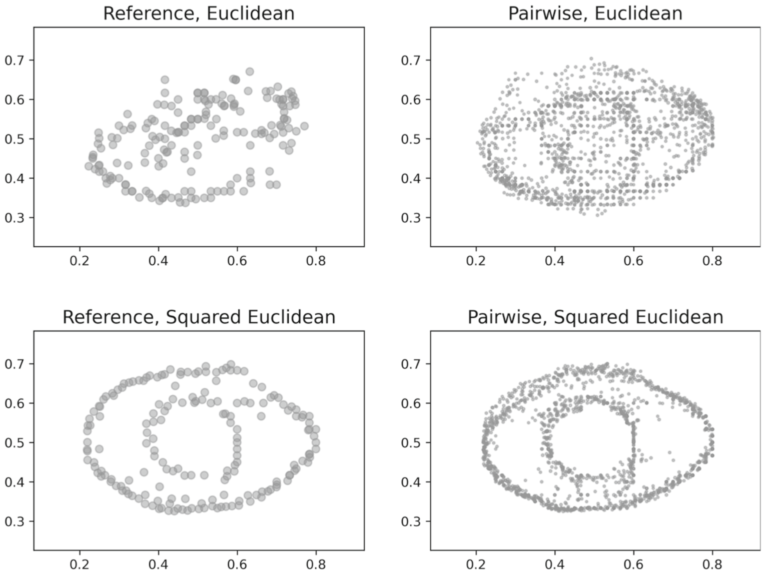


**Fig. 4** Barycenters computed with Algorithms 2 and 4 and for the data set in Fig. 2 and cost functions $c_1(x, y) = \|x - y\|$ and $c_2(x, y) = \|x - y\|^2$. The weight of a support point is indicated by its area in the plot

We attempt to measure the best compromise between low error and runtime by means of the sum of the standard scores of the logarithmic relative errors and runtimes, respectively, where the standard score or zscore is the value normalized by the population mean and standard deviation. Table 1 is sorted according to this ranking score. The reference and pairwise algorithm are the best with respect to this metric. As expected, the full-support versions of the reference and pairwise algorithms have worse runtime and also accuracy, which can likely be explained by the errors of the Sinkhorn algorithm. Nevertheless, they offer a competitive tradeoff between speed and accuracy with respect to the other methods, which shows that the advantage of the framework considered in this paper is not only due to the sparse support of the chosen data set. Altogether, the results of the proposed algorithms look promising.

### 5.2 Wasserstein-1 barycenters

Next, we compute approximate Wasserstein-1 barycenters of the same data set as in the previous Sect. 5.1 using the Algorithms 2 and 4. The results are depicted in Fig. 4 in the top row.

Note that the elliptic structure of the barycenter is only retained to some degree, which can probably be explained by the choice of $c_1$ as the cost function. For example, it is easy to show that the OT plans corresponding to $\mathcal{W}_2^2$ are translation equivariant. On the other hand, this property fails for any other $p \in [1, 2) \cup (2, \infty]$, as it is easy to derive from the example with $\mu, \nu \in \mathcal{P}(\mathbb{R}^2)$ defined by

$$\mu := \frac{1}{2}(\delta(0,0) + \delta(1,0)), \qquad \nu := \frac{1}{2}(\delta(0,0) + \delta(0,1)).$$

Thus, we also execute algorithms Algorithms 2 and 4, where we swap $c_1$ for the squared Euclidean costs $c_2$ in order to compute the OT plans $\pi^{ij} \in \Pi(\mu^i, \mu^j)$, but continue to compute the barycenter support using Weiszfeld's algorithm. The results are shown in Fig. 4 in the bottom row.

Now the elliptic structure is preserved a lot better and the results are very similar to the Wasserstein-2 barycenters. We conclude that the choice of cost function had a larger impact on the results than whether the barycenter support is constructed using the means or geometric medians. Algorithms 2 and 4 with $c_2$ thus seem like an interesting alternative to Algorithms 1 and 3 in the case where one expects outlier measures, since the median is more robust to outliers than the mean, see, e.g. [57].

