Biological Imaging (2024), 4: e5
doi:10.1017/S2633903X24000059

RESEARCH ARTICLE

Alignment of density maps in Wasserstein distance

Amit Singer[1,2] and Ruiyi Yang[2]

[1]Department of Mathematics, Princeton University, Princeton, NJ, USA
[2]Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ, USA
Corresponding author: Ruiyi Yang; Email: ry8311@princeton.edu

Received: 12 October 2023; **Revised:** 19 February 2024; **Accepted:** 09 March 2024

Keywords: Bayesian optimization; cryogenic electron microscopy; heterogeneity analysis; volume alignment; Wasserstein distance

Abstract

In this article, we propose an algorithm for aligning three-dimensional objects when represented as density maps, motivated by applications in cryogenic electron microscopy. The algorithm is based on minimizing the 1-Wasserstein distance between the density maps after a rigid transformation. The induced loss function enjoys a more benign landscape than its Euclidean counterpart and Bayesian optimization is employed for computation. Numerical experiments show improved accuracy and efficiency over existing algorithms on the alignment of real protein molecules. In the context of aligning heterogeneous pairs, we illustrate a potential need for new distance functions.

Impact Statement

This article proposes a fast algorithm for aligning three-dimensional volumes represented as density maps with a particular focus on applications in cryogenic electron microscopy. The algorithm achieves both improved accuracy and efficiency over existing methods on the alignment of real protein molecules. The article also demonstrates a potential need for new distance functions for the alignment of heterogeneous pairs of volumes.

1. Introduction

Alignment of three-dimensional objects is an important task in applications ranging from computer vision and robotics such as shape registration and model retrieval[1–3] to medical imaging and molecular biology where protein structures need to be aligned before further processing and conformational analysis.[4–6] Given a pair of 3D objects which are rigid transformations of each other, the goal is to recover the relative translation and rotation that would match the two objects. As the alignment procedure often needs to be applied multiple times in the applications above, designing an accurate and efficient algorithm is of great significance.

In this article, we shall be interested in the case where the 3D objects are represented as density maps, motivated by applications in cryogenic electron microscopy (cryo-EM).[7] To formalize our setup, suppose $\phi_1, \phi_2 : \mathbb{R}^3 \to \mathbb{R}$ are two probability density functions representing the volumes, with ϕ_2 being a transformed version of ϕ_1, that is,

$$\phi_2(x) = \phi_1(R_*(x+v_*)), \qquad \forall x \in \mathbb{R}^3, \tag{1}$$

for some $v_* \in \mathbb{R}^3$ and $R_* \in \mathrm{SO}(3)$, the rotation matrix group. The goal of the alignment problem is to recover the rotation R_* and translation v_* based on the density maps ϕ_1 and ϕ_2. Here we have assumed ϕ_1 and ϕ_2 to be probability densities only for framing our problem in Wasserstein distances below, while the proposed algorithm will work for density maps taking negative values or having non-unit masses. In practice, the volumes are given as three-dimensional arrays $V_i \in \mathbb{R}^{L\times L\times L}$ with L an integer, which can be treated as discretizations of the ϕ_i's, with the voxel values encoding their configurations.

A natural idea for solving the alignment problem is to search for the optimal translation and rotation through the following optimization task:

$$\left(\hat{v},\hat{R}\right) \in \underset{(v,R)\in\mathbb{B}\times\mathrm{SO}(3)}{\arg\min}\ d(\phi_1(R(\cdot+v)),\phi_2(\cdot)) =: \underset{(v,R)\in\mathbb{B}\times\mathrm{SO}(3)}{\arg\min}\ F_d(v,R), \tag{2}$$

where $\mathbb{B}$ is a cube containing v_* and d is a suitable distance function on the space $\mathcal{P}\left(\mathbb{R}^3\right)$ of all probability measures over $\mathbb{R}^3$. Most existing works with few exceptions (see Section 2) set d as the usual L^2 distance and (2) is then solved by gradient-based methods or a type of exhaustive search over the space $\mathbb{B}\times\mathrm{SO}(3)$. However, due to the irregular shapes of the volumes, the landscape of F_{L^2} could be highly nonconvex and gradient-based methods would fail with poor initialization. Exhaustive search-based methods, on the other hand, could return more accurate results but have formidable costs if implemented naively. Methods exploiting convolution structures of F_{L^2}[8] can lead to great computational speed up but are still considered expensive for large volumes.

Motivated by these issues, in this article, we shall propose an alignment algorithm based on solving (2) in the 1-Wasserstein distance, which is known to better reflect rigid transformations than Euclidean distances and hence creates a better loss landscape. Exploiting this fact, we employ tools from Bayesian optimization for numerical minimization of (2), which is able to return global optimizers yet with much fewer evaluations of the objective than an exhaustive search. The resulting algorithm achieves improved performance over existing ones as we will demonstrate on the alignment of real protein molecules.

1.1. Wasserstein versus Euclidean landscapes

The main motivation for considering Wasserstein distances in (2) comes from the better-resulting loss landscape as we will discuss in this subsection. Recall that for two probability measures $\mu,\nu\in\mathcal{P}\left(\mathbb{R}^3\right)$, the p-Wasserstein distance for $p\in[1,\infty)$ is defined as

$$W_p(\mu,\nu) = \left(\inf_{\gamma\in\Gamma(\mu,\nu)}\int_{\mathbb{R}^3\times\mathbb{R}^3}|x-y|^p d\gamma(x,y)\right)^{1/p},$$

where $\Gamma(\mu,\nu)$ is the set of all couplings between μ and ν, that is, the set of all joint probability measures over $\mathbb{R}^3\times\mathbb{R}^3$ whose marginals are μ and ν. Wasserstein distances have been widely studied and employed in for instance image retrieval,[9] deep learning,[10] and structural determination of molecular conformation[11] among many other areas of applied sciences.

For the alignment problem that we are interested in, the Wasserstein distances are better able to reflect the distances between a density map and its transformed version. For instance, it is shown in Ref. (12), Lemma 3.5 that for $p\in(1,\infty)$,

$$W_p(\phi(\cdot),\phi(\cdot+v)) = |v|, \qquad \forall v\in\mathbb{R}^3,$$

where $\phi(\cdot+v)$ denotes the v-shifted density and $|\cdot|$ is the Euclidean norm. Therefore if the volumes are simply translations of each other (i.e., when R equals identity I_3 in (1)), then the associated loss in (2) satisfies $F_{W_p}(v,I_3)=|v|$, a convex function with a unique minimum at $v=0$. However, the same is far from being true for the L^p loss $\|\phi(\cdot)-\phi(\cdot+v)\|_p$ if ϕ has an irregular shape.

Similar assertions can be made when the two volumes are pure rotations of each other (i.e., when $v=0$ in (1)). Precisely, one can show that see, for example, Ref. (13), Propositions 1 and 2

$$W_p(\phi(R_\theta\cdot),\phi(\cdot)) \le 2\sin\left(\frac{\theta}{2}\right)M_p(\phi)^{1/p}$$

for an in-plane rotation R_θ of angle θ, where $M_p(\phi)$ is the p-th moment of ϕ. The corresponding bound for the L^p distance would have an additional factor of $\|\nabla\phi\|_\infty$, which could be large and gives a looser control on the change of L^p distance with respect to the magnitude of θ. Figure 1b,c plots the loss $F_d(0,R)$ for two

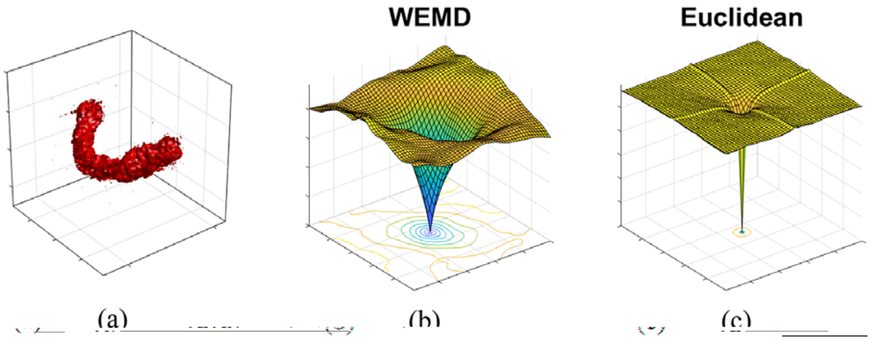

Figure 1. *(a) Visualization of the test volume. (b,c) Comparison of local landscapes of $F_d(0,R)$ when d is WEMD (cf. (14)) and Euclidean (L^2).*

distance functions when $R = R_\gamma \cdot R_\beta$ represents a rotation around the y-axis by $\beta \in [-\pi/2, \pi/2]$ followed by a rotation around the $z-$axis by $\gamma \in [-\pi/2, \pi/2]$, for the volume shown in Figure 1a. Here WEMD denotes the wavelet approximation of W_1 that we shall use for computation (see Section 3.3.1) and Euclidean stands for the usual L^2 distance between vectors. We notice that the landscape of the loss associated with WEMD is flatter or has a larger basin of attraction compared with that for the Euclidean distance, which can facilitate the search for the minimizer. The narrow basin of attraction in the Euclidean case suggests the necessity of some type of exhaustive search unless the initial guess for gradient-based methods happens to fall in such region.

Remark 1.1. Despite the improved landscape offered by Wasserstein distances, there are certain issues that could arise when aligning a heterogeneous pairs of volumes that we shall discuss in Section 5. In particular, Wasserstein distances could be unrobust to perturbations of the volumes, which motivates the need for new distance functions.

