Toolbox for Gromov-Wasserstein optimal transport: Application to unsupervised alignment in neuroscience

Masaru Sasaki [1◑], Ken Takeda [1◑], Kota Abe [1], Masafumi Oizumi [1*]

1 Graduate School of Arts and Science, The University of Tokyo, Meguro-ku, Tokyo, Japan

◑These authors contributed equally to this work.
* c-oizumi@g.ecc.u-tokyo.ac.jp

Abstract

Gromov-Wasserstein optimal transport (GWOT) has emerged as a versatile method for unsupervised alignment in various research areas, including neuroscience, drawing upon the strengths of optimal transport theory. However, the use of GWOT in various applications has been hindered by the difficulty of finding good optima, a significant challenge stemming from GWOT's nature as a non-convex optimization method. It is often difficult to avoid suboptimal local optima because of the need for systematic hyperparameter tuning. To overcome these obstacles, this paper presents a user-friendly GWOT hyperparameter tuning toolbox (GWTune) specifically designed to streamline the use of GWOT in neuroscience and other fields. The toolbox incorporates Optuna, an advanced hyperparameter tuning tool that uses Bayesian sampling to increase the chances of finding favorable local optima. To demonstrate the utility of our toolbox, we first illustrate the qualitative difference between the conventional supervised alignment method and our unsupervised alignment method using synthetic data. Then, we demonstrate the applicability of our toolbox using some typical examples in neuroscience. Specifically, we applied GWOT to the similarity structures of natural objects or natural scenes obtained from three data domains: behavioral data, neural data, and neural network models. This toolbox is an accessible and robust solution for practical applications in neuroscience and beyond, making the powerful GWOT methodology more accessible to a wider range of users. The open source code for the toolbox is available on GitHub. This work not only facilitates the application of GWOT, but also opens avenues for future improvements and extensions.

Introduction

Unsupervised alignment is a useful methodology for finding the optimal mappings between items in different domains when the correspondences between the items are completely unknown or not entirely given. Mathematically, unsupervised alignment tries to find the optimal mapping between two point clouds in different domains where the distances (or correspondences) between points "across" different domains are not given. The key mathematical challenge of unsupervised alignment is that the alignment needs to be performed based on the internal relationships between points only, given that these are only information available.

As a promising approach to unsupervised alignment, the Gromov-Wasserstein optimal transport (GWOT) method [1] has been applied with great success in various fields: for example, matching of 3D objects [1], translation of vocabularies in different languages [2, 3], and matching of single cells in single-cell multi-omics data [4]; and in neuroscience, alignment of different individual brains in fMRI data [5] and comparison

of color similarity structures between different individuals [6, 7]. GWOT is based on the mathematical theory of optimal transport [8]. GWOT finds the correspondence between the point clouds in different domains based only on the internal relations (distances) between the points in each domain. In short, GWOT aligns the point clouds according to the principle that a point in one domain should correspond to another point in another domain that has a similar relationship to other points.

The advantages of using GWOT in place of other methods, such as Deep Neural Network (DNN)-based methods [9], are (1) simplicity and interpretability of the algorithms, and (2) computational tractability. First, the objective function of GWOT is simple and can be interpreted as an extension of the well-researched Wasserstein distance. As a result of the optimization, we can obtain the Gromov-Wasserstein "distance" measure between similarity structures which satisfies the axiom of distance. We also obtain the optimal transportation plan, i.e. the optimized mappings between items in different domains. Second, although GWOT involves the computation of $\mathcal{O}(N_1^2 N_2^2)$ (where N_1 and N_2 is the number of elements in one domain and the other domain, respectively) order complexity in the naive implementation, it can be reduced to $\mathcal{O}(N_1^2 N_2 + N_2^2 N_1)$ order complexity [8]. This reduction in computational complexity makes unsupervised alignment between structures consisting of up to about $N = 10^4$ elements practically feasible [2].

Despite the broad potential impact on various fields, applying GWOT is not necessarily easy because (1) it is a non-convex optimization problem and (2) it involves hyperparameter tuning. First, like other non-convex optimization problems, the algorithm of GWOT reveals only local minima, which are sometimes good and sometimes bad in terms of minimizing the objective function. Thus, in general, we typically need to conduct many iterations to obtain good local minima, with many failed attempts required to obtain good minima. Second, to find good local minima, effective control of the hyperparameter for entropy regularization is known to be important. These tasks are not necessarily easy for peripheral users, however, because there is not much guidance on how to do this in the existing toolboxes, such as Python Optimal Transport (POT) [10]; and also because each iteration of finding local minima takes a considerable amount of time.

To overcome these obstacles, we present here a user-friendly toolbox for GWOT hyperparmeter tuning (GWTune) in neuroscience and other fields. Our toolbox uses POT [10] to perform a basic optimization of GWOT, and also uses Optuna [11] for hyperparameter tuning. POT is a toolbox that provides a comprehensive range of solvers for various types of optimal transportation problem, including GWOT, but does not support hyperparameter tuning. Optuna is an advanced hyperparameter tuning toolbox that implements various tuning methods, including Bayesian sampling, as used in the present research, to increase the chances of finding favorable local optima in a reasonable amount of time.

To demonstrate the applicability of our toolbox, especially in the field of neuroscience, we present several applications in three data domains: behavioral data, neural data, and neural network models. For behavioral data, we analyzed psychological similarity judgments of natural objects from the THINGS dataset [12–14]. We used GWOT to align the similarity structures of natural objects across different participant groups. For neural data, we analyzed data from mouse neuropixel recordings from the Allen Brain Institute [15, 16]. We applied GWOT to align the similarity structures of neural responses to different visual images across individuals. For neural network models, we analyzed the internal representations of the vision Deep Neural Network models (DNNs). We applied GWOT to align the similarity structures of the activities of model neurons between ResNet50 [17] and VGG19 [18].

We expect that this toolbox represents a robust solution for various research

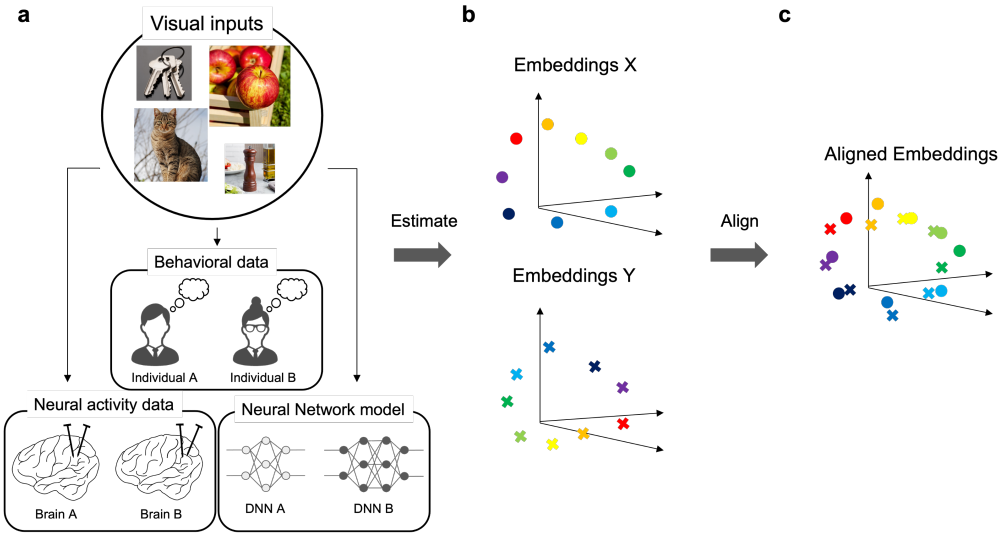

Fig 1. Alignment in neuroscience. (a) Three data domains in neuroscience. Behavioral data, neural activity data, and neural network models. As examples, behavioral responses in different individuals, neural responses in different brains, and model neuron activities in different deep neural network (DNN) models to visual inputs are considered. (b) Embeddings estimated from the three data domains. (c) Alignment of the two embeddings.

applications in various fields, making the powerful GWOT methodology accessible to a wider range of users. In particular, in neuroscience, GWOT can be used for alignment between behavioral data, neural data, or data from neural network models, and also between alignment across different data domains, such as behavior vs. neural, or model vs. neural, etc. The open source code for the toolbox is available on GitHub. `https://oizumi-lab.github.io/GWTune/`

Alignment in neuroscience

In this paper, we consider the problem of alignment in neuroscience as a specific example (Fig. 1), but the toolbox is readily applicable to data in other research areas. Data in neuroscience for which alignment is considered in the current literature can be categorized into three main targets: neural data, behavioral data, and neural network models (Fig. 1a). Neural data include any kind of recordings of neural activity, such as neuropixels, Ca imaging, EEG, ECoG, fMRI, etc. Behavioral data include any kind of behavioral reports, e.g. similarity judgments and ratings, multiple choice, free verbal or written descriptions, etc. Neural network models include any type of neural network model, e.g. convolutional neural networks, recurrent neural networks, transformers, etc. There may be other types of data to which the alignment framework in this paper is applicable, but here we consider these three main targets for simplicity and clarity.