### 5.3 Multiple different sets of weights

For this numerical application, we compute barycenters between four given measures for multiple sets of weights $\lambda^k = (\lambda_1^k, \dots, \lambda_N^k)$, $\lambda^k \in \Delta_4$, $k = 1, \dots, K$, obtaining an interpolation between those measures. An advantage of the presented algorithms for that application is that the optimal transport plans between the input measures, which are the bottleneck computations, only need to be performed once, whereas the matrix multiplications for interpolations with new weights are fast. We use the proposed algorithms for a data set of four measures given as images of size $50 \times 50$, for sets of weights that bilinearly interpolate between the four unit vectors. The original measures are shown in the four corners of Fig. 6. For the reference algorithm, we use the upper left measure as the reference measure. The results are shown in Figs. 5 and 6.

While the running time of the reference algorithm is shorter, its solution has several artifacts, in particular when the weight $\lambda_1^k$ of the reference measure is low. On the other hand, through effectively averaging the reference algorithm for different choices of the reference measure, the pairwise algorithm is able smooth out some of these artifacts. We compare the results of both algorithms for $\lambda = (0.04, 0.16, 0.16, 0.64)$ in Fig. 7. We also computed the upper error bound $\eta$ of the pairwise algorithm given by (4.6) exemplarily for uniform weights, which is 3.6%.

### 5.4 Texture interpolation

For another application, we lift the experiment of Sect. 5.3 from interpolation of measures in Euclidean space to interpolation of textures via the synthesis method from [6], using their publicly available source code.[8] While the authors already interpolated between two different textures in that paper, requiring only the solution of a two-marginal optimal transport problem to obtain a barycenter, we can do this for multiple textures using approximate barycenters for multiple measures. Briefly, the authors proposed to encode a texture as a collection of smaller patches $F_j$, where each, say, $4 \times 4$-patch is encoded as a point $x_j \in \mathbb{R}^{16}$. The texture is then modeled as a "fea-

[8] https://github.com/ahoudard/wgenpatex.