1.2. Computation via Bayesian optimization

The landscape analysis above suggests potential benefits in using Wasserstein distances as the loss function in solving (2). However, a question remains for its numerical optimization as Wasserstein distances are less analytically tractable and more computationally costly. In particular, computing gradients of (2) would be challenging both analytically and numerically, and an exhaustive search would require a huge computational budget due to the lack of a convolution structure. With these issues in mind, we shall instead adopt a Bayesian optimization approach, which does not require gradient information of the objective (2) while being able to return accurate solutions with much fewer evaluations of (2) than an exhaustive search.

First of all, let us make the following simple observation that the relative translation can be recovered by centering the two density maps, so that the problem reduces to estimating the rotation R_* alone. Indeed, suppose without loss of generality that ϕ_1 is already centered, that is, $\int_{\mathbb{R}^3} x\phi_1(x)dx = 0$. Then (1) implies after a change of variable that

$$\int_{\mathbb{R}^3} x\phi_2(x)dx = \int_{\mathbb{R}^3} x\phi_1(R_*(x+v_*))dx = R_*^{-1}\int_{\mathbb{R}^3} x\phi_1(x)dx - v_* = -v_*.$$

Therefore one can recover v_* by computing the center of mass of ϕ_2 and the shifted volume $\tilde{\phi}_2(x) := \phi_2(x - v_*) = \phi_1(R_* x)$ is then a purely rotated version of ϕ_1. This leads to a viable approach for estimating the shift vector between ϕ_1, ϕ_2 when they are noise free, as we shall demonstrate in Section 4.3. For the rest of this article, we shall mainly focus on the rotational recovery, that is, we assume

$$\phi_2(x) = \phi_1(R_* x), \qquad \forall x \in \mathbb{R}^3$$

and find the best rotation that minimizes

$$\widehat{R} \in \underset{R \in \mathrm{SO}(3)}{\arg\min}\, d(\phi_1(R(\cdot)), \phi_2(\cdot)) =: \underset{R \in \mathrm{SO}(3)}{\arg\min}\, F_d(R). \tag{3}$$

This will be achieved with Bayesian optimization as we overview next.

On a high level, Bayesian optimization is an iterative procedure that searches for optimizer candidates by solving a sequence of surrogate problems instead of the original one (3). At the t-th iteration, one collects the candidates $\{R_i\}_{i=1}^t$ picked so far together with the associated function values $\{F_d(R_i)\}_{i=1}^t$ to form using Bayesian techniques a surrogate function f_t whose landscape resembles F_d while being much cheaper to optimize. The $(t+1)$-th candidate is then chosen by solving the surrogate problem

$$R_{t+1} \in \underset{R \in \mathrm{SO}(3)}{\arg\min}\, f_t(R), \tag{4}$$

and is incorporated to the history $\{R_i, F_d(R_i)\}_{i=1}^t$ for updating f_t. After a total number of T iterations, the approximate solution to (3) is returned as

$$\widehat{R} \in \underset{t=1,\ldots,T}{\arg\min}\, F_d(R_t).$$

Note that the whole procedure only requires access to function evaluations of F_d but not its gradient.

Intuitively speaking, the algorithm explores the search space $\mathrm{SO}(3)$ based on the landscapes of the surrogate functions f_t's, which will approximate that of F_d as t increases but are much simpler to decode. As can be expected, the construction of f_t is crucial for the algorithm to perform well. In this article, we shall take f_t as a Gaussian process interpolant (see Section 3.2 for more details), which admits a simple analytic formula whose gradient is also available in closed form so that the surrogate problems (4) can be solved cheaply. Our numerical experiments in Section 4.2 show that a total of $T = 200$ iterations would suffice for accurate rotational alignment for real protein molecules, suggesting it as a practical algorithm. Furthermore, the fact that only evaluations of F_d are required implies that the proposed framework can be applied to arbitrary loss functions in (3), beyond vanilla Wasserstein or Euclidean distances. This could be a useful property when aligning a pair of heterogeneous volumes as we discuss in Section 5 where more sophisticated loss functions may be needed. The algorithm can also be extended beyond the density map assumption to volumes represented for instance by point clouds as long as one can define a suitable loss function as in (3).

1.3. *Our contributions*

The contributions of this article are summarized as follows:

- We propose a novel algorithm for aligning three-dimensional objects (Section 3), which achieves both improved accuracy and efficiency over existing methods on real datasets of protein molecules from cryo-EM (Section 4).
- Our algorithmic framework can be extended beyond the 1-Wasserstein distance to arbitrary distance functions. In the context of aligning a heterogeneous (similar but non-identical) pair of volumes, we show a potential need for new loss functions in which case our algorithm can be seamlessly incorporated (Section 5).

2. Related work

A classical approach for rotational alignment when the volumes are represented as point clouds is to perform principal component analysis and align the resulting eigenvectors. This method could give exact recovery in theory (up to reflections) but is not robust to perturbations of the given volumes, which is usually the case in practice, and breaks when the volumes admit certain symmetry. More recent works in the computer vision literature include Refs. (2,14,15) (see also the references therein), which again assume representations of the volumes as point clouds or other shape descriptors and parametrizations. Our problem setting is slightly different in that the point clouds representing our volumes are always fixed as the same Cartesian grid, where it is the voxel values that characterize their configurations. Nevertheless, we remark that the proposed algorithm in this article can be extended easily to the point cloud setting.

In the context of density map alignment that we consider, the optimization approach (2) is most often adopted, which is then solved by gradient-based or exhaustive search-based methods. Setting d as the L^2 distance, (2) is equivalent to correlation maximization

$$\left(\widehat{v},\widehat{R}\right) = \underset{(v,R)\in\mathbb{B}\times\mathrm{SO}(3)}{\arg\max} \ \langle \phi_1(R(\cdot+v)),\phi_2(\cdot)\rangle_{L^2}, \tag{5}$$

where $\langle\cdot,\cdot\rangle_{L^2}$ denotes the L^2 inner product. The Chimera package(16) implements a steepest ascent algorithm for solving (5) by relying on an initial alignment that is close to the true one, which is usually done manually by the user. Ref. (17) proposes using Kullback–Leibler divergence as the loss function which is later optimized with gradient descent, but no simulations are presented. Setting d as an entropic regularization of the 2-Wasserstein distance, the recent work(18) solves (2) with stochastic gradient descent by iteratively computing the optimal transport plans. Improved results are obtained over Chimera but the algorithm still requires the initial alignment to be within certain range of the true one. Extending such ideas, Ref. (19) uses Gromov-Wasserstein distance for partial alignment of density maps. These are the only works we are aware of that employ Wasserstein-based distances for the alignment problem. Our proposed method differs from Refs.(18,19) by employing Bayesian optimization and not involving explicit calculation of transport plans.

For this reason, exhaustive search-based methods are also popular and appear necessary. One subcategory of works in this direction attempts to search for the best alignment over a dense grid of translations and rotations that maximizes (5). Since the number of grid points is typically large, the key to these methods is a fast algorithm for computing (5) given a pair of v and R.(8,20,21) This can be achieved by expanding the correlation in spherical harmonic bases (with the efficient spherical Fourier transform(22)) and using the fact that rotation corresponds to application of Wigner-D matrices. Translations can be treated similarly with fast Fourier transform techniques. Existing packages include Xmipp,(23) and EMAN2(24) which employs a hierarchical tree-based algorithm.

Another subcategory of works approaches this problem by considering the projections of the volumes. The main idea is to not maximize the correlation between the volumes as in (5) but instead their projections, where only inner products between images are computed and the search space can be reduced to five-dimensional.(25) The recent work(26) improves this idea by employing common lines based techniques to further accelerate the search of matching projections.

3. The proposed method

In this section, we present in detail our proposed method to solve (3) with d as the 1-Wasserstein distance:

$$\widehat{R}\in \underset{R\in\mathrm{SO}(3)}{\arg\min}\, W_1(\phi_1(R(\cdot)),\phi_2(\cdot)) =: \underset{R\in\mathrm{SO}(3)}{\arg\min}\, F_{W_1}(R). \tag{6}$$

Our algorithm is summarized in Algorithm 1, which exploits the general framework of Bayesian optimization. To make our presentation self-contained, we shall introduce some necessary background before explicating the algorithmic details. We refer to Ref. (27) for a more thorough introduction to Bayesian optimization. The interested reader can skip to Section 3.3 for a description of our full algorithm.

As overviewed in Section 1.2, Bayesian optimization is an iterative procedure that searches for an optimizer of (6) by solving instead a sequence of surrogate problems. At each iteration, the surrogate problem is to optimize an *acquisition function* constructed based on a *probabilistic model* and all past queries of the objective function F_{W_1}. The acquisition function should be cheap to optimize while at the same time encodes enough information on the landscape of F_{W_1}. For our purpose in this article, we shall adopt a *Gaussian process* for probabilistic modeling of F_{W_1} (Section 3.1) and a *Gaussian process interpolant* for the acquisition function (Section 3.2).

3.1. Gaussian processes

The starting point of Bayesian optimization is to build a probabilistic model for the function F_{W_1} in (6) that we wish to optimize, which is used to construct an acquisition function as in Section 3.2. Since F_{W_1} is a nonparametric function, a common choice is to model it as a sample path from a Gaussian process.[28] Recall that a Gaussian process (GP) over a space $\mathcal{M}$ is a collection of random variables $\{u(x), x \in \mathcal{M}\}$ where any finite subcollection is jointly Gaussian. For any finite set $\{x_i\}_{i=1}^N \subset \mathcal{M}$, the mean vector $m \in \mathbb{R}^n$ and the covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ of the finite-dimensional Gaussian $[u(x_1), \ldots, u(x_n)]^T$ can be specified in a consistent way through a mean function $\mu(\cdot)$ and a covariance function $c(\cdot,\cdot)$ so that $m_i = \mu(x_i)$ and $\Sigma_{ij} = c(x_i, x_j)$. Intuitively speaking, a GP is a random process whose realizations fluctuate around $\mu(\cdot)$ according to restrictions imposed by $c(\cdot,\cdot)$. In particular, $\mu(\cdot)$ and $c(\cdot,\cdot)$ completely determine the distribution of the GP and from the modeling perspective it suffices to make the appropriate choices for them.