Despite the variety of data types, we consider the same common alignment problem in this paper: the alignment of point clouds (Fig. 1b). A point cloud is a set of data points in space, usually defined by a set of coordinates in space. The first step in alignment is to convert raw data into a set of points in space. For example, in the case of neural data or neural network models, we can simply use the firing rates of real

neurons or the activities of model neurons to a set of different stimuli as points in space, where the dimension represents the activity of each neuron. Alternatively, we can consider linear transformations of neural activity and create new dimensions to represent the original neural activity. In the case of behavioral data, we typically estimate so-called psychological embeddings [13, 19] that best explain the behavioral responses to different stimuli, and treat the estimated embeddings as points in space. In this paper, we use the common terminology "embeddings" to mean points in space regardless of whether the data are neural, behavioral, or model. Given the sets of estimated embeddings, they are aligned, typically by minimizing some objective functions.

There are two broad categories of alignment method: supervised alignment and unsupervised alignment. The methods of supervised alignment align the sets of embeddings with given correspondences between the embeddings in the different sets. In contrast, the methods of unsupervised alignment do this without given correspondences, but rather simultaneously find the correspondences during the optimization process. Most neuroscience studies to date have only considered supervised alignment. The most widely and commonly used method is Representational Similarity Analysis (RSA) [20–22]. Although unsupervised alignment methods have not yet been applied in many cases in neuroscience, various applications will indeed be identified (e.g. [5, 6]). As we will show later in the Results section, unsupervised alignment methods can reveal structural similarities or differences with greater detail than supervised alignment.

In this toolbox, we implement the unsupervised alignment method using Gromov-Wasserstein Optimal Transport (GWOT). Here, we provide a concise explanation of GWOT. Mathematical details of supervised alignment, unsupervised alignment, and GWOT are described in the Supporting Information.

Gromov-Wasserstein Optimal Transport (GWOT)

Gromov-Wasserstein optimal transport [1] is an unsupervised alignment technique that finds correspondence between two point clouds (embeddings) in different domains based only on internal distances within the same domain. Mathematically, the goal of the Gromov-Wasserstein optimal transport problem is to find the optimal transportation plan Γ between the embeddings in different domains, given internal distance matrices D and D' within the same domains (Fig. 2a). The transportation cost, i.e., the objective function, considered in GWOT is given by

$$\min_{\Gamma} \sum_{i,j,k,l} (D_{ij} - D'_{kl})^2 \Gamma_{ik} \Gamma_{jl}. \tag{1}$$

Note that a transportation plan Γ must satisfy the following constraints: $\sum_j \Gamma_{ij} = p_i$, $\sum_i \Gamma_{ij} = q_j$ and $\sum_{ij} \Gamma_{ij} = 1$, where $\boldsymbol{p}$ and $\boldsymbol{q}$ is the source and target distribution of resources for the transportation problem, respectively, whose sum is 1. Under this constraint, the matrix Γ is considered as a joint probability distribution with the marginal distributions being $\boldsymbol{p}$ and $\boldsymbol{q}$. As for the distributions $\boldsymbol{p}$ and $\boldsymbol{q}$, we set $\boldsymbol{p}$ and $\boldsymbol{q}$ to be the uniform distributions. Each entry Γ_{ij} describes how much of the resources on the i-th point in the source domain should be transported onto the j-th point in the target domain. The entries of the normalized row $\frac{1}{p_i}\Gamma_{ij}$ can be interpreted as the probabilities that the embedding $\boldsymbol{x}_i$ corresponds to the embeddings $\boldsymbol{y}_j$.

Hyperparameter tuning

Entropy regularization ϵ

Previously, it has been demonstrated that adding an entropy-regularization term can improve computational efficiency and help identify good local optimums of the

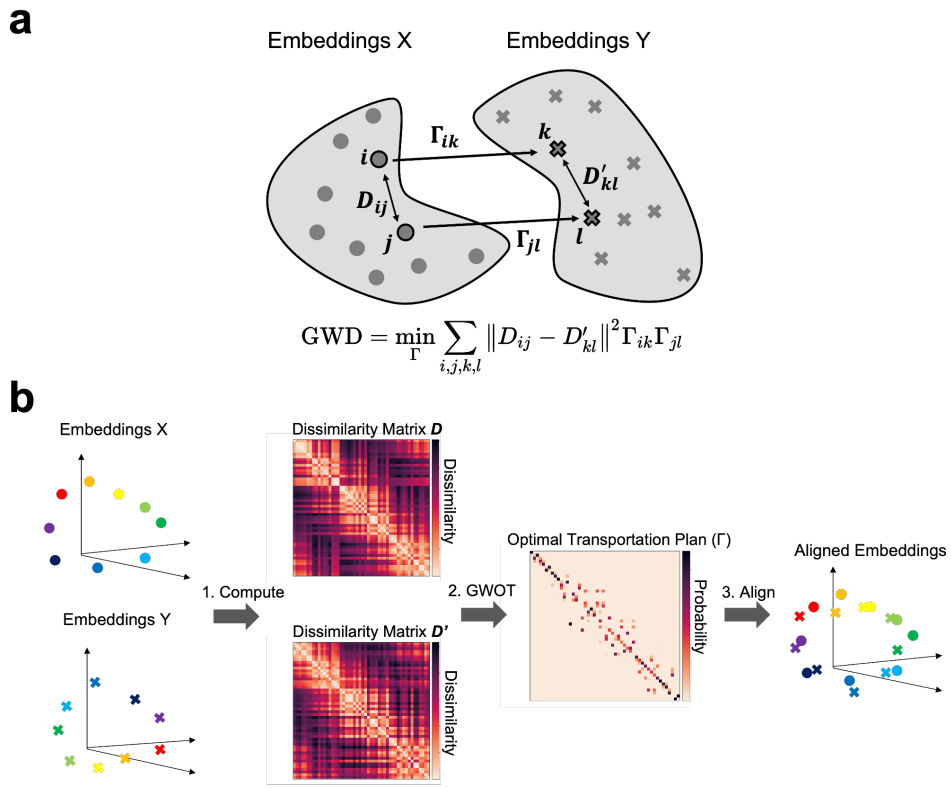

Fig 2. Unsupervised alignment using GWOT. (a) Schematic of Gromov-Wasserstein optimal transportation (GWOT). D_{ij} represents the distance between the ith embedding and the jth embedding in one set of embeddings X and D'_{kl} represents the distance between the kth embedding and the lth embedding in the other set of embeddings Y. Γ_{ik} represents the amount of transportation from the ith embedding in X and the kth embedding in Y. (b) The process of unsupervised alignment using GWOT.

Gromov-Wasserstein optimal transport problem [8, 23].

$$\min_{\Gamma} \sum_{i,j,k,l} (D_{ij} - D'_{kl})^2 \Gamma_{ik}\Gamma_{jl} + \epsilon H(\Gamma), \tag{2}$$

where $H(\Gamma)$ is the entropy of a transportation plan.

To find good local optimums, we need to conduct hyperparameter tuning on ϵ in Eq. 2. After hyperparameter tuning, we choose the value of ϵ, where the optimized transportation plan minimizes the Gromov-Wasserstein distance without the entropy-regularization term (Eq. 1) following the procedure proposed in a previous study [4].

Initialization of transportation plan Γ

In some applications, it is not enough to simply search for the hyperparameter ϵ, but it is also necessary to search for different initializations. To avoid getting stuck in bad local minima, it is effective to randomly initialize the transportation plan and try many random initializations. See Supporting Information for details of hyperparameter tuning on ϵ and initialization.

Unsupervised alignment using GWOT

The entire process of unsupervised alignment using GWOT is summarized in Fig. 2b.

1. The distance (dissimilarity) matrices are computed from the embeddings.
2. Optimization of GWOT is performed between the distance matrices.
3. The embeddings are aligned using the optimal transportation plan Γ found by GWOT.

See Supporting Information for details of alignment.

Design and Implementation

Overview

The main aim of this toolbox is to enable effective and user-friendly hyperparameter tuning for GWOT. For this purpose, we use two key Python libraries: Optuna for hyperparameter tuning and POT (Python Optimal Transport) for GWOT optimization. Optuna is an automatic hyperparameter optimization software framework that is widely used in machine learning [11]. Optuna provides state-of-the-art algorithms for sampling hyperparameters and pruning unpromising trials. POT provides comprehensive solvers for various types of optimal transportation problem, including GWOT. However, as mentioned above, hyperparameter tuning for GWOT is not supported, although hyperparameter tuning is essential in applications. Our toolbox complements the optimization of GWOT implemented in POT by enabling hyperparameter tuning using Optuna.

Code flow

Step 1: Preparation of input data

First, dissimilarity matrices or sets of embeddings are prepared for use in subsequent unsupervised alignment. When embeddings are provided, the dissimilarity matrices are computed by the distances between the embeddings. There are many ways to compute the distances, including cosine distance, Euclidean distance, and inner product. See the specific examples of embeddings and dissimilarity matrices in the Applications section.