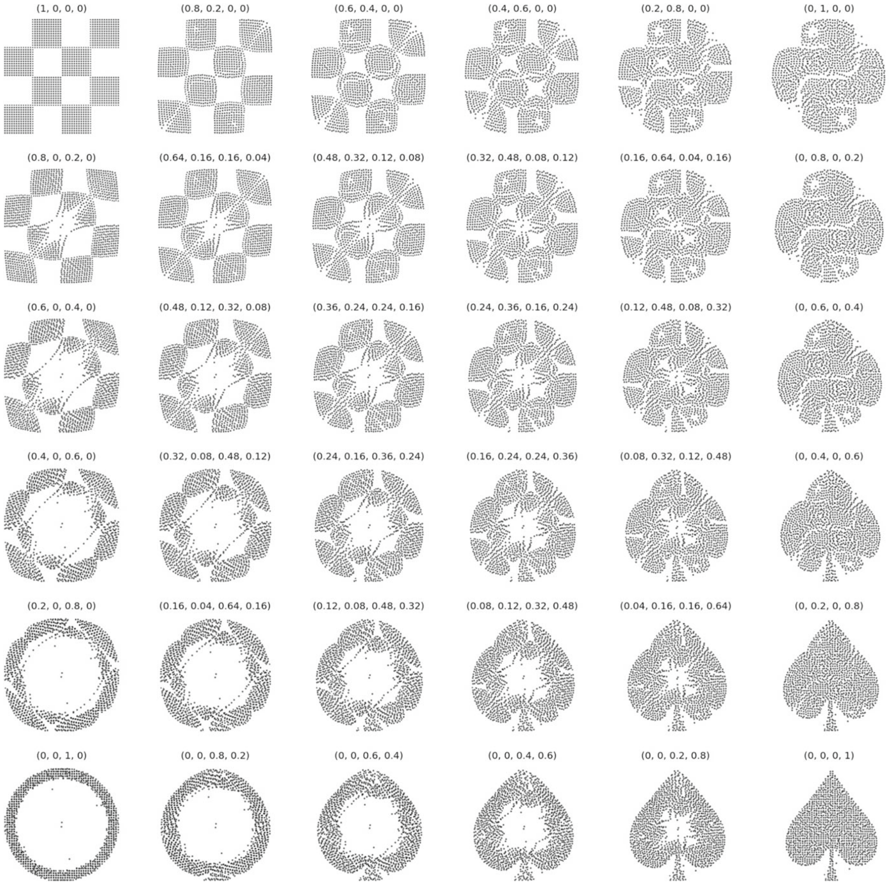


**Fig. 5** Approximate barycenters for different sets of weights computed by Algorithm 1

ture measure" $\frac{1}{M}\sum_{j=1}^{M}\delta(x_j) \in \mathcal{P}(\mathbb{R}^{16})$, such that this description is invariant under different positions of its patches within the image. Finally, this is repeated for image patches at several scales $s$, obtaining a collection of measures $(\mu^s)$, $s = 1, \dots, S$. Synthesizing an image is done by optimizing an optimal transport loss between its feature measure and some reference measure (and then summing over $s$), as obtained, e.g., from a reference image. Thus, the synthesized image tries to imitate the reference image in terms of its feature measures. Here, we choose four texture images of size 256 × 256 from the "Describable Textures Dataset" [58]. We compute their feature measures $\mu^{1,s}, \dots, \mu^{4,s}$ for each scale. Next, as in Sect. 5.3, we compute approximate barycenters $\tilde{\nu}^{k,s}$ for all $k$ and $s$ using the reference algorithm, where $k$ runs over different sets of weights, and perform the image synthesis for each $k$ using the $\tilde{\nu}^{k,s}$ as feature measures to imitate. The results are shown in Fig. 8. Using this approach, one obtains a visually pleasing interpolation between the four given textures.

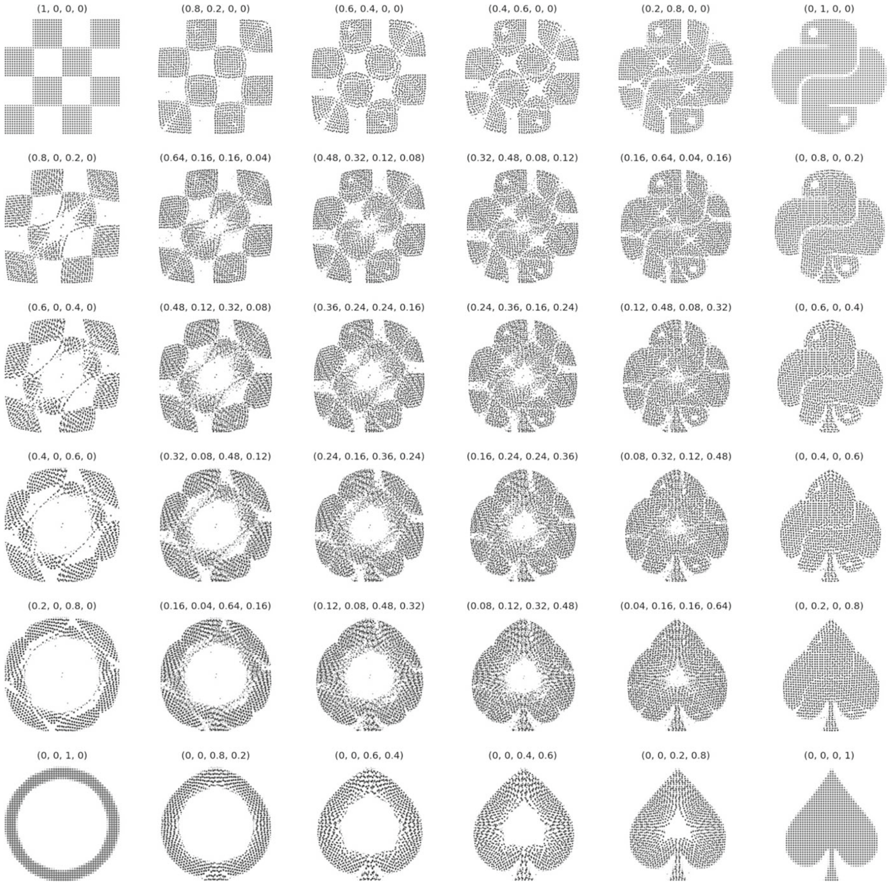
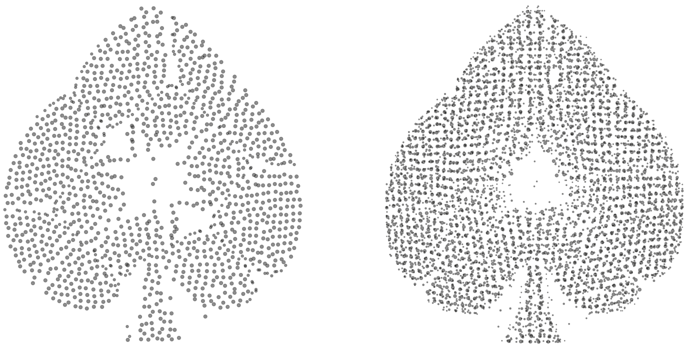