Typically the mean is set to be zero and the covariance function encodes one's prior belief on sample path properties such as smoothness. When the space $\mathcal{M}$ is a subset of the Euclidean space, one of the most commonly used covariance functions is the squared exponential

$$c(x,y) = \sigma^2 \exp\left(-\frac{|x-y|^2}{2\ell^2}\right), \quad x, y \in \mathbb{R}^n, \tag{7}$$

where $\sigma, \ell > 0$ are respectively the marginal variance and correlation lengthscale parameters. Roughly speaking, σ determines the overall magnitude of the sample paths, and function values at two points with distance on the order of ℓ are nearly uncorrelated. An important feature of the squared exponential covariance function is that it leads to infinitely differentiable sample paths see, for example, Ref. (28) Section 4.2, which are suitable choices for modeling smooth functions.

Based on our discussion in Section 1.1 that the Wasserstein distances vary relatively smoothly with respect to rigid transformations, the squared exponential serves as a natural choice for our problem. Recall that in our setting the space $\mathcal{M} = \mathrm{SO}(3)$ is a subset of $\mathbb{R}^{3 \times 3}$. Therefore we shall define our covariance function as

$$c(R,S) = \sigma^2 \exp\left(-\frac{\|R-S\|_F^2}{2\ell^2}\right), \quad R, S \in \mathrm{SO}(3), \tag{8}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Notice that this is equivalent to viewing each matrix as a vector in $\mathbb{R}^9$ and applying (7), as a result of which the covariance function (8) retains positive definiteness. Therefore the probabilistic model for F_{W_1} in (6) that we shall adopt is a GP over $\mathrm{SO}(3)$ with mean zero and covariance (8).

Remark 3.1. Here we briefly discuss other possible choices of covariance functions over $\mathrm{SO}(3)$. On the one hand, any other Euclidean covariance functions such as the Matérn family see, for example, Ref. (28), Section 4.2 can be employed by viewing $\mathrm{SO}(3)$ as a subset of $\mathbb{R}^{3 \times 3}$. Our choice of (8) is motivated by its superior empirical performance in our alignment problem and its simple form that facilitates numerical optimization of the acquisition function as will be discussed in Section 3.2.

On the other hand, the space $\mathrm{SO}(3)$ admits a manifold structure, which suggests a covariance function over the manifold $\mathrm{SO}(3)$ that takes geometry into account. This for instance can be achieved by

$$c(R,S) = \sigma^2 \exp\left(-\frac{\Theta(R,S)^2}{2\ell^2}\right), \quad \Theta(R,S) = \arccos\left(\frac{\mathrm{Trace}(RS^T) - 1}{2}\right), \quad R, S \in \mathrm{SO}(3), \tag{9}$$

where $\Theta(R,S)$ is the relative angle between the two rotations, whose absolute value is also the geodesic distance on $\mathrm{SO}(3)$. However, *geodesic exponential kernels* such as (9) are not positive definite in general for all $\ell > 0$, although empirical evidence suggests positive definiteness for a range of ℓ's in certain cases.[29] Moreover, our simulation experience suggests slightly better accuracy when using (8) over (9) so we shall stick to the covariance function (8). Finally, we point to some more sophisticated covariance functions over SO(3) proposed by Ref. (30).

3.2. Gaussian process interpolant as surrogate

With a probabilistic model for F_{W_1}, we shall replace the original optimization problem (6) by a sequence of simpler surrogate problems. This will be achieved by constructing simple-to-optimize approximations f_t to the objective function F_{W_1} in (6). In this article, we shall take f_t as the conditional expectation of the GP proposed in Section 3.1 after "observing" the data $\{(R_i, F_{W_1}(R_i))\}_{i=1}^t$, which we explain now.

Suppose we have picked the first t candidates $\{R_i\}_{i=1}^t$ (for the initial candidates, we can for instance generate t_0 random rotation matrices). Together with the associated objective function values $Y_i = F_{W_1}(R_i)$, we shall interpret the pairs $\{(R_i, Y_i)\}_{i=1}^t$ as observations we have obtained for the unknown function F_{W_1}. Now in the Bayesian regression framework, we have

$$F_{W_1} \sim \Pi, \quad F_{W_1}(R_i) = Y_i,$$

where Π is the GP model as in Section 3.1. Therefore a natural estimator for F_{W_1} is the conditional expectation

$$f_t(x) = \mathbb{E}_{G\sim\Pi}[G(x) \mid G(R_i) = Y_i, 1 \le i \le t],$$

which is known[31] to minimize the squared error loss $\mathbb{E}_{F_{W_1}\sim\Pi}\left|F_{W_1}(x) - \widehat{F}(x)\right|^2$ over all $\widehat{F}(x)$ that is measurable with respect to $\{F_{W_1}(R_i)\}_{i=1}^t$ when F_{W_1} is indeed a sample path from Π. In particular, f_t interpolates F_{W_1}, that is, $f_t(R_i) = F_{W_1}(R_i)$ for $1 \le i \le t$, and approximates F_{W_1} increasingly well as more observations of F_{W_1} are obtained. Furthermore, it can be shown that, for example, Ref. (32), Theorem 3.3 f_t admits a simple analytic formula

$$f_t(x) = k(x)^T K^{-1} Y, \quad x \in \mathrm{SO}(3), \tag{10}$$

where $k(x) \in \mathbb{R}^t$ is a vector with entries $[k(x)]_i = c(x, R_i)$, $K \in \mathbb{R}^{t\times t}$ is a matrix with entries $K_{ij} = c(R_i, R_j)$, and $Y \in \mathbb{R}^t$ is a vector with entries $Y_i = F_{W_1}(R_i)$. Notice that f_t is simply a linear combination of the covariance functions

$$f_t(x) = \sum_{i=1}^t \left[K^{-1}Y\right]_i c(x, R_i), \quad x \in \mathrm{SO}(3),$$

and admits an analytic formula for its Euclidean gradient under the covariance function choice (8) as

$$\nabla^{\mathrm{Eu}} f_t(x) = \sum_{i=1}^t \left[K^{-1}Y\right]_i c(x, R_i)\left(\frac{R_i - x}{\ell^2}\right), \quad x \in \mathrm{SO}(3). \tag{11}$$

Therefore optimization of f_t can be carried out much more cheaply compared with the original problem (6) by supplying the gradient to standard optimization packages.

3.3. The full algorithm

Now we are ready to present the full algorithm, as summarized in Algorithm 1. Starting with a GP model for F_{W_1} and initial candidates $\{R_i\}_{i=1}^{t_0}$ (which can be randomly generated), we form the surrogate function (10) based on all observations $\{(R_i, F_{W_1}(R_i))\}_{i=1}^t$ obtained so far and search for the next candidate by solving

$$R_{t+1} \in \underset{R\in\mathrm{SO}(3)}{\arg\min} f_t(R). \tag{12}$$

Now with the new observation $(R_{t+1}, F_{W_1}(R_{t+1}))$ included, we update the surrogate to f_{t+1} and repeat the process (12). After a total of T iterations, we shall return the candidate with smallest objective function value, that is,

$$R_{\text{est}} \in \arg\min_{t=1,\ldots,T} F_{W_1}(R_t) \tag{13}$$

as the solution to the original problem (2). Such a procedure would serve as a prototype for our proposed method. To further improve its practicality, below we shall introduce two modifications: (i) approximation of the W_1 distance and (ii) an optional local refinement step.

3.3.1. Wavelet approximation of W_1 distance

Despite the favorable induced landscape as shown in Section 1.1, a notable issue for Wasserstein distances is their high computational cost. For practical applications in cryo-EM that we shall consider, the density maps are represented as three-dimensional arrays of size $L \times L \times L$ with L on the order of hundreds. Therefore the cost of exact computation of W_1 distances is prohibitive and scales in the worst case as $O\left(N^3 \log N\right)$ with $N = L^3$. For this reason, we seek an approximation of W_1 through the wavelet approximation proposed by Ref. (33) which reduces the above cost to $O(N)$.

Precisely, the wavelet earth mover's distance (WEMD)(33) is defined as

$$\|\phi_1 - \phi_2\|_{\text{WEMD}} = \sum_{\lambda} 2^{-j(1+n/2)} |\mathcal{W}\phi_1(\lambda) - \mathcal{W}\phi_2(\lambda)|, \tag{14}$$

where $n = 3$ denotes the dimension of the density maps and $\mathcal{W}\phi_i$ denotes a 3D wavelet transform. The index λ consists of the triplet (ϵ, j, k), where ϵ takes values in a finite set of size $2^n - 1$ that for instance represents tensor products of 1D wavelets, and j is the scale parameter that ranges over $\in \mathbb{Z}_{\geq 0}$ and k ranges over $\in \mathbb{Z}^n$. It is proved in Ref. (33), Theorem 2 that the metric defined above is equivalent to the W_1 distance. Such an approximation has the additional advantage that the distance (14) can be defined for density maps that take negative values, which is usually the case in practice, whereas the original W_1 distance is restricted to probability densities.