Step 2: Setting parameters for optimization of GWOT

Second, important parameters for hyperparameter tuning should be set. As mentioned above, the key point for good optimization is hyperparameter tuning on (1) the ε value of the entropy regularization and (2) the initialization of transportation plan Γ. See Supporting Information for details.

Step 3: Optimization of Gromov-Wasserstein distance

After Step 1 (providing the dissimilarity matrices) and Step 2 (setting the parameters), the user is now ready to perform the optimization of the Gromov-Wasserstein distance (GWD) with entropic regularization.

We provide a Python pseudo code that summarizes the process from Step 1 to Step 3 in Algorithm 1. This code is pseudo code in the sense that it cannot be executed as is, but the names of the classes or methods are real. As Step 1, two data of embeddings are stored in the class `Representation`. Next, as Step 2, some important parameters such

Algorithm 1 Pseudo code of optimization of GWOT

```
rep1 = Representation(embedding1)
rep2 = Representation(embedding2)
representations_list = [rep1, rep2]                                   ▷ Step1

config = OptimizationConfig(sampler = 'TPE', init_matrix = 'random',
    num_trials=100, epsilon_list=[0.01, 0.1])                         ▷ Step2

align_representation = AlignRepresentations(config, representations_list)
align_representation.gw_alignment()                                   ▷ Step3
```

as `sampler` (the sampler method), `init_matrix` (the initialization method), `num_trials` (the number of hyperparameter tuning trials on ϵ), and `epsilon_list` (the range of ϵ) are set by using the class `OptimizationConfig`. Then, as Step 3, the GWD is optimized by applying the `gw_alignment` method to the `AlignRepresentations` class object. As a result, for each ϵ value, the optimized transportation plans and the corresponding GWDs are obtained.

Step 4: Evaluation of results

Finally, it is critical to evaluate the optimization results from several perspectives: matching rate of unsupervised alignment, comparison between local minima and visualization of aligned embeddings. In the toolbox, the optimization results can be evaluated by applying the corresponding methods to the `AlignRepresentations` class object shown in Algorithm 1. See Supporting Information for details.

Results

Here, we demonstrate the application of our toolbox to three different types of data: behavioral data, neural data, and models. Before dealing with real data, we will first illustrate the differences between supervised and unsupervised alignment using synthetic data. We will then demonstrate the results on real data.

For all the following applications, we used `TPESampler` to sample ϵ values and `random` option to initialize transportation plans. Matching rate of unsupervised alignment is evaluated based on the optimized transportation plan with the minimum GWD, assuming that the correct assignment matrix is the diagonal matrix; in other words, that the same indexes between two dissimilarity matrices correspond to each other.

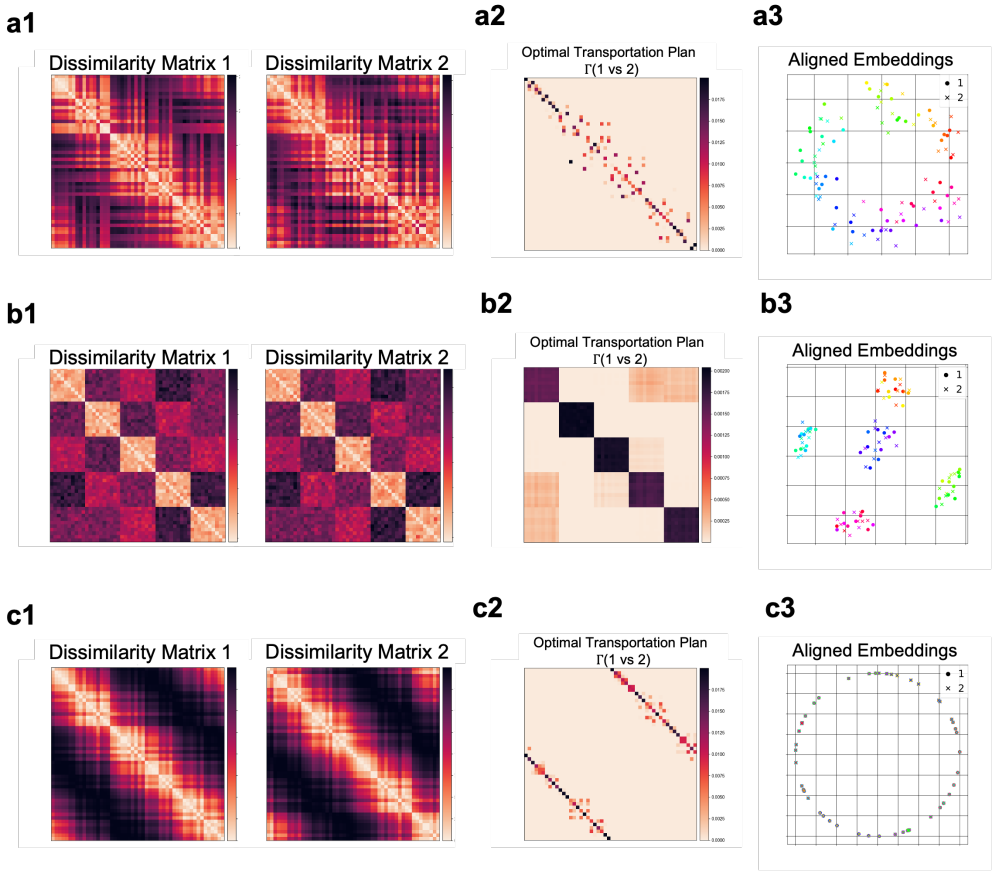

Fig 3. Unsupervised alignment in synthetic data, revealing structural similarities and differences with greater detail than supervised alignment. In all three cases, the Spearman correlation based on the RSA framework between the two distance matrices is constant at 0.90. In contrast, unsupervised alignment yields qualitatively different results, distinguishing the structural differences between the three cases. (a1, b1, c1) Two distance matrices of 50 points in the two groups. (a2, b2, c2) Optimized transportation plan Γ^* between the dissimilarity matrices 1 and 2. (a3, b3, c3) Embeddings of dissimilarity matrix 1 (X) and the aligned embeddings of dissimilarity matrix 2 (QY) plotted in the embedded space of X. (a) Case in which there are one-to-one level correspondences between the embeddings of X and Y, and unsupervised alignment correctly finds the similar correspondences. (b) Case in which there are categorical (many-to-many) level correspondences between X and Y. (c) Case in which the assumed "correct" correspondences differ from the correspondences empirically found by unsupervised alignment.

Synthetic data illustrating the differences between supervised and unsupervised alignment

Unsupervised alignment can reveal structural similarities or differences between distance structures with greater detail than supervised alignment. Conventionally in neuroscience, similarity in distance structures is assessed using Representational Similarity Analysis (RSA) [20–22], which falls into the category of supervised alignment, because it assumes correspondences between the points. In the conventional RSA framework, the similarity of distance structures is, for example, typically evaluated by

Pearson or Spearman correlation between distance matrices. We use Pearson correlation as a typical example of RSA. In the following, we show three examples that illustrate qualitative differences between our unsupervised approach and the conventional RSA.

In Fig. 3, we show three toy examples where the correlation values computed in the conventional RSA framework are all the same, whereas the results of unsupervised alignment are qualitatively different. Specifically, we consider the following three patterns of point cloud. For instance, suppose that these embeddings represent the neural responses of two different brains to the same stimulus set (e.g., color stimuli). Assume that the rows and columns of the two dissimilarity matrices in Fig. 3 are sorted in the same stimulus order.

First, we consider the example where the supervised and unsupervised alignment methods lead to qualitatively similar conclusions in Fig. 3a. The simple correlation coefficient between the two dissimilarity matrices in Fig. 3(a1) is very high, 0.9. Then, Fig. 3(a2) shows the optimal transportation plan Γ found by the GWOT method applied to the two dissimilarity matrices. In this case, the diagonal elements of Γ have high values, which means that neural responses to the same stimuli are matched with each other in the different brains. The unsupervised alignment method confirms that the two dissimilarity structures are closely matched with each other at the level of individual stimuli. This conclusion would not differ from that obtained with the supervised alignment method.

However, in the case of Figs. 3b and c, the conclusions obtained by supervised and unsupervised alignment are qualitatively different. In the case of Fig. 3b, we assume that there are some coarse categories in the stimulus set (e.g., dogs, cats, cars, etc.), represented as the block structures in the dissimilarity matrix in Fig. 3(b1). Although the correlation coefficient is high, at 0.9 (as in Fig. 3a), the unsupervised alignment shows that the match is only at the coarse-categorical level. That is, the neural responses do not match at the level of individual stimuli. Furthermore, in the case of Fig. 3c, the neural responses to certain stimuli in one brain correspond to those to "different" stimuli (Fig. 3(c2)) in the other brain, even though the correlation coefficient is again high, at 0.9. This is clearly different from the case of Fig. 3a, in which the neural responses to the same stimuli match each other. Figs. 3(a3)(b3)(c3) show the aligned embeddings, illustrating the underlying structural differences between the three cases.