**Fig. 6** Approximate barycenters for different sets of weights computed by Algorithm 3



(a) Algorithm 1 (reference).

(b) Algorithm 3 (pair wise).

**Fig. 7** Close comparison of two approximate barycenters of the reference and pairwise algorithms for the weights $\lambda = (0.04, 0.16, 0.16, 0.64)$

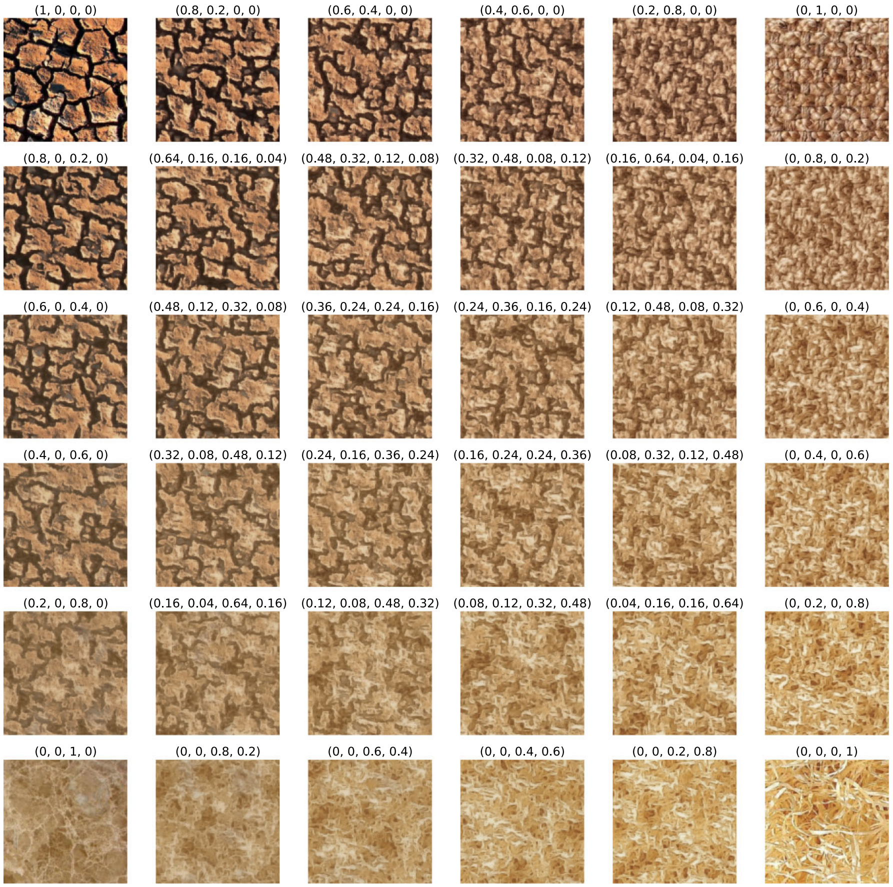


**Fig. 8** Interpolation of four different textures that are displayed in the four corners. The weight set for the barycenter computations performed for each image is shown above each synthesized image

## 6 Conclusion

In this paper, we derived two straightforward algorithms from a well-known framework for Wasserstein-$p$ barycenters for $p = 1, 2$. We analyzed them theoretically and practically, showing that they are easy to implement, produce sparse solutions and are thus memory-efficient. We validated their speed and precision using numerical examples.