Now we shall replace all occurrences of the W_1 distance in our previous algorithmic procedure by the WEMD distance (14). Precisely, the alignment problem we shall be solving becomes

$$\widehat{R} \in \arg\min_{R \in \mathrm{SO}(3)} \|\phi_1(R(\cdot)) - \phi_2(\cdot)\|_{\text{WEMD}} =: \arg\min_{R \in \mathrm{SO}(3)} F_{\text{WEMD}}(R), \tag{15}$$

and the surrogate problem (12) together with the returned solution (13) will be defined in terms of F_{WEMD} instead. Note that the Bayesian optimization framework only requires access to function evaluations, so that replacing F_{W_1} by F_{WEMD} does not require extra modifications of the algorithm. In principle, F_{WEMD} can be further replaced by any other distance function, which is an important feature of the adopted framework that we shall elaborate more in Remark 3.4.

3.3.2. Local refinement

In Section 4.2, we will show numerically that after $T = 200$ iterations, the algorithm described so far returns reasonably accurate recovery of the relative rotation R_* for real protein molecules. However, for a finite t, the surrogate f_t only approximates F_{W_1} and so does its minimizer. In order to obtain close-to-exact recovery, the candidates should form a dense enough cover of $\mathrm{SO}(3)$, which would require many more samples than $T = 200$ and is computationally infeasible. For this reason, we introduce an optional local refinement step by employing the Nelder–Mead algorithm. Precisely, we shall return

$$R_{\text{refine}} \in \arg\min_{R \in \mathrm{SO}(3)} \|\phi_1(R(\cdot)) - \phi_2(\cdot))\|_2, \tag{16}$$

where (16) is optimized with the Nelder–Mead algorithm initialized at R_{est} given by (13). This concludes the description of our algorithm, presented in Algorithm 1.

Note that we switched to the L^2 loss in (16) for reasons to be explained in Remark 3.5. We further remark that (16) can be solved in principle by other standard optimization algorithms such as BFGS. We have deliberately chosen Nelder–Mead because it only requires loss function evaluations in a similar spirit as Bayesian optimization. Such a property can be useful and crucial in the context of aligning heterogeneous pairs of volumes, where we show in Section 5 a potential need for new and more sophisticated distance functions which one may only know how to evaluate. In this case, our proposed framework would still be applicable.

Algorithm 1. Volume Alignment in WEMD via Bayesian Optimization

```
Input: Volumes $\phi_1, \phi_2$; loss $F_{\mathrm{WEMD}}$ (15); GP covariance (8); initialization $\{(R_i, F_{\mathrm{WEMD}}(R_i))\}_{i=1}^{t_0}$.
    for $t = t_0, \dots, T$ do
    Compute $f_t$ as in (10) and find
        $$R_{t+1} \in \arg\min_{R \in \mathrm{SO}(3)} f_t(R)$$
    Add $(R_{t+1}, F_{\mathrm{WEMD}}(R_{t+1}))$ to $\{(R_i, F_{\mathrm{WEMD}}(R_i))\}_{i=1}^{t}$.
    end for
    Set the estimated rotation as
        $$R_{\mathrm{est}} \in \arg\min_{t=1,\dots,T} F_{\mathrm{WEMD}}(R_t).$$
    (Optional) Solve the following with Nelder–Mead algorithm initialized at $R_{\mathrm{est}}$
        $$R_{\mathrm{refine}} \in \arg\min_{R \in \mathrm{SO}(3)} \|\phi_1(R(\cdot)) - \phi_2(\cdot)\|_2.$$
Ouput: $R_{\mathrm{est}}$ and (optional) $R_{\mathrm{refine}}$.
```

We end this section with further remarks on the algorithmic details.

Remark 3.2 *(Choice of f_t). In Bayesian optimization, f_t is called the acquisition function and there have been extensive research on its choice see, for example, Ref. (27), all of which could have been employed in our problem setting. Our choice of (10) is motivated by its simple form that admits an analytic formula for its gradient, which facilitates solving (12). For readers familiar with Bayesian optimization, (10) corresponds to GP-UCB*[34] *with no exploration, which appears to be a suboptimal choice. However for practical implementation, one may only want to solve (12) approximately with early stopping to speed up the search, which can be treated as another form of exploration. Our experience suggests better performance for using (10) than adding a term proportional to the conditional variance as in the standard GP-UCB. Furthermore, (10) is independent of σ, the marginal variance parameter in (8), which would otherwise be present in the standard form and requires tuning.*

Remark 3.3 *(Numerical optimization of f_t and addressing handedness). The majority of the efforts in Algorithm 1 are devoted to solving the surrogate problems (12), whose accuracy and efficiency are the key to the algorithm. We note that the kernel matrix K in (10) could be ill-conditioned if two candidates R_t and $R_{t'}$ are very close to each other. For numerical stability, a nugget term is usually incorporated to (10) so that we consider instead*

$$f_t = k(x)^T (K + \tau I_t)^{-1} Y, \tag{17}$$

with a small τ for practical implementation.

Taking advantage of the manifold structure of $\mathrm{SO}(3)$, we shall optimize f_t with the Riemannian optimization package.[35,36] The Euclidean gradient (11) can still be supplied for speed up, which is automatically transformed into Riemannian gradients by the package. In cryo-EM applications, one often needs to address also the handedness of the molecules, that is, when ϕ_1 and ϕ_2 differ additionally by a reflection. This can be achieved in our framework by optimizing f_t instead over $\mathrm{O}(3)$, the space of orthogonal matrices, which corresponds to the Stiefel manifold $\mathrm{St}(3,3)$ in the package.[35,36] However, we remark that this is not much different from reflecting one of ϕ_1, ϕ_2 first and aligning the resulting pair since the new search space $\mathrm{O}(3)$ is twice as large as $\mathrm{SO}(3)$ and the number of iterations T for accurate alignment is also expected to double.

Remark 3.4 *(Choice of loss function). Algorithm 1* is presented in terms of the loss function F_{WEMD} (15) and only requires access to its evaluations but nothing else. An immediate observation is that Algorithm 1 can be applied to solve the alignment problem with any distance function d in (3), as a consequence of the Bayesian optimization framework. Our choice of W_1 distance is motivated by the fact that it can be efficiently approximated with WEMD (14). Based on the discussion in Section 1.1, we believe a general W_p distance could also be used, with for instance an entropic regularization[37] for practical implementation.

To illustrate the advantage of employing Wasserstein-based loss functions, we will show in Section 4.2 the improved performance of Algorithm 1 over its L^2 loss counterpart. Finally, in the context of aligning a pair of heterogeneous volumes in Section 5, we shall discuss a potential need for new loss functions other than the vanilla Wasserstein or Euclidean distances. In such a setting, the new loss function could be analytically intractable and expensive to evaluate, which renders unclear the applicability of gradient-based or exhaustive search-based algorithms. Nevertheless, Algorithm 1 could be seamlessly incorporated as long as we can afford a small number of evaluations of the loss function and potentially give a feasible solution.

Remark 3.5 *(Refinement inL^2distance). We have switched to theL^2distance in the refinement step (16) for the following two reasons. First, the recovery$R_{\mathrm{est}}$is already very close to the global minimizer of (15), which is likely to lie also in the basin of attraction in theL^2loss, although the latter is generally much narrower as shown in Figure 1. Therefore a local search inL^2loss would be sufficient and more efficient than in the WEMD.*

Second, the WEMD or W_1 distance appear to be more sensitive than the L^2 distance to perturbations of the density maps in terms of the optimal alignment. In practice, the density maps may correspond to two different reconstructions of the same object and are only provided as three-dimensional arrays $V_1, V_2 \in \mathbb{R}^{L\times L\times L}$ for an integer L. Therefore, the minimizers of the empirical versions of (15) and (16) are not necessarily equal to the true relative rotation R_* but only approximately. However, we found that the minimizer of the empirical version of (15) could be non-negligibly different from R_* in many cases where there is no issue with the L^2 distance, suggesting the L^2 loss as a more robust local search metric. Such observation is also related to the alignment of heterogeneous pairs that we shall discuss in Section 5.

4. Numerical experiments

In this section, we apply our proposed method in Section 3 to align real protein molecules from a cryo-EM database.[7] Section 4.1 contains the implementation details, in particular the choices of hyperparameters in Algorithm 1. In Section 4.2, we investigate the performance of Algorithm 1 under different down-sampling levels, total number of iterations, and noise corruption, while comparing with the L^2 loss version of Algorithm 1. In Section 4.3, we compare the performance of Algorithm 1 with the two recent works.[18,26] The algorithmic complexity is discussed in Section 4.4. Our code is available at https://github.com/RuiyiYang/BOTalign.

4.1. Implementation details

For GP modeling, we shall use the Gaussian kernel defined in (8) with $\sigma = 1$. Notice that the surrogate f_t defined in (10) is independent of σ so its choice is indeed arbitrary. The choice of ℓ, on the other hand, would have an effect and is empirically tuned for optimal algorithmic performance. As mentioned, we shall investigate the performance of Algorithm 1 under both the WEMD loss and the L^2 loss. The values of ℓ are fixed as 0.75 and 1 respectively throughout the experiments. The WEMD distance is computed using PyWavelet[38] with the `sym3` wavelet and maximum scale level $s = 6$ following.[39]

We shall initialize Algorithm 1 with a single candidate I_3, the identity matrix. Our experience suggests that the initialization does not affect much the performance. For optimization of the surrogate problems, we shall follow the discussion in Remark 3.2 and consider f_t defined in (17) with $\tau = 10^{-3}$ for numerical stability. The surrogate f_t is then optimized with Riemannian steepest descent with random initialization using Pymanopt,[36] with an early stopping if both the gradient norm and the step size are less than 0.1. We empirically found that such an early stopping greatly improves the efficiency of Algorithm 1 while not losing much accuracy (see Remark 3.2).