Taken together, unsupervised alignment reveals a more detailed structural correspondence between two sets of embeddings, whether it is a one-to-one fine "correct" mapping (Fig.3a) or a many-to-many more coarse mapping (Fig.3b), or the mapping is different from the assumed correspondences (Fig.3c). This is clearly and qualitatively different from the supervised alignment method because it gives exactly the same evaluation in the three cases, and thus cannot distinguish the results of Figs. 3b and c from the result of Fig. 3a.

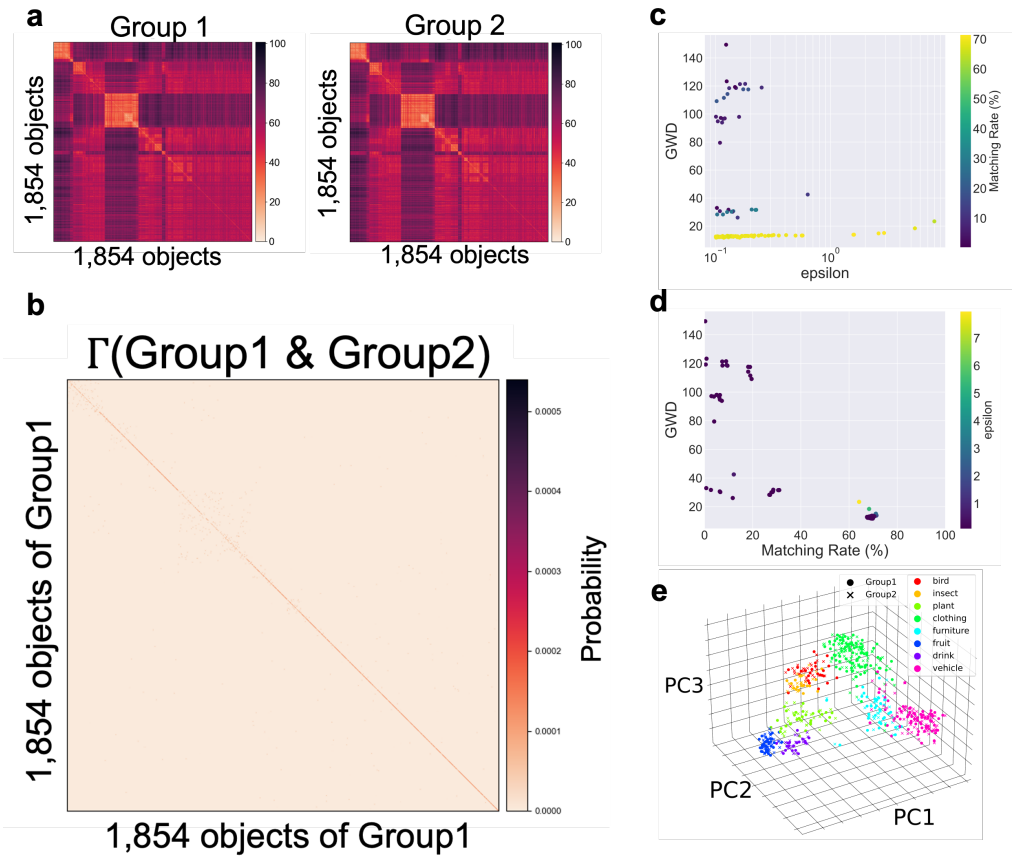

Fig 4. Unsupervised alignment between the psychological embeddings of naturalistic objects in two different participant groups. (a) Dissimilarity matrices of 1,854 naturalistic objects in participant Group 1 and Group 2. (b) Optimized Transportation plan Γ^* between the distance matrices of Group 1 and Group 2. (c) Relationship between GWD and hyperparameter ϵ. Color represents matching rate. (d) Relationship between GWD and matching rate. Color represents hyperparameter ϵ. (e) The embeddings of Group 1 and the aligned embeddings of Group 2 plotted in Group 1's embedded space. Only objects with certain coarse category labels ("bird", "insect", "plant", etc.) are shown here. Each dot represents the embedding of an object. Colors represent the coarse categories of the objects.

Behavioral data: Human psychological embeddings of natural objects

To demonstrate the alignment of behavioral data, we used the THINGS dataset, an open dataset containing human similarity judgments for 1,854 naturalistic objects [12–14]. We first estimated the psychological embeddings of 1,854 objects (see Supporting Information for the details). We then computed the dissimilarity matrix of the 1,854 natural objects for each participant group, in which the dissimilarity between objects is quantified by the Euclidean distance between the embeddings of the objects (Fig. 4a). As can be seen in Fig. 4a, the two dissimilarity matrices are closely similar, with a correlation coefficient of 0.96. However, as we showed in the toy example (Fig. 3b), a high correlation does not necessarily mean that the dissimilarity structures between the two groups match each other at the individual stimulus level. In fact, in the

THINGS dataset, there are also coarse categories in 1,854 objects, and thus it is possible that the two structures only match at the coarse categorical level, as shown in 3b.

Next, we performed GWOT between the dissimilarity matrices of the two groups. In Fig. 4b, we show the optimized transportation plan with the minimum GWD between Group 1 and Group 2. As shown in Fig. 4b, most of the diagonal elements in Γ^* have high values, indicating that most of the objects in one group correspond to the same objects in the other groups with high probability. This result means that the two similarity structures match at the level of individual objects.

To show the optimization results in detail, we reveal the relationship between the hyperparameter ϵ, GWD, and the matching rate of the unsupervised alignment over 100 iterations in Figs. 4c and d. Fig. 4c shows that good local minima with a low GWD as well as high matching rate are found over a wide range of epsilon between 0.1 and 10. Fig. 4d shows a downward trend to the right, i.e., lower GWD values tend to result in higher accuracy. This downward trend to the right is necessary for successful unsupervised alignment. The optimal solution with the minimum GWD has the matching rate of 70.0%.

Finally, using the optimal transportation plan, we aligned the psychological embeddings of natural objects across the two groups of participants. For visualization, we projected the original embeddings in 90 dimensions into three dimensions using Principal Component Analysis (PCA) (Fig. 4e). Also, since the display of all 1,854 objects would result in crowding, we show only those objects belonging to the 8 categories indicated by different colors in Fig. 4e. We can see that the projected neural responses to the same categories are positioned close to each other in the projected space. The top 1 categorical level matching rate is 88.4%.

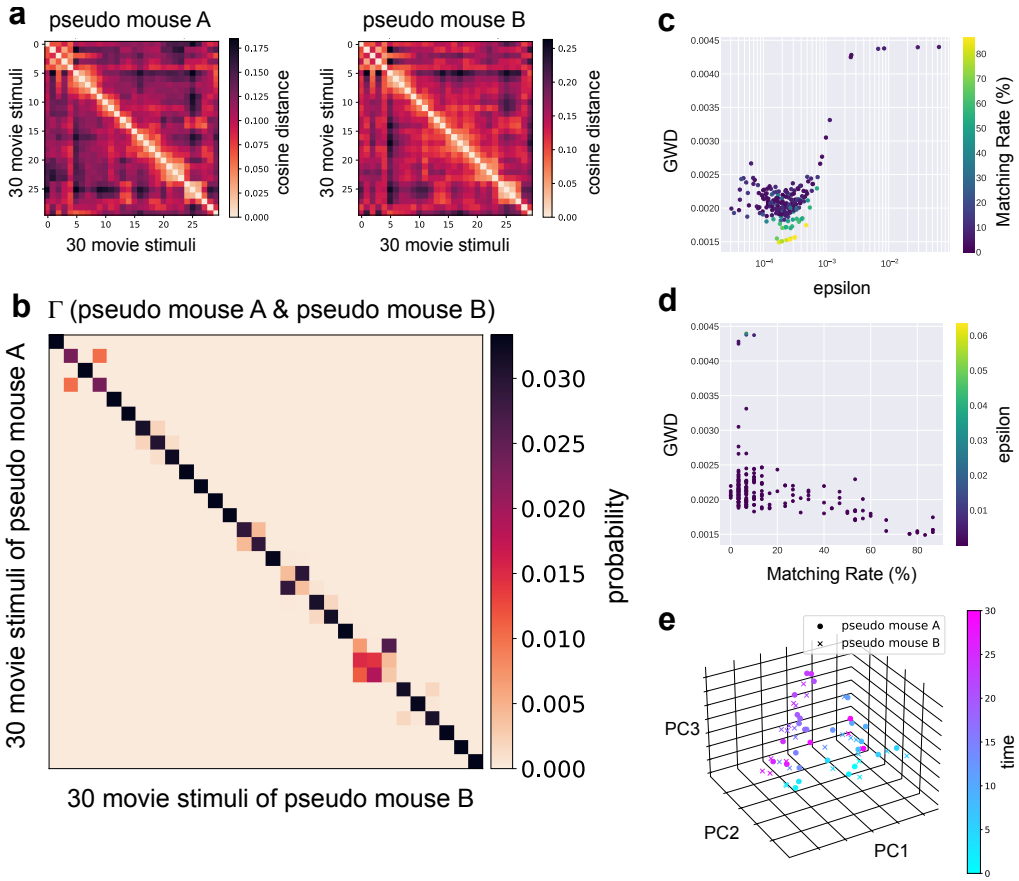

Fig 5. Unsupervised alignment between the neural representations of primary visual cortex (VISp) in two pseudo-mice. (a) Dissimilarity matrices in the two pseudo-mice for 30 segmented 1-second movie stimuli, made from a 30-second continuous natural movie stimulus. (b) Optimized transportation plan Γ^* between the dissimilarity matrices of the pseudo-mice. (c) Relationship between GWD and hyperparameter ϵ. Color represents matching rate. (d) Relationship between GWD and matching rate. Color represents hyperparameter ϵ. (e) The embeddings of pseudo-mouse A and the aligned embeddings of pseudo-mouse B plotted in the pseudo-mouse A's embedded space. Colors represent the time of the short movie stimuli.