In the future, it would be interesting to generalize the discussed algorithms and bounds to other $p \geq 1$. For instance, for $p = \infty$, the barycentric map $M^p_{\lambda,\pi}$ corresponds to the solution of the so-called smallest-sphere-problem, which can be solved by Welzl's algorithm [59]. Finding a lower bound as in Proposition 4.2 for general $p \geq 1$ is not straightforward, since the proofs of (4.2) and (4.3) are specific to $p = 2$ and $p = 1$.

**Acknowledgements** Many thanks to Gabriele Steidl and Florian Beier for fruitful discussions.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Data availability** The datasets generated during and/or analysed during the current study are available in the GitHub repositories https://github.com/jvlindheim/free-support-barycenters and https://www.robots.ox.ac.uk/\protect\unhbox\voidb@x\penalty\@M\vgg/data/dtd/.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

## References

1. Agueh, M., Carlier, G.: Barycenters in the Wasserstein space. SIAM J. Math. Anal. **43**(2), 904–924 (2011). https://doi.org/10.1137/100805741
2. Puccetti, G., Rüschendorf, L., Vanduffel, S.: On the computation of Wasserstein barycenters. J. Multivariate Anal. **176**, 104581 (2020). https://doi.org/10.1016/j.jmva.2019.104581
3. Turner, K., Mileyko, Y., Mukherjee, S., Harer, J.: Fréchet means for distributions of persistence diagrams. Discrete Comput. Geom. **52**(1), 44–70 (2014). https://doi.org/10.1007/s00454-014-9604-7
4. Trouvé, A., Younes, L.: Local geometry of deformable templates. SIAM J. Math. Anal. **37**(1), 17–59 (2005). https://doi.org/10.1137/S0036141002404838
5. Zemel, Y., Panaretos, V.M.: Fréchet means and Procrustes analysis in Wasserstein space. Bernoulli **25**(2), 932–976 (2019). https://doi.org/10.3150/17-bej1009
6. Houdard, A., Leclaire, A., Papadakis, N., Rabin, J.: A generative model for texture synthesis based on optimal transport between feature distributions. J. Math. Imaging Vis. (2022). https://doi.org/10.1007/s10851-022-01108-9
7. Rabin, J., Peyré, G., Delon, J., Bernot, M.: Wasserstein barycenter and its application to texture mixing. In: international conference on scale space and variational methods in computer vision, pp 435–446 (2011). Springer
8. Bonneel, N., van de Panne, M., Paris, S., Heidrich, W.: Displacement interpolation using Lagrangian mass transport. In: Proceedings of the 2011 SIGGRAPH Asia Conference. SA '11. association for computing machinery, New York, NY, USA (2011). https://doi.org/10.1145/2024156.2024192
9. Solomon, J., de Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., Guibas, L.: Convolutional Wasserstein distances: efficient optimal transportation on geometric domains. ACM Trans. Graph. **34**(4), 1–11 (2015). https://doi.org/10.1145/2766963
10. Elvander, F., Haasler, I., Jakobsson, A., Karlsson, J.: Multi-marginal optimal transport using partial information with applications in robust localization and sensor fusion. Signal Process. **171**, 107474 (2020). https://doi.org/10.1016/j.sigpro.2020.107474
11. Srivastava, S., Li, C., Dunson, D.B.: Scalable Bayes via barycenter in Wasserstein space. J. Mach. Learn. Res. **19**, 8–35 (2018)
12. Panaretos, V.M., Zemel, Y.: Statistical aspects of Wasserstein distances. Annu. Rev. Stat. Appl. **6**, 405–431 (2019). https://doi.org/10.1146/annurev-statistics-030718-104938
13. Peyré, G., Cuturi, M., et al.: Computational optimal transport: with applications to data science. Found. Trends Mach. Learn. **11**(5–6), 355–607 (2019)
14. Altschuler, J.M., Boix-Adserà, E.: Wasserstein barycenters are NP-hard to compute. SIAM J. Math. Data Sci. **4**(1), 179–203 (2022). https://doi.org/10.1137/21M1390062