In practice, the density maps of the volumes are given as three-dimensional arrays $V \in \mathbb{R}^{L \times L \times L}$ for some integer L. In other words, V is supported on the Cartesian grid and computing its rotated versions is a nontrivial and essential procedure. In our algorithm, this is done with the ASPIRE package,[40] which first computes the nonuniform Fourier transform of V over the rotated grid and then applies an inverse Fourier transform. Note that this step is needed when computing the loss function values in Algorithm 1.

Finally, to further speed up the computation, a common practice in cryo-EM is to downsample the given volumes $V \in \mathbb{R}^{L \times L \times L}$ to be of size $\mathbb{R}^{L_0 \times L_0 \times L_0}$ for some integer $L_0 < L$. This leads to faster computation of WEMD distances and potentially a better loss landscape by removing the fine-scale structures of the protein molecules. For the rest of this section, we shall treat L_0 and the total number of iterations T in Algorithm 1 as user-chosen parameters and in Section 4.2 demonstrate its performance when $L_0 \in \{32, 64\}$ and $T \in \{150, 200\}$.

4.2. Alignment of real protein molecules

In this section, we shall illustrate the performance of Algorithm 1 on real protein molecules from the publicly available Electron Microscopy Data Bank.[7] The experimental setup is as follows. For a given volume $V_1 \in \mathbb{R}^{L \times L \times L}$, we randomly generate a rotation matrix $R_* \in \mathrm{SO}(3)$ and compute its rotated version $V_2 \in \mathbb{R}^{L \times L \times L}$ using the Fourier transform-based approach mentioned above. The goal is then to recover the rotation R_* given only V_1 and V_2. In this subsection we focus on pure rotation recovery and incorporation of translation will be demonstrated in Section 4.3. The test volumes shown in Figure 2 will be used throughout the numerical experiments.

4.2.1. Algorithm 1 without refinement

As mentioned, we shall first downsample the given volumes to $V_i^{\mathrm{DS}} \in \mathbb{R}^{L_0 \times L_0 \times L_0}$ and then align the V_i^{DS}'s with Algorithm 1. The downsampling level L_0 and the total number of iterations T in Algorithm 1 are treated as user-chosen parameters which could vary depending on the molecules at hand. Below we shall investigate their effects on the performance of Algorithm 1, first without the refinement step to focus on the Bayesian optimization performance.

Denoting the estimated rotation by R_{est}, we quantify the performance by the relative angle $|\Theta(R_*, R_{\mathrm{est}})|$ between R_* and R_{est} defined in (9). Figure 3 shows the results for four combinations of L_0 and T for the molecules shown in Figure 2. To illustrate the benefits of alignment in Wasserstein distances, also shown in Figure 3 are results for the parallel experiments with WEMD loss in Algorithm 1 replaced by the L^2 loss. The experiments are repeated 50 times with R_* regenerated in each. The run time is recorded on a laptop with Intel(R) Core(TM) i7–7500 CPU@ 2.70GHz. We see that with high probability, the WEMD version of Algorithm 1 is able to recover the relative rotation up to a 5-degree error with only 200 evaluations of the loss function and in most cases outperforms its L^2 counterpart with comparable computing time.

4.2.2. Algorithm 1 with refinement

With the results shown in Figure 3, we shall continue to demonstrate the performance of Algorithm 1 when refinement is included. We recall that the refinement step is a local search (16) solved with the Nelder–Mead algorithm initialized at the estimate R_{est} returned by the first part of Algorithm 1. We note that here we have the freedom to choose the combination of L_0 and T for obtaining R_{est} with optimal efficiency. In particular, setting $L_0 = 64$ and $T = 200$ as shown in Figure 3 leads to the best overall performance, but the other choices also give reasonably accurate recovery while requiring much less run

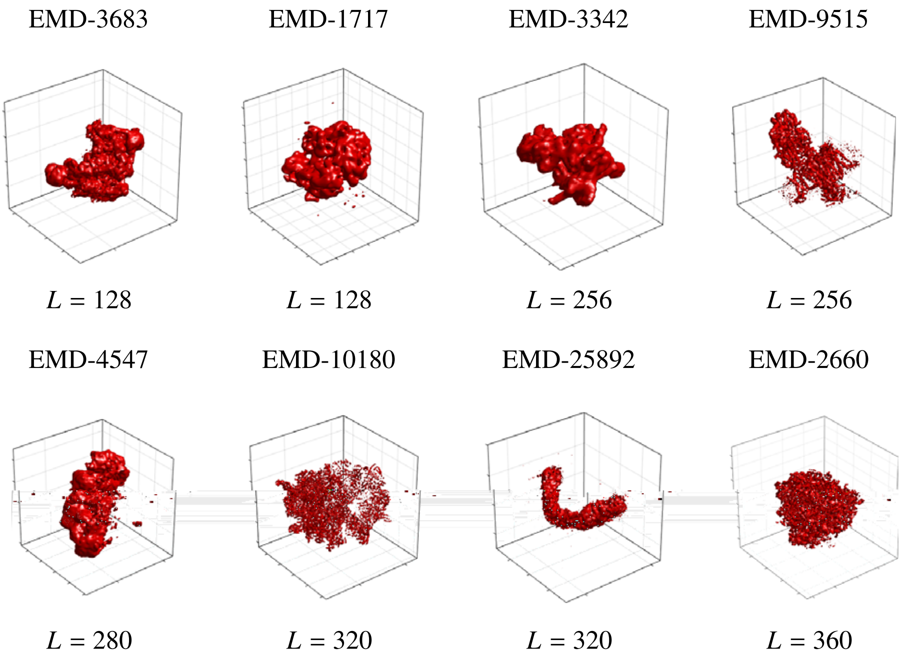

Figure 2. *Visualization of the test volumes.*

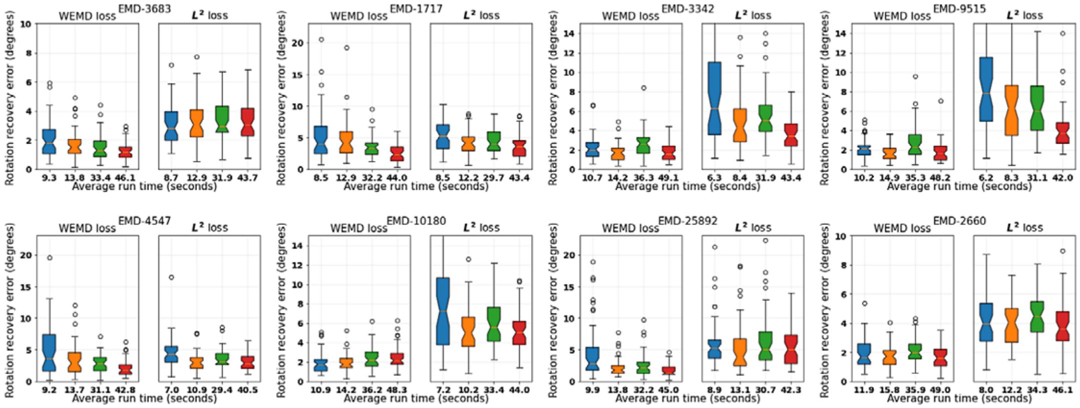

Figure 3. *Performance comparison between Algorithm 1 and its L^2 loss version without refinement. The four boxplots in each subfigure correspond to (from left to right) $(L_0, T) = (32,150), (32,200), (64,150), (64,200)$. The vertical axis represents rotation recovery error $|\Theta(R_*, R_{\text{est}})|$ in degrees. The tick labels record the average run time in seconds.*

time. Meanwhile, it would not be too surprising that the R_{est}'s returned by these other choices could lie in the basin of attraction around R_* so that local convergence is still retained after the refinement.

In Figure 4, we show that this is indeed the case for the combinations $(L_0, T) = (32,200)$ and $(64,150)$ for both the WEMD and L^2 versions of Algorithm 1, which give after refinement more accurate recovery but with less run time than using the combination $(64,200)$ without refinement. Here in the refinement step, we are fixing the downsampling level to be 32. In particular, we see that the local search step by Nelder–Mead appears to be not very stringent on its initializations so that even those returned by the L^2 version of Algorithm 1 would suffice for good final accuracy. However, this could be a coincidence due to the benign structures of the test volumes. The better initial estimates returned by the WEMD version as

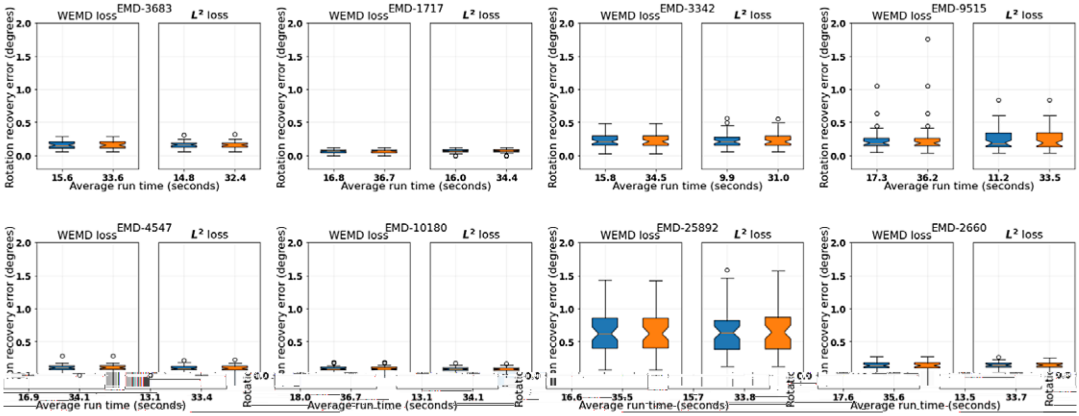

Figure 4. *Performance comparison between Algorithm 1 and its L^2 loss version with refinement. The two boxplots in each subfigure correspond to (from left to right)* $(L_0, T) = (32{,}200)\text{and}(64{,}150)$. *The vertical axis represents rotation recovery error* $|\Theta(R_*, R_{\text{est}})|$ *in degrees. The tick labels record the average run time in seconds.*

shown in Figure 3 could already be returned as a solution and at the same time are more reassuring as initializations for the local refinement. For this reason, the WEMD version serves as our main algorithm.