Neural data: Neuropixel visual coding in mice

To demonstrate neural data alignment, we used the Neuropixels Visual Coding dataset from Allen Brain Observatory (see Supporting Information for the details). First, we computed the dissimilarity matrix of neural responses of primary visual cortex (VISp) in two pseudo-mice to 30 segmented 1-second movie stimuli (Fig. 5a). The dissimilarity is quantified as the cosine distance between the neural responses of the pseudo mice. As can been seen, these two dissimilarity matrices are quite similar (correlation coefficient ρ between them is 0.87).

Next, we performed the optimization of GWD between the dissimilarity matrices of the two pseudo-mice. To optimize the hyperparameter ϵ, we sampled 200 different values of ϵ within a range of 1.0×10^{-5} to 1.0×10^{-1}. For each sampled ϵ, we randomly initialized the transportation plans. We show the optimized transportation plan with the minimum GWD in Fig. 5b. As can be seen, almost all of the diagonal elements have the highest values (top 1 matching rate is 83.3%). Even in the case of mismatch of

movie segments, the mismatch occurs between the movie segments that are close in time and thus similar to each other. This demonstrates a strong structural correspondence between the neural responses in the two pseudo-mice.

To see the details of the optimization results, we show the relationship between the hyperparameter ϵ, GWD, and the matching rate of the unsupervised alignment over 200 iterations in Figs. 5c and d. Fig. 5c shows that GWD is optimized when ϵ is around 1.0×10^{-4}. When ϵ is lower than or higher than this range of values, the values of GWD are higher. Fig. 5d shows a downward trend to the right, i.e., lower GWD values tend to result in higher accuracy. This downward trend to the right is necessary for successful alignment.

Finally, using the optimal transportation plan, we aligned the neural responses of the two pseudo-mice. We performed this alignment using the responses of all neurons (900 neurons) in each pseudo-mouse, but for visualization, we projected the 900 dimensional neural responses into three dimensions using Principal Component Analysis (PCA) (Fig. 5e). We can see that the projected neural responses to the same movie stimulus are positioned close to each other and also that the responses to movie stimuli close in time (indicated by colors) are close to each other in the projected space.

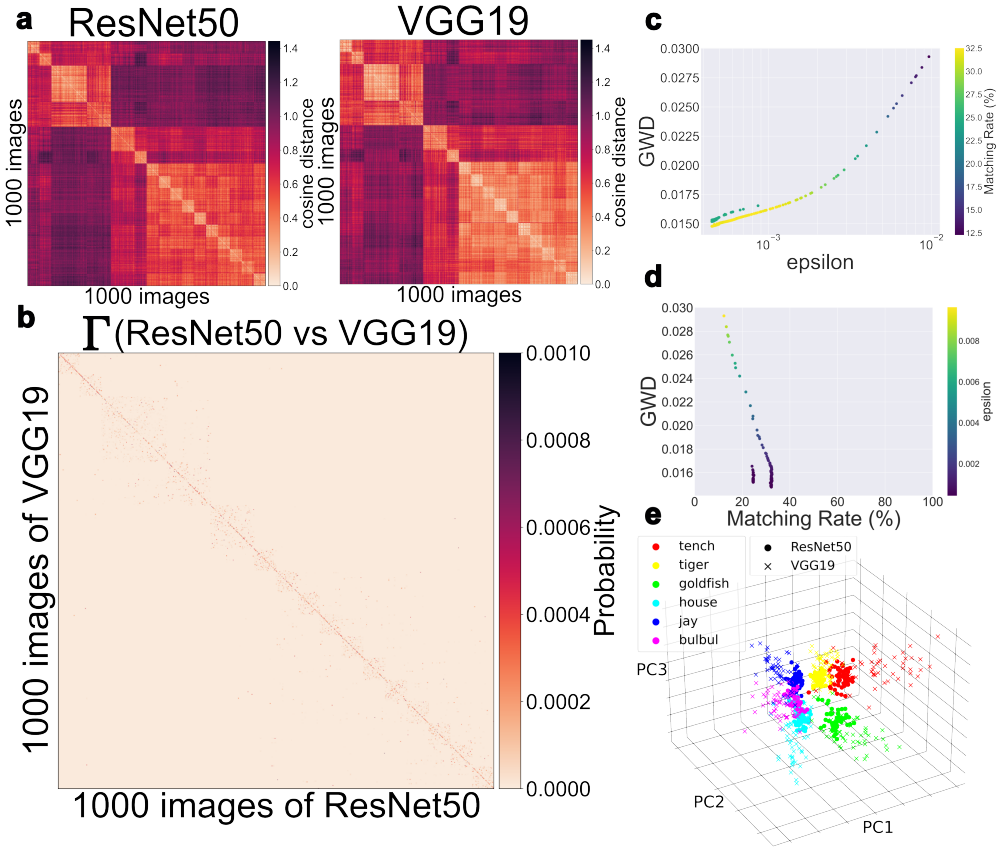

Fig 6. Unsupervised alignment between embeddings of the fully-connected layers of the vision DNNs, ResNet50 and VGG19. (a) Dissimilarity matrices of 1,000 natural objects in the two different vision DNNs, ResNet50 and VGG19. (b) Optimized transportation plan Γ between the dissimilarity matrices of ResNet50 and VGG19. (c) Relationship between GWD and hyperparameter ϵ. Color represents matching rate. (d) Relationship between GWD and matching rate. Color represents hyperparameter ϵ. (e) Embeddings of ResNet50 and the aligned embeddings of VGG19 plotted in ResNet50's embedded space. All objects with a class label ("goldfish", "house", "tiger", etc.) are shown. Each dot represents the embedding of each object. Colors represent the class of the images.

Model: Vision Deep Neural Networks

First, we computed the dissimilarity matrices of the 1,000 natural images belonging to 20 classes for ResNet50 and VGG19 as the cosine distance between the last fully-connected layer of the two DNNs. (Fig. 6a). The correlation coefficient between the two matrices are fairly high, with a correlation coefficient of 0.91. However, since there are 20 image classes, this high degree of correlation can be induced simply by categorical level correspondences, as seen in the toy example in Fig. 3b (note that the rows and columns of the matrices are sorted by class labels, i.e., 50 images from index i to index $i + 49$ belong to the same class where $i \equiv 0 \mod 50$). Thus, it is important to assess whether these embeddings match at the individual image level or at the categorical level by using the unsupervised alignment.

Next, we performed the optimization of GWD between the dissimilarity matrices of the two DNNs. As shown in Fig. 6b, many diagonal elements in Γ^* have high values,

indicating that many embeddings in one DNN correspond to the embeddings of the same images in the other DNN with high probability. This result means that the two similarity structures of the two DNNs match at the level of individual images, even though the details of the network architectures are largely different. Even for mismatches, we can see high values close to the diagonal elements, which means that the embeddings in one DNN match those belonging to the same image class in the other DNN because the image indexes in Fig. 6a are sorted by class labels.

To see the details of the optimization results, we show the relationship between the hyperparameter ϵ, GWD, and the matching rate of the unsupervised alignment over 500 iterations in Figs. 6c and d. Fig. 6c shows the local minima with lower GWD values in the lower epsilon ranges below 10^{-3}. Fig. 6d shows a downward trend to the right, i.e., lower GWD values tend to result in higher matching rate, as is also the case in Fig. 4 and Fig. 5. The optimal solution with the minimum GWD has a top 1 matching rate of 32.2%.

Finally, using the optimal transportation plan, we aligned the embeddings of the two vision DNNs. For visualization, we projected the original embeddings of 1,000 model neurons into three dimensions using Principal Component Analysis (PCA) (Fig. 6e). Also, since the display of all 1,000 objects would result in crowding, we only show the objects belonging to 6 image classes indicated by different colors in Fig. 6e. We can see that the projected neural responses to the same categories are positioned close to each other in the projected space. Top 1 categorical level matching rate is 78.1%.

Availability and future directions

In this paper, we present the GWOT hyperparameter tuning toolbox (GWTune) for unsupervised alignment to facilitate many possible use cases, especially for neuroscience but also for other research areas. The source code for our toolbox is available at `https://oizumi-lab.github.io/GWTune/`. All examples shown in this paper can be tested in the tutorial notebooks to help new users get started. In addition, another example using the similarity judgment data of 93 colors previously reported in [6] is also available. The processed data used to generate the results presented in this study are also available in the "data" folder of the same GitHub repository.