15. Cuturi, M., Doucet, A.: Fast computation of Wasserstein barycenters. In: international conference on machine learning, pp. 685–693 (2014). PMLR
16. Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., Peyré, G.: Iterative Bregman projections for regularized transportation problems. SIAM J. Sci. Comput. **37**(2), 1111–1138 (2015). https://doi.org/10.1137/141000439
17. Kroshnin, A., Tupitsa, N., Dvinskikh, D., Dvurechensky, P., Gasnikov, A., Uribe, C.: On the complexity of approximating Wasserstein barycenters. In: Chaudhuri, K., Salakhutdinov, R. (eds.) proceedings of the 36th international conference on machine learning. Proceedings of Machine Learning Research, vol. 97, pp. 3530–3540. PMLR, Long Beach, California, USA (2019). https://proceedings.mlr.press/v97/kroshnin19a.html
18. Ge, D., Wang, H., Xiong, Z., Ye, Y.: Interior-point methods strike back: Solving the Wasserstein barycenter problem. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in neural information processing systems, vol. 32. Curran Associates, Inc., Vancouver Convention Center, Vancouver, CA (2019). https://proceedings.neurips.cc/paper/2019/file/0937fb5864ed06ffb59ae5f9b5ed67a9-Paper.pdf
19. Yang, L., Li, J., Sun, D., Toh, K.-C.: A fast globally linearly convergent algorithm for the computation of Wasserstein barycenters. J. Mach. Learn. Res. **22**(21), 1–37 (2021)
20. Lin, T., Ho, N., Cuturi, M., Jordan, M.I.: On the complexity of approximating multimarginal optimal transport. J. Mach. Learn. Res. **23**(65), 1–43 (2022)
21. Janati, H., Cuturi, M., Gramfort, A.: Debiased Sinkhorn barycenters. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research, vol. 119, pp. 4692–4701. PMLR, virtual (2020). http://proceedings.mlr.press/v119/janati20a.html
22. Takezawa, Y., Sato, R., Kozareva, Z., Ravi, S., Yamada, M.: Fixed support tree-sliced Wasserstein barycenter. In: Camps-Valls, G., Ruiz, F.J.R., Valera, I. (eds.) Proceedings of the 25th international conference on artificial intelligence and statistics. Proceedings of Machine Learning Research, vol. 151, pp. 1120–1137. PMLR, virtual (2022). https://proceedings.mlr.press/v151/takezawa22a.html
23. Lin, T., Ho, N., Chen, X., Cuturi, M., Jordan, M.: Fixed-support Wasserstein barycenters: Computational hardness and fast algorithm. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in neural information processing systems, vol. 33, pp. 5368–5380. Curran Associates, Inc., virtual (2020). https://proceedings.neurips.cc/paper/2020/file/3a029f04d76d32e79367c4b3255dda4d-Paper.pdf
24. Dvinskikh, D., Tiapkin, D.: Improved complexity bounds in Wasserstein barycenter problem. In: International conference on artificial intelligence and statistics, pp. 1738–1746 (2021). PMLR
25. Gangbo, W., Święch, A.: Optimal maps for the multidimensional Monge-Kantorovich problem. Comm. Pure Appl. Math. **51**(1), 23–45 (1998). https://doi.org/10.1002/(SICI)1097-0312(199801)51:1<23::AID-CPA2>3.0.CO;2-H
26. Benamou, J.-D., Carlier, G., Nenna, L.: Generalized incompressible flows, multi-marginal transport and Sinkhorn algorithm. Numer. Math. **142**(1), 33–54 (2019). https://doi.org/10.1007/s00211-018-0995-x
27. Haasler, I., Ringh, A., Chen, Y., Karlsson, J.: Multimarginal optimal transport with a tree-structured cost and the Schrödinger bridge problem. SIAM J. Control Optim. **59**(4), 2428–2453 (2021). https://doi.org/10.1137/20M1320195
28. Beier, F., von Lindheim, J., Neumayer, S., Steidl, G.: Unbalanced multi-marginal optimal transport. J. Math. Imaging Vis. (2022). https://doi.org/10.1007/s10851-022-01126-7
29. Anderes, E., Borgwardt, S., Miller, J.: Discrete Wasserstein barycenters: optimal transport for discrete data. Math. Methods Oper. Res. **84**(2), 389–409 (2016). https://doi.org/10.1007/s00186-016-0549-x
30. Altschuler, J.M., Boix-Adsera, E.: Wasserstein barycenters can be computed in polynomial time in fixed dimension. J. Mach. Learn. Res. **22**(44), 1–19 (2021)
31. Borgwardt, S., Patterson, S.: Improved linear programs for discrete barycenters. INFORMS J. Optim. **2**(1), 14–33 (2020). https://doi.org/10.1287/ijoo.2019.0020
32. Borgwardt, S., Patterson, S.: A column generation approach to the discrete barycenter problem. Discrete Optim. **43**, 100674 (2022). https://doi.org/10.1016/j.disopt.2021.100674
33. Borgwardt, S.: An lp-based, strongly-polynomial 2-approximation algorithm for sparse Wasserstein barycenters. Oper. Res. **22**(2), 1511–1551 (2020)
34. Qian, Y., Pan, S.: An inexact PAM method for computing Wasserstein barycenter with unknown supports. Comput. Appl. Math. **40**(2), 1–29 (2021). https://doi.org/10.1007/s40314-020-01395-1