4.2.3. Robustness to noise

We further test the performance of Algorithm 1 in the presence of noise, where we fix the test volume to be EMD-3683 and add to each entry of V_1, V_2 an independent Gaussian noise of variance σ^2 across a range of signal-to-noise ratios, defined as SNR$=\|V_1\|_2^2/\left(L^3\sigma^2\right)$. Figure 5 visualizes a central slice of the noise-corrupted volumes. The performance of Algorithm 1 is shown in Figure 6, which shows a decent level of robustness. An interesting observation is that the L^2 version with downsampling level $L_0 = 32$ appears to be more robust than the WEMD version for very high noise levels. This is related to our discussion in Section 5 on the alignment of heterogeneous pairs since the noise corrupted volumes can be treated as different conformations of the clean one. We will show by an example that the Wasserstein-based distances could be more susceptible to heterogeneity.

4.3. Comparison with existing algorithms

Finally, we shall compare Algorithm 1 with two recent alignment algorithms proposed by Ref. (26), which exploits common line-based methods for fast projection matching, and Ref. (18), which considers an entropic regularization of 2-Wasserstein loss in (6) and employs stochastic gradient descent for optimization. In the following comparison, we shall also consider translation recovery. More precisely, given a volume $V_1 \in \mathbb{R}^{L\times L\times L}$, we randomly generate a rotation matrix R_* and first compute its rotated version V_2^{rot}, whose shifted version V_2 is then treated as the given volume. Here the shift vector is uniformly randomly generated over the cube $[-S, S]^3$ with $S = 0.05L$. This corresponds to a typical situation in cryo-EM applications where the given volumes are already preprocessed and approximately centered.

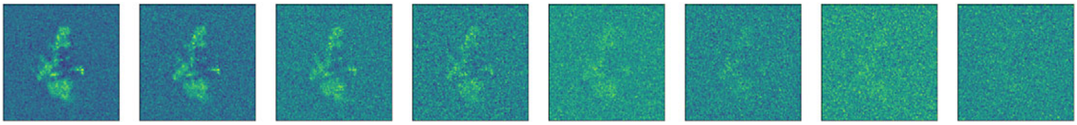

Figure 5. *Visualization of a central slice of EMD-3683 under different signal-to-noise ratios.*

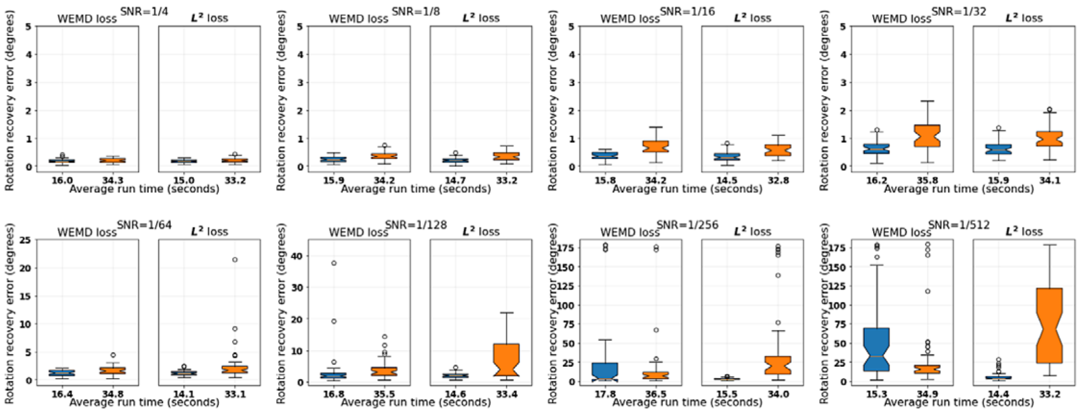

Figure 6. *Performance comparison between Algorithm 1 and its L^2 loss version with refinement under noise corruption. The two boxplots in each subfigure correspond to (from left to right)* $(L_0, T) = (32{,}200)$ *and* $(64{,}150)$*. The vertical axis represents rotation recovery error* $|\Theta(R_*, R_{\text{est}})$ *in degrees. The tick labels record the average run time in seconds.*

As mentioned in the introduction, we shall recover the shift by centering the volumes. We point out that the volumes given in the database[7] contain negative values so a thresholding is applied first before computing the center of mass, where the threshold is chosen based on the recommended contour level for each molecule in Ref. (7) or could be tuned empirically. The code provided by Ref. (18) also focuses only on rotation recovery so we apply the same centering step for their algorithm.

For the comparisons below, we shall use $(L_0, T) = (32{,}200)$ with refinement in our Algorithm 1, which will be denoted as Bayesian Optimal Transport Align (BOTalign). The algorithm in Ref. (26) will be denoted as EMalign following the authors, and is applied with downsampling 32 and their recommended number of reference projections 30, also implemented with their local refinement. Lastly, AlignOT stands for the algorithm in Ref. (18) with $n = 500$ in their topology representing network step and maximum number of iteration $N = 500$ in their stochastic gradient descent. Again, we repeat the experiments 50 times for each molecule and the results are shown in Figure 7 and Table 1.

We see that our algorithm achieves the best accuracy with minimal run time. We remark that AlignOT is a local search algorithm where the authors report good recovery if the angle between R_* and the identity

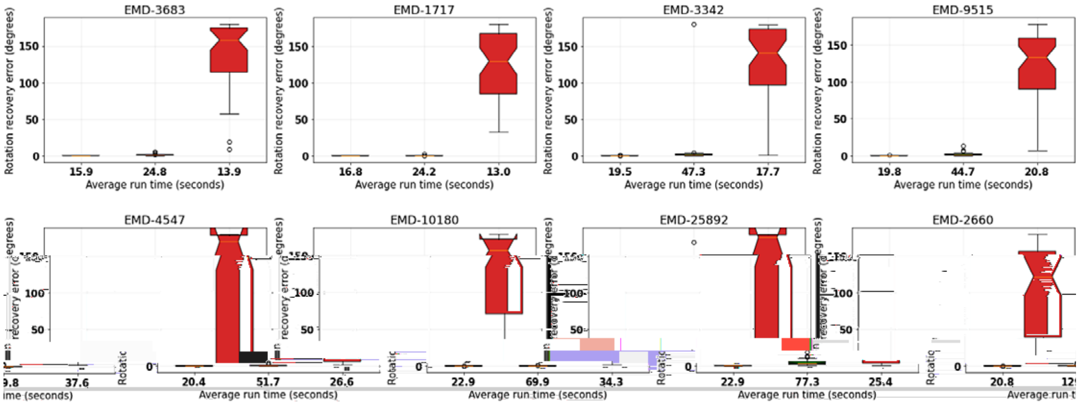

Figure 7. *Comparison with existing methods. The three boxplots in each subfigure correspond to (from left to right) BOTalign (our method), EMalign, and AlignOT. The vertical axis represents rotation recovery error* $|\Theta(R_*, R_{\text{est}})|$ *in degrees. The tick labels record the average run time in seconds.*

Table 1. Summary statistics of the recovery errors in Figure 7

EMD ID	3683	1717	3342	9515	4547	10180	25892	2660
BOTalign	0.26 (0.13)	0.11 (0.05)	0.23 (0.11)	0.25 (0.14)	0.11 (0.05)	0.10 (0.04)	0.58 (0.29)	0.18 (0.08)
EMalign	0.95 (1.09)	0.15 (0.31)	5.18 (24.98)	2.04 (2.23)	0.74 (0.77)	0.24 (0.18)	8.09 (23.34)	0.60 (0.60)
AlignOT	140.73 (43.33)	122.12 (45.87)	123.53 (56.28)	119.26 (49.13)	97.92 (85.39)	118.00 (69.14)	126.69 (77.90)	103.96 (61.38)

Note: In each block are the mean and standard deviation (in parentheses).

is within 75 degrees. This is not contradictory with the results shown in Figure 7 as here R_* is randomly generated, which more than half of the time is 75 degrees away from the identity. EMalign is a global search algorithm that improves over earlier alignment algorithms especially in terms of running time (see their Tables 2 and 3). Our algorithm achieves further improvements when aligning clean molecules. We mention that EMalign performs exhaustive search over shifts as well as rotations and could have an advantage when the noise level is high as the centering step would be less accurate. In this case, more sophisticated center of mass estimation method such as Ref. (41) needs to be employed instead in our approach.

4.4. *Algorithmic complexity*

Here we briefly discuss the complexity of our Algorithm 1 with respect to the size L of the volumes. The majority of the computational cost of our algorithm takes place in (i) solving the surrogate problem (12) and (ii) evaluating the loss F_{WEMD}. The former step is always a three-dimensional optimization problem and does not depend explicitly on L. Our empirical experience suggests that with the early stopping that we have adopted its cost is relatively small compared to evaluating F_{WEMD}. The latter would involve two steps, where one first rotates one of the volumes using nonuniform fast Fourier transform that costs $O\left(L^3 \log L\right)$, and then computes the WEMD with a cost of $O\left(L^3\right)$. One may need to use a larger maximum scale level s for large L's but the dependence is in general only logarithmic. Therefore the total cost of Algorithm 1 is on the order of $O\left(TL^3 \log L\right)$ where T is the total number of iterations which can be fixed for instance as 200. Therefore the dependence on T improves over naive exhaustive search methods which cost $O\left(|S| \times L^3\right)$ where $|S| = O\left(10^5\right) \sim O\left(10^6\right)$ is the size of the rotation grid to search over, and the dependence on L improves over the convolution based methods such as Refs. (8,20) which would take $O\left(L^4\right)$.