Finally, we discuss several important cases for the application of GWOT that are not covered in this paper. The first is the case in which the number of points (embeddings) is different. In this study, we only consider the case where the number of points is the same and the set of external stimuli or inputs is the same. For example, in the case of the Allen neuropixels data, the same movie stimuli are used between the pseudo-mice. However, GWOT is applicable to cases in which the number of stimuli is different, or the stimuli themselves are different. For example, we can consider the alignment between neural responses to a set of N stimuli in one brain and neural responses to a different set of M stimuli in the other brain. GWOT can be readily used in such cases and will be useful in finding the correspondences among neural responses to different stimuli.

The second case involves the alignment between data from different modalities. For example, we can consider the alignment between neural activity and behavior or a model. This application is important for assessing how well a given neural network model captures neural responses in the brain or how well it explains behavior in humans or other animals [7, 24–29]. Moreover, we can also consider the alignment between behavior and neural activity, which is important for identifying neural correlates that explain a given behavior in humans or other animals [14, 30–35]. We hope that this toolbox will help researchers to apply GWOT to various types of data, including cases such as these, and provide novel insights that cannot be obtained by conventional supervised alignment methods.

Acknowledgments

We thank Genji Kawakita for early code contributions. We also thank Ariel Zeleznikow-Johnston and Naotsugu Tsuchiya for providing the data on color similarity judgments for the toolbox tutorial. This work was supported by JST Moonshot R&D Grant Number JPMJMS2012, and JSPS KAKENHI Grant Numbers 20H05712 and 23H04834.

References

1. Mémoli F. Gromov–Wasserstein Distances and the Metric Approach to Object Matching. Found Comut Math. 2011;11(4):417–487.

2. Alvarez-Melis D, Jaakkola T. Gromov-Wasserstein Alignment of Word Embedding Spaces. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; 2018. p. 1881–1890.

3. Alaux J, Grave E, Cuturi M, Joulin A. Unsupervised Hyper-alignment for Multilingual Word Embeddings. In: International Conference on Learning Representations; 2019.Available from: `https://openreview.net/forum?id=HJe62s09tX`.

4. Demetci P, Santorella R, Sandstede B, Noble WS, Singh R. SCOT: Single-Cell Multi-Omics Alignment with Optimal Transport. Journal of Computational Biology. 2022;29(1):3–18.

5. Thual A, Tran QH, Zemskova T, Courty N, Flamary R, Dehaene S, et al. Aligning individual brains with fused unbalanced Gromov Wasserstein. In: Advances in Neural Information Processing Systems; 2022.Available from: `https://openreview.net/forum?id=vy7B8z0-4D`.

6. Kawakita G, Zeleznikow-Johnston A, Takeda K, Tsuchiya N, Oizumi M. Is my "red" your "red"?: Unsupervised alignment of qualia structures via optimal transport; 2023. Available from: `https://psyarxiv.com/h3pqm/`.

7. Kawakita G, Zeleznikow-Johnston A, Tsuchiya N, Oizumi M. Comparing Color Similarity Structures between Humans and LLMs via Unsupervised Alignment; 2023. Available from: `https://doi.org/10.48550/arXiv.2308.04381`.

8. Peyré G, Cuturi M, Solomon J. Gromov-Wasserstein Averaging of Kernel and Distance Matrices. In: Proceedings of The 33rd International Conference on Machine Learning. vol. 48; 2016. p. 2664–2672. Available from: `http://hdl.handle.net/1721.1/112918`.

9. Conneau A, Lample G, Ranzato M, Denoyer L, Jégou H. Word translation without parallel data. In: arXiv; 2017.Available from: `https://doi.org/10.48550/arXiv.1710.04087`.

10. Flamary R, Courty N, Gramfort A, Alaya MZ, Boisbunon A, Chambon S, et al. POT: Python optimal transport. J Mach Learn Res. 2021;22(1):3571–3578.

11. Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A Next-Generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2019. p. 2623—2631.

12. Hebart MN, Dickter AH, Kidder A, Kwok WY, Corriveau A, Van Wicklin C, et al. THINGS: A database of 1,854 object concepts and more than 26,000 naturalistic object images. PLOS ONE. 2019;14(10):1–24.

13. Hebart MN, Zheng CY, Pereira F, Baker CI. Revealing the multidimensional mental representations of natural objects underlying human similarity judgements. Nat Hum Behav. 2020;4(11):1173–1185.

14. Hebart MN, Contier O, Teichmann L, Rockter AH, Zheng CY, Kidder A, et al. THINGS-data, a multimodal collection of large-scale datasets for investigating object representations in human brain and behavior. eLife. 2023;12:e82580.

15. Siegle JH, Jia X, Durand S, Gale S, Bennett C, Graddis N, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. Nature. 2021;592(7852):86–92.

16. De Vries SE, Siegle JH, Koch C. Sharing neurophysiology data from the Allen Brain Observatory. Elife. 2023;12:e85550.

17. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016; p. 770–778.

18. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: arXiv; 2015.Available from: `https://doi.org/10.48550/arXiv.1409.1556`.

19. Roads BD, Love BC. Enriching ImageNet with Human Similarity Judgments and Psychological Embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021. p. 3547–3557.

20. Kriegeskorte N, Kievit RA. Representational geometry: integrating cognition, computation, and the brain. Trends in Cognitive Sciences. 2013;17(8):401–412.

21. Nili H, Wingfield C, Walther A, Su L, Marslen-Wilson W, Kriegeskorte N. A Toolbox for Representational Similarity Analysis. PLOS Computational Biology. 2014;10(4):1–11.

22. Schütt HH, Kipnis AD, Diedrichsen J, Kriegeskorte N. Statistical inference on representational geometries. eLife. 2023;12:e82566.

23. Peyré G, Cuturi M. Computational Optimal Transport: With Applications to Data Science. Found Trends Mach Learn. 2019;11(5–6):355–607.

24. Yamins DL, Hong H, Cadieu CF, Solomon EA, Seibert D, DiCarlo JJ. Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the national academy of sciences. 2014;111(23):8619–8624.

25. Kell AJE, Yamins DLK, Shook EN, Norman-Haignere SV, McDermott JH. A Task-Optimized Neural Network Replicates Human Auditory Behavior, Predicts Brain Responses, and Reveals a Cortical Processing Hierarchy. Neuron. 2018;98(3):630–644.e16.

26. Zhuang C, Yan S, Nayebi A, Schrimpf M, Frank MC, DiCarlo JJ, et al. Unsupervised neural network models of the ventral visual stream. Proceedings of the National Academy of Sciences. 2021;118(3):e2014196118.

27. Rajalingham R, Issa EB, Bashivan P, Kar K, Schmidt K, DiCarlo JJ. Large-Scale, High-Resolution Comparison of the Core Visual Object Recognition Behavior of Humans, Monkeys, and State-of-the-Art Deep Artificial Neural Networks. Journal of Neuroscience. 2018;38(33):7255–7269.

28. Konkle T, Alvarez GA. A self-supervised domain-general learning framework for human ventral stream representation. Nat Commun. 2022;13(1):1–12.

29. Conwell C, Prince JS, Kay KN, Alvarez GA, Konkle T. What can 1.8 billion regressions tell us about the pressures shaping high-level visual representation in brains and machines? bioRxiv. 2023; p. 2022.03.28.485868.

30. Kriegeskorte N, Mur M, Ruff DA, Kiani R, Bodurka J, Esteky H, et al. Matching Categorical Object Representations in Inferior Temporal Cortex of Man and Monkey. Neuron. 2008;60(6):1126–1141.

31. Mur M, Meys M, Bodurka J, Goebel R, Bandettini P, Kriegeskorte N. Human Object-Similarity Judgments Reflect and Transcend the Primate-IT Object Representation. Frontiers in Psychology. 2013;4.

32. Chikazoe J, Lee DH, Kriegeskorte N, Anderson AK. Population coding of affect across stimuli, modalities and individuals. Nat Neurosci. 2014;17(8):1114–1122.

33. Horikawa T, Cowen AS, Keltner D, Kamitani Y. The Neural Representation of Visually Evoked Emotion Is High-Dimensional, Categorical, and Distributed across Transmodal Brain Regions. iScience. 2020;23(5):101060.

34. Koide-Majima N, Nakai T, Nishimoto S. Distinct dimensions of emotion in the human brain and their representation on the cortical surface. Neuroimage. 2020;222:117258.

35. Sagar V, Shanahan LK, Zelano CM, Gottfried JA, Kahnt T. High-precision mapping reveals the structure of odor coding in the human brain. Nat Neurosci. 2023;26(9):1595–1602.

36. Gower JC, Dijksterhuis GB. Procrustes Problems. OUP Oxford; 2004.

37. Williams AH, Kunz E, Kornblith S, Linderman S. Generalized shape metrics on neural representations. Advances in Neural Information Processing Systems. 2021;34:4738–4750.

38. Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for Hyper-Parameter Optimization. In: Advances in Neural Information Processing Systems; 2011.Available from: `https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf`.

39. Bergstra J, Yamins D, Cox D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In: Proceedings of the 30th International Conference on Machine Learning; 2013. p. 115–123. Available from: `https://proceedings.mlr.press/v28/bergstra13.html`.

40. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009. p. 248—255.

Supporting information

General problem setting of alignment

We consider here the problem of aligning two sets of point clouds X and Y (Figs. 1b and c). As detailed above, these points, for instance, represent neural responses, behavioral responses, or activity in a neural network model. We commonly call these points "embeddings". X and Y are $d \times n$ matrices, where n is the number of embeddings and d is the dimension of embedding vectors.

$$X = \begin{pmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n \end{pmatrix}, \; Y = \begin{pmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_n \end{pmatrix}, \tag{3}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are column vectors, which are the ith embeddings of X and Y, respectively.

The general problem setting in this study is to find the optimal alignment between X and Y without assuming any correspondence between the columns (the embeddings) of X and Y. For example, we may sometimes know that $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ are the neural responses to the same external stimulus, suggesting that the i-th column of X corresponds to the j-th column of Y. Even in this case, we do not use this information in the general unsupervised alignment setting.

As a general problem, we consider solving the following problem:

$$\min_P \min_Q \|X - QYP\|_F^2, \tag{4}$$

where $\|\cdot\|_F$ is the Frobenius norm $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$, P is the $n \times n$ assignment matrix that establishes correspondence between the column vectors of X and those of Y (i.e., $\boldsymbol{x}_j \leftarrow \sum_i P_{ij} \boldsymbol{y}_i$), and Q is the $d \times d$ orthogonal matrix that rotates Y to fit into X. If we only allow one element in each column of P to be 1 for each row and set the other elements to 0, the problem becomes finding a one-to-one correspondence between the columns (embeddings) of X and Y, or equivalently, finding the optimal permutation of the column indexes of X. In this study, we examine a more general scenario where the elements of matrix P can take on a real number between 0 and 1. These values represent the degree of correspondence between the i-th column of matrix X and the j-th column of matrix Y. This more flexible approach allows us to model the correspondences between the columns (embeddings) of X and Y in a more comprehensive way.

There are two broad categories of methods for this general problem, supervised alignment and unsupervised alignment. These are explained in detail in the following sections.

Supervised alignment

Supervised alignment is a method in which the assignment matrix P is given. In the case of the optimization problem in Eq. 4, it becomes the well-known Procrustes problem [36], which has a closed form solution. For instance, if we simply assume that the column indexes of X match those of Y, and therefore P is the identity matrix, the optimization problem is given by

$$\min_Q \|X - QY\|_F^2. \tag{5}$$

Given the singular value decomposition $U\Sigma V^\top$ of $XY^\top$, the solution to the Procrustes problem is given by $Q^* = UV^\top$.

Most neuroscience studies to date have only considered supervised alignment. The most widely and commonly used method is Representational Similarity Analysis (RSA) [20–22]. In the conventional framework of RSA, a simple correlation between the distance matrices of X and Y is computed. In this sense, this is technically not the same as the supervised alignment considered in this study (e.g., Procrustes alignment), but we can categorize the conventional RSA as supervised alignment in the sense that it assumes the correspondences between the columns (embeddings) of X and Y. See [37] for a comprehensive list of dissimilarity metrics categorized into supervised alignment and a mathematical characterization of the metrics.

Unsupervised alignment

Unsupervised alignment is the method in which the assignment matrix P is not given. In this case, we need to jointly optimize P and Q in Eq. 4, which is a non-convex optimization problem without a closed-form solution. To address this, we first find an optimal assignment matrix P using Gromov-Wasserstein optimal transport (GWOT) in an unsupervised manner. We then compute the Procrustes solution Q^* based on the assignment matrix obtained from the GWOT analysis. Denoting the optimal transportation plan (the assignment matrix) by Γ^*, the problem to solve becomes

$$\min_Q \|X - QY\Gamma^*\|_F^2. \tag{6}$$

The solution can be found by the singular value decomposition of $X(Y\Gamma^*)^\top$.

Unsupervised alignment in embedded space

With the optimized transportation plan Γ found by GWOT, the embeddings of Y are mapped to the embeddings of X as follows

$$\boldsymbol{x}_j \leftarrow \sum_{i=1}^{n} \Gamma_{ij} \boldsymbol{y}_i. \tag{7}$$

This mapping is then used to find the rotation matrix Q in Eq. 6.

Details of code and implementation

Parameters for the optimazation of GWOT

Hyperparameter tuning of ε To tune the hyperparameter ε values, the range of ε values, how many trials ε values are sampled, and the sampling method need to be set. Even if ε is set to have a wide range, the algorithm would find good epsilon values if it is run over many trials. In general, however, before performing an extensive search, it is a good practice to narrow down the range after a relatively small number of test runs.

As for sampling ϵ, the following three sampling methods, originally implemented by Optuna, are available in the toolbox. First, `TPESampler`, an efficient sampler based on Bayesian sampling, is generally recommended for most cases, and the availability of such efficient sampling methods is indeed the reason why Optuna is used in the toolbox. `TPESampler` is the default sampler in Optuna, which has been shown to perform well in many settings (see `https://github.com/optuna/optuna/issues/2964` for details of the benchmark experiment). The other two samplers are simple grid search and random search samplers. `TPESampler` will be more effective than the other two samplers in our toolbox in most cases because it attempts to find more likely good ranges of ϵ values

based on the results of past trials, while the other samplers sample ϵ values independently of past history. See [11] for the details of each sampler.

`TPESampler` (recommended): A sampler that uses Tree-structured Parzen Estimator (TPE) based on Bayesian optimization. Given the past history of the values of the objective function y (GWD in our case) under hyperparameter values x, the estimator attempts to maximize the expected improvement of y by modeling the conditional probability, $p(x|y)$, based on kernel density estimation. See [38], [39] for mathematical details.

`grid`: A grid search sampler that exhaustively samples ε values on a user-specified grid in a one-dimensional ε space.

`random`: A sampler that randomly and independently samples ε values from the user-defined range of ε values.

Initialization of transportation plans Γ Several initialization methods for transportation plans are available. All of them ensure that the initialized transportation plans Γ satisfy the mass conservation constraint, i.e., the marginals of the transportation plans are equal to $\boldsymbol{p}$ and $\boldsymbol{q}$. One method, `random`, is to initialize the transportation plans randomly and independently on each trial, while the others are fixed initializations over all trials. In general, trying many different initializations is more likely to find better local minima than trying one fixed initialization. Thus, we recommend the `random` option over fixed initializations for most use cases. The fixed initialization methods such as `uniform` should be used in conjunction with random initializations to obtain better optimization results. See Supporting Information for details.

The `diag` option or the `user_define` option should be used with care in unsupervised alignment to avoid "cheating". In the strictest sense, unsupervised alignment is a problem setting where alignment is searched without relying on known or presumed correspondences. In this strict problem setting, use of the `diag` option or `user_define` option only is considered as a cheat option because these options inherently tend to find local minima with a high matching rate in unsupervised alignment. When using these options, the following must be done. First, the `random` option should also be used, and the local minima found by the `random` option should be fairly compared with those found by some given correspondences. If the `random` option finds local minima with lower GWD but a lower matching rate, users must consider these local minima as the optimal solutions found by the GWOT algorithm instead of those found by `user_define` or `diag` options. Second, the fact that some correspondences are given should be explicitly stated to distinguish results with some known information from those without any given information.

Note, however, that the use of the `diag` or `user_define` option is justified or even preferred in problem settings other than the strict unsupervised alignment. For example, in the semi-supervised alignment problem setting, where some of the correspondences are known but the others are not, providing the known correspondences as an initialization will be a natural choice.

`random` (recommended): This method initializes each element of the transportation matrix by sampling from a uniform distribution $[0, 1]$, and then normalizing it to satisfy the mass conservation constraint. The number of initializations on the same trial with the same ϵ value can also be specified. Note, however, that even if the number of initializations on the same ϵ value is set to 1, if many different ϵ values are sampled, many different initializations will be performed on similar ϵ values when `TPESampler` is used. This is because `TPESampler` will extensively search for likely good ϵ values that will end up with similar ϵ values in later trials.

Thus, increasing the number of initializations is similar to increasing the number of ϵ samples when using `TPESampler`.

`uniform`: This method initiallizes a transportation plan by taking the direct product of the marginals $\boldsymbol{p}$ and $\boldsymbol{q}$, i.e., $\boldsymbol{p} \otimes \boldsymbol{q}$. When both $\boldsymbol{p}$ and $\boldsymbol{q}$ are set to a uniform distribution, this method generates a uniform transportation plan consisting of uniform values, whereas if $\boldsymbol{p}$ or $\boldsymbol{q}$ is not a uniform distribution, the intitalized transportation plan is not a uniform matrix. However, for ease of understanding, we simply call this option `uniform`, assuming that $\boldsymbol{p}$ and $\boldsymbol{q}$ are set to a uniform distribution for most use cases. This method is the same as the default initialization implemented in POT.