35. Claici, S., Chien, E., Solomon, J.: Stochastic Wasserstein barycenters. In: Dy, J., Krause, A. (eds.) In: Proceedings of the 35th international conference on machine learning. proceedings of machine learning research, vol. 80, pp. 999–1008. PMLR, Stockholmsmässan, Stockholm, SE (2018). https://proceedings.mlr.press/v80/claici18a.html
36. Luise, G., Salzo, S., Pontil, M., Ciliberto, C.: Sinkhorn barycenters with free support via Frank–Wolfe algorithm. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in neural information processing systems, vol. 32, pp. 9322–9333. Curran Associates, Inc., Vancouver Convention Center, Vancouver, CA (2019). https://proceedings.neurips.cc/paper/2019/file/9f96f36b7aae3b1ff847c26ac94c604e-Paper.pdf
37. Li, L., Genevay, A., Yurochkin, M., Solomon, J.M.: Continuous regularized Wasserstein barycenters. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 17755–17765. Curran Associates, Inc., virtual (2020). https://proceedings.neurips.cc/paper/2020/file/cdf1035c34ec380218a8cc9a43d438f9-Paper.pdf
38. von Lindheim, J.: Approximative algorithms for multi-marginal optimal transport and free-support wasserstein barycenters. arXiv preprint arXiv:2202.00954 (2022)
39. Heinemann, F., Munk, A., Zemel, Y.: Randomized Wasserstein barycenter computation: resampling with statistical guarantees. SIAM J. Math. Data Sci. **4**(1), 229–259 (2022). https://doi.org/10.1137/20M1385263
40. Izzo, Z., Silwal, S., Zhou, S.: Dimensionality reduction for Wasserstein barycenter. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021). https://openreview.net/forum?id=cDPFOsj2G6B
41. Carlier, G., Ekeland, I.: Matching for teams. Econom. Theory **42**(2), 397–418 (2010). https://doi.org/10.1007/s00199-008-0415-z
42. Friesecke, G., Penka, M.: The GenCol algorithm for high-dimensional optimal transport: general formulation and application to barycenters and Wasserstein splines. arXiv preprint arXiv:2209.09081 (2022)
43. Ambrosio, L., Gigli, N., Savaré, G.: Gradient flows in metric spaces and in the space of probability measures, 2nd edn. Lectures in Mathematics ETH Zürich, p. 334. Birkhäuser Verlag, Basel, Basel, CH (2008)
44. Beier, F., Beinert, R., Steidl, G.: On a linear Gromov-Wasserstein distance. IEEE Trans. Image Process. **31**, 7292–7305 (2022). https://doi.org/10.1109/TIP.2022.3221286
45. Cai, T., Cheng, J., Schmitzer, B., Thorpe, M.: The linearized Hellinger-Kantorovich distance. SIAM J. Imaging Sci. **15**(1), 45–83 (2022). https://doi.org/10.1137/21M1400080
46. Wang, W., Slepčev, D., Basu, S., Ozolek, J.A., Rohde, G.K.: A linear optimal transportation framework for quantifying and visualizing variations in sets of images. Int. J. Comput. Vis. **101**(2), 254–269 (2013)
47. Mérigot, Q., Delalande, A., Chazal, F.: Quantitative stability of optimal transport maps and linearization of the 2-Wasserstein space. In: Chiappa, S., Calandra, R. (eds.) In: Proceedings of the twenty third international conference on artificial intelligence and statistics. proceedings of machine learning research, vol. 108, pp. 3186–3196. PMLR, virtual (2020). https://proceedings.mlr.press/v108/merigot20a.html
48. Moosmüller, C., Cloninger, A.: Linear optimal transport embedding: provable Wasserstein classification for certain rigid transformations and perturbations. Inf. Inference (2022). https://doi.org/10.1093/imaiai/iaac023
49. Álvarez-Esteban, P.C., del Barrio, E., Cuesta-Albertos, J.A., Matrán, C.: A fixed-point approach to barycenters in Wasserstein space. J. Math. Anal. Appl. **441**(2), 744–762 (2016). https://doi.org/10.1016/j.jmaa.2016.04.045
50. Bigot, J., Klein, T.: Characterization of barycenters in the Wasserstein space by averaging optimal transport maps. ESAIM Probab. Stat. **22**, 35–57 (2018). https://doi.org/10.1051/ps/2017020
51. Cazelles, E., Tobar, F., Fontbona, J.: A novel notion of barycenter for probability distributions based on optimal weak mass transport. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems, vol. 34, pp. 13575–13586. Curran Associates, Inc., virtual (2021). https://proceedings.neurips.cc/paper/2021/file/70d5212dd052b2ef06e5e562f6f9ab9c-Paper.pdf
52. Beck, A., Sabach, S.: Weiszfeld's method: old and new results. J. Optim. Theory Appl. **164**(1), 1–40 (2015). https://doi.org/10.1007/s10957-014-0586-7
53. Bajaj, C.: The algebraic degree of geometric optimization problems. Discrete Comput. Geom. **3**(2), 177–191 (1988). https://doi.org/10.1007/BF02187906