5. Alignment of heterogeneous pairs

In this section, we shall discuss the problem of aligning a pair of similar but non-identical volumes, which we denote as a heterogeneous pair. Such problem arises naturally in cryo-EM, where the same protein molecule can exhibit different conformations. For example, a molecule can consist of two moving parts that are rotating with respect to each other, or the molecule can be more elongated in certain states than others. We shall show by an example that alignment in W_1 distance as we have proposed could be problematic in the presence of heterogeneity. The same issue is present for the L^2 distance although milder. This motivates a need of new loss functions for aligning a heterogeneous pair, which ideally extracts and compares the common part of volumes. Such sophisticated loss functions are likely to render gradient-based or exhaustive search-based optimization ineffective, whereas our algorithmic framework in Algorithm 1 could be seamlessly incorporated.

To start with, let us consider a pair of simulated volumes V_1 and V_2 in Figure 8, which have an overall similar shape but certain differences in their upper right parts, representing different conformations of the

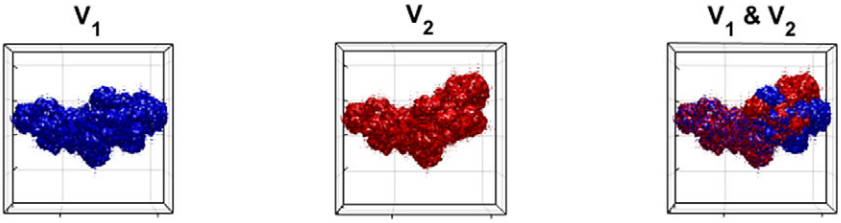

Figure 8. *Side views of a heterogeneous pair of volumes. Notice that V_1 and V_2 differ mostly in the upper right portion.*

same molecule. The two volumes are considered as already aligned as the "base" portions of the volumes are matched. However, if we apply Algorithm 1 (without refinement) to this pair, the resulting aligned volume is shown as the rightmost subplot in Figure 9, which turns out to be misaligned, in that the left portion of the volumes are now mismatched.

To see what is happening in this case, let us give a heuristic explanation by considering the following two-dimensional abstraction of the volumes in Figure 10. Here I_1 is a semi-circle and I_2 is an extended (but slightly thinner) semi-circular arc with the same radius as I_1, where the extended portion of I_2 is considered as the heterogeneous part, in analogy with V_1 and V_2 in Figure 9. Now we consider two possible alignment of these two arcs as in Figure 10 and give a rough calculation of the corresponding W_1 losses in both cases.

Suppose the extended portion in I_2 has length ε. Recall that the W_1 distance has the interpretation of the amount of mass transportation needed from arc I_1 to the arc I_2. For Alignment 1, the optimal transport plan would be to map both the endpoints of I_1 to the endpoints of I_2, and to map every other point in the middle in a proportional way. This would incur a W_1 loss approximately equal to

$$W_1(I_1,I_2) \approx \int_0^1 \epsilon x dx = \frac{\epsilon}{2}. \qquad \text{(Alignment 1)}$$

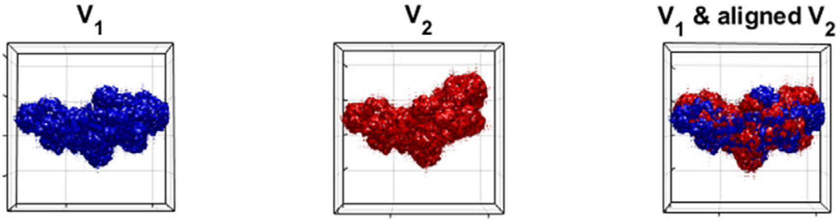

Figure 9. *Volume alignment with heterogeneity in Wasserstein distance.*

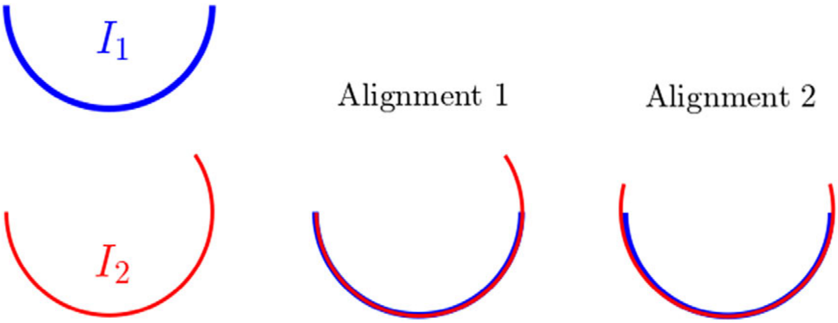

Figure 10. *Synthetic heterogeneous pair.*

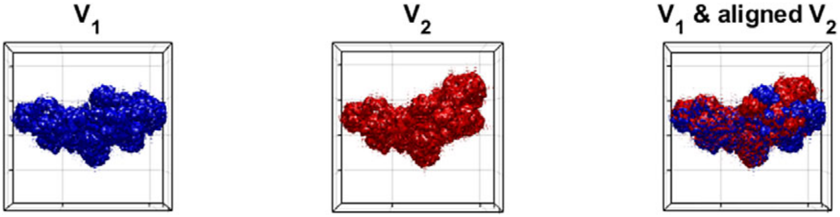

Figure 11. *Volume alignment with heterogeneity in Euclidean distance.*

For Alignment 2, the optimal transport plan would be to map each half of I_1 proportionally to the corresponding halves of I_2, which would incur a W_1 loss roughly

$$W_1(I_1, I_2) \approx 2 \cdot \int_0^{1/2} \epsilon x dx = \frac{\epsilon}{4}. \qquad \text{(Alignment 2)}$$

Therefore Alignment 2 leads to a smaller W_1 loss and is preferred as we have observed in Figure 9.

On the other hand, the L^2 losses between I_1 and I_2 remain the same regardless of whether Alignment 1 or 2 is applied. An implementation of Algorithm 1 with WEMD replaced by the L^2 loss actually gives close-to-correct alignment as shown in Figure 11, with an error about 3 degrees. This suggests that the L^2 loss might be a better choice in the presence of heterogeneity and explains its choice in our refinement step to avoid the effect of perturbations of the given volumes.

However, the empirical success of L^2 alignment in Figure 11 may be a coincidence thanks to other finer structures of the molecules. Our synthetic example in Figure 10 does raise an interesting question on what loss function to be used when aligning heterogeneous pairs. In particular, there is a potential need of a more sophisticated distance function d_{heter} that for instance only compares the shared components in the volumes by extracting the common features. In the context of compositional heterogeneity where the two volumes have different total masses as a result of missing subunits, Ref. (19) proposes a partial alignment procedure based on Gromov-Wasserstein divergence. We shall leave such investigation of more advanced distance functions for future work. Meanwhile, we remark that in this case, exhaustive search-based methods that rely on efficient representations of the loss function over SO(3) such as Ref. (20) would be challenging unless certain special structures of d_{heter} can be exploited. The same is likely to be true for gradient-based methods where one would need to rely on numerical differentiation which could be less accurate and inefficient. Nevertheless, our Algorithm 1 provides a ready-to-use recipe as long as we can evaluate d_{heter}.

6. Discussion

In this article, we proposed an alignment algorithm using a Wasserstein-based distance as the loss function, which is optimized with tools from Bayesian optimization followed by a local refinement procedure. Numerical experiments show improved performance of our algorithm over existing methods on the alignment of real protein molecules from cryo-EM. The proposed algorithm can be extended to arbitrary loss functions, which could be a feasible solution in the presence of heterogeneity where we have illustrated a potential need of novel distance functions. We have presented the algorithmic framework for volumes represented as density maps, but it can be easily extended to other volume representations as long as one can define a suitable loss function as in (3).

Our algorithm focuses mainly on the clean volume case, where the problem reduces to rotation estimation since the relative translation can be recovered by the centering step. We remark that this is not a simplifying assumption in the context of cryo-EM since 3D alignment is usually carried out on reconstructed molecules, which are much cleaner than their projection images to start with. However, this could be a limitation of our approach, where in the presence of noise more sophisticated center of mass estimation methods such as Ref. (41) would be necessary.

On the other hand, we remark that it is possible to incorporate translation estimation in the Bayesian optimization framework that we have adopted. In particular, the alignment objective in (15) and its surrogate problem (12) can be both extended to the product space $\mathbb{B} \times \mathrm{SO}(3)$. The algorithmic framework proceeds as before except the covariance function defining the GP would also need to be extended to $\mathbb{B} \times \mathrm{SO}(3)$. This for instance can be achieved simply with the product covariance

$$c((v,R),(w,S)) = \sigma^2 \exp\left(-\frac{\|R-S\|_F^2}{2\ell_r^2}\right) \exp\left(-\frac{\|v-w\|_2^2}{2\ell_s^2}\right).$$

However, our simulations suggest slow exploration of the search space and poor recovery in comparable amount of time as existing methods. This may be due to the fact that Bayesian optimization is known to work better in lower dimensions. The additional set $\mathbb{B}$ doubles the search space dimension and necessitates many more observations of the loss function to decode its landscape.