`diag` (use with caution): This method uses a diagonal matrix with constant values as initialization. This is intended only for the special case where there is a known one-to-one correspondence between the indices of the dissimilarity matrices and these indices are sorted in the same order. Other more general correspondence, such as one-to-many or many-to-many correspondence, should be provided by using the `user_define` option.

`user_define` (use with caution): This method initializes a transportation plan with a user-defined matrix. As in the case of "diag", it is intended for this method that the user provide some known or presumed correspondences.

Evaluation of the results

Matching rate of unsupervised alignment

`AlignRepresentations.calc_accuracy(eval_type)`: If there is a grand truth mapping between two sets of embeddings or if there is some known correspondence by which the user wants to evaluate the quality of the unsupervised alignment, it is recommended that the correct matching rate between the empirically found correspondence and the known correspondence be computed. There are two ways to compute the correct matching rate, one based on optimal transportation plans and the second on aligned embeddings. To define the correct matching rate, we define the "correct" assignment matrix C as $C_{ij} = 1$ if i and j are matched pairs and $C_{ij} = 0$ if otherwise.

Based on optimal transportation plan `eval_type="ot_plan"`: By using the correct assignment matrix C, the correct matching rate based on the optimal transportation plan is defined as follows. When i and j are a correct matched pair, the following function checks if the element of the transportation matrix, Γ_{ij} is the maximum among the other elements $\Gamma_{ij'}$ in the same row,

$$\text{Match}(i) = \begin{cases} 1, & \text{if } \Gamma_{ij} = \max_{j'}(\Gamma_{ij'}) \text{ and } C_{ij} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The matching rate is then the percentage of index i that matches with the correct pair j, which can be calculated as:

$$\text{Matching rate} = \frac{\sum_{i=1}^{n} \text{Match}(i)}{n}. \tag{9}$$

The matching rate defined above is the top 1 matching rate. More generally, the top k matching rate is similarly computed by checking whether Γ_{ij} is within the top-k highest values among the other elements $\Gamma_{ij'}$ when $C_{ij} = 1$.

Based on aligned embeddings `eval_type="k_nearest"`: The matching rate can also be evaluated based on aligned embeddings as follows. First, by using the optimized transportation plan Γ^*, the rotation matrix Q^* is computed by solving Eq. 6. The aligned embeddings, X and Q^*Y, are then obtained. Given the two sets of embeddings X and Q^*Y, the distances (e.g., Euclidean distance or cosine distance) between all pairs of the embeddings of X and Q^*Y are computed. If the distance matrix between X and Q^*Y is denoted by Λ, the k-nearest neighbor matching rate can be computed by checking whether Λ_{ij} is within the top k smallest values among the other elements $\Lambda_{ij'}$ in the same row i when $C_{ij} = 1$.

Inspection of other local minima
`AlignRepresentations.show_optimization_log`: It is crucial and highly recommended to not only focus on the best solution with the minimum GWD value, but also to examine the other local minima with high GWD values. First, it is always helpful to plot the relationship between GWD and ϵ to investigate how GWD values depend on ϵ and how close or far the best solution with the minimum GWD value is from other local minima. Also, when the matching rate is computed, it is also informative to plot the relationship between GWD and the matching rate to understand whether a high matching rate is achievable in certain applications. For example, if it is observed that lower GWD values tend to result in higher matching rates, then a high matching rate is achievable.

Visualization of aligned embeddings
`AlignRepresentations.visualize_embedding`: It is helpful to visually inspect the quality of the unsupervised alignment by plotting the aligned embeddings in 2D or 3D space. Using the rotation matrix Q^* in Eq. 6, one set of embeddings X, and the other set of aligned embeddings Q^*Y, are obtained for the plot, where X are the pivot embeddings to which the embeddings of Y are aligned. Then, for example, Principle Components Analysis (PCA) is used to project high-dimensional embeddings into 2D or 3D space, but other dimensionality reduction methods are also applicable. If there is a known correspondence between the embeddings, the user can visually check whether the aligned embeddings are positioned close to the corresponding embeddings.

Data and pre-processing

THINGS data

In the THINGS dataset, participants performed an odd-one-out task, where they were presented with three naturalistic objects from the THINGS dataset and asked to report which item in the triplet was the most dissimilar to the other two objects. This dataset includes approximately 4.70 million similarity judgments from about 12,000 participants collected through online crowdsourcing. By randomly subsampling the participants and making two non-overlapping participant groups, we consider the alignment between the two participant groups. Each participant group contains ~ 800 thousand similarity judgments, a sample size proven to be sufficient for estimating meaningful and consistent representations of natural objects [13, 14].

After making the two participant groups, we next need to estimate the embeddings of the 1,854 natural objects from the odd-one-out judgment data. We followed the procedure of previous studies to estimate the embeddings [6, 13, 14]. See Supporting information for details.

First, the embeddings of the 1,854 objects were initialized with 90 randomly assigned dimensions ranging from 0 to 1. Second, the Euclidean distance between all pairs of the

embedding vectors was computed and considered as the dissimilarity between the embeddings,

$$D_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2, \tag{10}$$

where $\|\cdot\|_2$ is the L2 norm $\|\boldsymbol{z}\|_2 = \sqrt{\sum_i z_i^2}$. Conversely, the similarity between the embeddings was quantified as the negative Euclidean distance,

$$S_{ij} = -D_{ij}. \tag{11}$$

Third, using the similarity between the embeddings, we estimated the probability that a participant chooses image k as an odd object among the triplet (i, j, k), which is equivalent to the probability of choosing image i and j as the most similar object pair among three possible pairs. Here, the probability was estimated by the softmax function of the similarity between the embeddings of the pair (i, j),

$$\Pr[i, j] = \frac{\exp(S_{ij})}{\exp(S_{ij}) + \exp(S_{jk}) + \exp(S_{kl})}, \tag{12}$$

where S_{ij} is given by Eq. 11. Fourth, we updated the embeddings by minimizing the following loss function. For the l-th triplet in the dataset, let $(c_l^{(1)}, c_l^{(2)})$ denote the index pair chosen by a participant as the most similar pair. Then, the loss function is given by

$$L = -\sum_{l=1}^{n_{\text{train}}} \log \Pr[c_l^{(1)}, c_l^{(2)}] + \lambda \sum_{i=1}^{m} \|\boldsymbol{x}_i\|_1, \tag{13}$$

where n_{train} is the total number of triplets in the training dataset, m is the number of the natural objects, and $\|\cdot\|_1$ denotes the L1 norm $\|\boldsymbol{z}\|_1 = \sum_i |z_i|$. The first term is the cross-entropy loss and the second term is the L1 norm regularization with the hyperparameter λ. The loss function was optimized by the Adam algorithm with an initial learning rate of 0.001, using a fixed number of 1,000 epochs. The hyperparameter λ was optimized by 5-fold cross-validation.

Neuropixels visual coding from the Allen Brain Observatory

The Neuropixels Visual Coding dataset consists of large-scale electrophysiological recordings in the mouse visual system including multiple cortical areas [15]. Among the recorded areas, we chose the primary visual cortex (VISp) as an example. The number of neurons recorded in VISp is about 60, depending on the subject. As for visual stimuli, we chose a natural movie stimulus ("natural movie one"), which is a 30-second scence from a black and white movie, as an example.

Here, we considered the alignment of the aggregated neural responses across multiple mice, rather than the alignment of neural responses between individual mice. This is because we found that the alignment of individual mice was impossible for any of several possible reasons, such as the limited number of recorded neurons, fluctuation or noise in neural responses, and individual differences. Specifically, we aggregated the neural responses of 14 mice, corresponding to the responses of approximately 900 neurons in total, and considered them to be the neural responses of a "pseudo"-mouse. Similarly, we also made the aggregated neural responses from 14 different mice and considered it as another "pseudo"-mouse. We then considered the alignment between the two pseudo-mice created in this way.

For the alignment, we considered the similarity structures of neural responses to different short movie stimuli. Although the movie stimulus we used ("natural movie one") is a 30-second continuous movie stimulus, we segmented it into 1-second short movie stimuli and treated them as 30 segmented movie stimuli. We then computed the trial average of the spike counts during each 1-second short movie stimulus and considered these as the neural responses of the pseudo-mice to be aligned.

Vision DNN

To demonstrate the alignment of the internal representations of neural network models, we considered the alignment of vision Deep Neural Network Models (DNNs) as a typical case study. Specifically, we used the pre-trained models of ResNet50 [17] and VGG19 [18] in PyTorch. We performed unsupervised alignment between these models.

As an initial step, we extracted the embeddings from the last fully connected layers in the two DNNs, whose dimensions are 1,000. As input images, we used the validation set in the ImageNet dataset [40], which contains 50,000 natural images belonging to 1,000 classes. For ease of computation, we used only 1,000 images belonging to 20 classes, where each class contains 50 images.