54. Santambrogio, F.: Optimal Transport for Applied Mathematicians. Progress in Nonlinear Differential Equations and their Applications, vol. 87, p. 353. Birkhäuser/Springer, Cham, Cham, CH (2015). https://doi.org/10.1007/978-3-319-20828-2. Calculus of variations, PDEs, and modeling
55. Flamary, R., Courty, N., Gramfort, A., Alaya, M.Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N.T.H., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D.J., Tavenard, R., Tong, A., Vayer, T.: Pot: Python optimal transport. J. Mach. Learn. Res. **22**(78), 1–8 (2021)
56. Ye, J., Wu, P., Wang, J.Z., Li, J.: Fast discrete distribution clustering using Wasserstein barycenter with sparse support. IEEE Trans. Signal Process. **65**(9), 2317–2332 (2017). https://doi.org/10.1109/TSP.2017.2659647
57. Lopuhaä, H.P., Rousseeuw, P.J.: Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. Ann. Statist. **19**(1), 229–248 (1991). https://doi.org/10.1214/aos/1176347978
58. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conf. on computer vision and pattern recognition (CVPR) (2014)
59. Welzl, E.: Smallest enclosing disks (balls and ellipsoids). In: new results and new trends in computer science (Graz, 1991). Lecture Notes in Comput. Sci., vol. 555, pp. 359–370. Springer, Graz, AT (1991). https://doi.org/10.1007/BFb0038202