In the context of aligning heterogeneous pairs of volumes, there are still many improvements of the current work that are worth exploring. As a referee pointed out, the WEMD framework might be able to inform us about which parts of the volumes need further transport after obtaining an optimal alignment. Here we outline the idea without going into the mathematical details. As noted in Ref. (33), computing the W_1 distance between two density maps ϕ_1 and ϕ_2 is equivalent to the following optimization problem

$$\max_{f\text{ 1-Lipschitz}} \int f(x)[\phi_1(x) - \phi_2(x)]dx,$$

whose solution f can be approximated by a wavelet expansion with coefficients $f_\lambda = \mathrm{sign}(p_\lambda) \cdot 2^{-j(1+n/2)}$, where p_λ's are the wavelet coefficients of the difference density map $p = \phi_1 - \phi_2$. Therefore f is computable and according to the Kantorovich–Rubenstein duality theorem, it corresponds to the Lagrange multiplier of the marginal constraints involving ϕ_1 and ϕ_2. It can then be used to locate regions where the marginal constraints could be relaxed and hence regions where further transport is needed. This would have important applications in heterogeneous alignment and we wish to explore such ideas in the future.

Data availability statement. Replication data and code can be found at https://github.com/RuiyiYang/BOTalign.

Acknowledgments. R.Y. would like to thank Marc Aurèle Gilles and Oscar Mickelin for sharing protein data and useful codes.

Author contributions. A.S. and R.Y. conceived the project. R.Y. designed the algorithm and completed the technical work. Both authors approved the final submitted draft.

Funding statement. This research was supported in part by AFOSR under Grant FA9550-20-1-0266, in part by Simons Foundation Math+X Investigator Award, in part by NSF under Grant DMS-2009753, and in part by NIH/NIGMS under Grant R01GM136780-01.

Competing interest. The authors declare no competing interests exist.

Ethical standards. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

1. Chen D-Y and Ouhyoung M (2006) A 3d model alignment and retrieval system.
2. Makadia A, Patterson A and Daniilidis K (2006) Fully automatic registration of 3d point clouds. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. **1**. New York: IEEE, pp. 1297–1304.
3. Saupe D and Vranić DV (2001) 3d model retrieval with spherical harmonics and moments. In *Pattern Recognition: 23rd DAGM Symposium Munich, Germany, September 12–14, 2001 Proceedings 23*. Berlin: Springer, pp. 392–397.
4. Han X, Terashi G, Christoffer C, Chen S and Kihara D (2021) Vesper: Global and local cryo-em map alignment using local density vectors. *Nature Communications* **12**(1), 2090.
5. Joseph AP, Lagerstedt I, Jakobi A, … Winn M (2020) Comparing cryo-em reconstructions and validating atomic model fit using difference maps. *Journal of Chemical Information and Modeling* **60**(5), 2552–2560.

6. Kawabata T (2008) Multiple subunit fitting into a low-resolution density map of a macromolecular complex using a gaussian mixture model. *Biophysical Journal* **95**(10), 4643–4658.
7. Lawson CL, Patwardhan A, Baker ML, Hryc C, Garcia ES, Hudson BP, Lagerstedt I, Ludtke SJ, Pintilie G, Sala R, Westbrook JD, Berman HM, Kleywegt GJ and Chiu W (2016) Emdatabank unified data resource for 3dem. *Nucleic Acids Research* **44** (D1), D396–D403.
8. Kyatkin AB and Chirikjian GS (2000) Algorithms for fast convolutions on motion groups. *Applied and Computational Harmonic Analysis* **9**(2), 220–241.
9. Rubner Y, Tomasi C and Guibas LJ (2000) The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* **40**(2), 99.
10. Arjovsky M, Chintala S and Bottou L (2017) Wasserstein generative adversarial networks. In *International Conference on Machine Learning*. Sydney, VIC: PMLR, pp. 214–223.
11. Zelesko N, Moscovich A, Kileel J and Singer A (2020) Earthmover-based manifold learning for analyzing molecular conformation spaces. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. Iowa City, IA: IEEE, pp. 1715–1719.
12. Hamm K, Henscheid N and Kang S (2022) Wassmap: Wasserstein isometric mapping for image manifold learning. *arXiv preprint arXiv:2204.06645*.
13. Rao R, Moscovich A and Singer A (2020) Wasserstein k-means for clustering tomographic projections. *arXiv preprint arXiv: 2010.09989*.
14. Althloothi S, Mahoor MH and Voyles RM (2013) A robust method for rotation estimation using spherical harmonics representation. *IEEE Transactions on Image Processing* **22**(6), 2306–2316.
15. Kazhdan M (2007) An approximate and efficient method for optimal rotation alignment of 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(7), 1221–1229.
16. Pettersen EF, Goddard TD, Huang CC, … Ferrin TE (2004) Ucsf chimera—A visualization system for exploratory research and analysis. *Journal of Computational Chemistry* **25**(13), 1605–1612.
17. Chirikjian GS, Kim PT, Koo J-Y and Lee CH (2004) Rotational matching problems. *International Journal of Computational Intelligence and Applications* **4**(04), 401–416.
18. Riahi AT, Woollard G, Poitevin F, Condon A and Duc KD (2023) Alignot: An optimal transport based algorithm for fast 3d alignment with applications to cryogenic electron microscopy density maps. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **20**, 3842.
19. Riahi AT, Zhang C, Chen J, Condon A and Duc KD (2023) Empot: Partial alignment of density maps and rigid body fitting using unbalanced gromov-wasserstein divergence. arXiv preprint arXiv:2311.00850.
20. Chen Y, Pfeffer S, Hrabe T, Schuller JM and Förster F (2013) Fast and accurate reference-free alignment of subtomograms. *Journal of Structural Biology* **182**(3), 235–245.
21. Rangan AV (2022) Radial recombination for rigid rotational alignment of images and volumes. *Inverse Problems* **39**(1), 015003.
22. Healy Jr DM, Rockmore DN, Kostelec PJ and Moore SS (2002) Ffts for the 2-sphere-improvements and variations.
23. De la Rosa-Trevín J, Otón J, Marabini R, Zaldívar A, Vargas J, Carazo J, and Sorzano C (2013) Xmipp 3.0: An improved software suite for image processing in electron microscopy. *Journal of Structural Biology* **184**(2), 321–328.
24. Tang G, Peng L, Baldwin PR, Mann DS, Jiang W, Rees I, and Ludtke SJ (2007) Eman2: An extensible image processing suite for electron microscopy. *Journal of Structural Biology* **157**(1), 38–46.
25. Yu L, Snapp RR, Ruiz T and Radermacher M (2013) Projection-based volume alignment. *Journal of Structural Biology* **182**(2), 93–105.
26. Harpaz Y and Shkolnisky Y (2023) Three-dimensional alignment of density maps in cryo-electron microscopy. *Biological Imaging* **3**, e8.
27. Frazier PI (2018) A tutorial on Bayesian optimization. arXiv preprint arXiv:1807.02811.
28. Rasmussen CE and Williams CK (2006) *Gaussian Processes for Machine Learning*, vol. **1**. Berlin: Springer.
29. Feragen A and Hauberg S (2016) Open problem: Kernel methods on manifolds and metric spaces. What is the probability of a positive definite geodesic exponential kernel? In *Conference on Learning Theory*. New York: PMLR, pp. 1647–1650.
30. Jaquier N, Borovitskiy V, Smolensky A, Terenin A, Asfour T and Rozo L (2022) Geometry-aware Bayesian optimization in robotics using Riemannian Matérn kernels. In *Conference on Robot Learning*. London: PMLR, pp. 794–805.
31. Stein ML (1999) *Interpolation of Spatial Data: Some Theory for Kriging*. New York: Springer Science & Business Media.
32. Kanagawa M, Hennig P, Sejdinovic D, and Sriperumbudur BK (2018) Gaussian processes and kernel methods: A review on connections and equivalences. arXiv preprint arXiv:1807.02582.
33. Shirdhonkar S and Jacobs DW (2008) Approximate earth mover's distance in linear time. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage, AK: IEEE, pp. 1–8.
34. Srinivas N, Krause A, Kakade SM and Seeger M (2009) Gaussian process optimization in the bandit setting: No regret and experimental design. arXiv preprint arXiv:0912.3995.
35. Boumal N, Mishra B, Absil P-A and Sepulchre R (2014) Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research* **15**(42), 1455–1459.
36. Townsend J, Koep N and Weichwald S (2016) Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *The Journal of Machine Learning Research* **17**(1), 4755–4759.

37. Cuturi M (2013) Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*. Kyoto: Kyoto University, p. 26.
38. Lee G, Gommers R, Waselewski F, Wohlfahrt K and O'Leary A (2019) Pywavelets: A Python package for wavelet analysis. *Journal of Open Source Software* **4**(36), 1237.
39. Kileel J, Moscovich A, Zelesko N and Singer A (2021) Manifold learning with arbitrary norms. *Journal of Fourier Analysis and Applications* **27**(5), 82.
40. Wright G, Andén J, Bansal V, Xia J, Langfield C, Carmichael J, Brook R, Shi Y, Heimowitz A, Pragier G, Sason I, Moscovich A, Shkolnisky Y and Singer A (2023) Computationalcryoem/aspire-python: v0.11.0.
41. Heimowitz A, Sharon N and Singer A (2021) Centering noisy images with application to cryo-em. *SIAM Journal on Imaging Sciences* **14**(2), 689–716.

Cite this article: Singer A & Yang R (2024). Alignment of density maps in Wasserstein distance. *Biological Imaging* 4: e5. doi: https://doi.org/10.1017/S2633903X24000059