Unsupervised Learning for Optimal Transport plan prediction between unbalanced graphs

Sonia Mazelet
CMAP, Ecole Polytechnique
Palaiseau, France
sonia.mazelet@polytechnique.edu

Rémi Flamary
CMAP, Ecole Polytechnique
Palaiseau, France
remi.flamary@polytechnique.edu

Bertrand Thirion
Mind, Inria-Saclay
Palaiseau, France
bertrand.thirion@inria.fr

Abstract

Optimal transport between graphs, based on Gromov-Wasserstein and other extensions, is a powerful tool for comparing and aligning graph structures. However, solving the associated non-convex optimization problems is computationally expensive, which limits the scalability of these methods to large graphs. In this work, we present Unbalanced Learning of Optimal Transport (ULOT), a deep learning method that predicts optimal transport plans between two graphs. Our method is trained by minimizing the fused unbalanced Gromov-Wasserstein (FUGW) loss. We propose a novel neural architecture with cross-attention that is conditioned on the FUGW tradeoff hyperparameters. We evaluate ULOT on synthetic stochastic block model (SBM) graphs and on real cortical surface data obtained from fMRI. ULOT predicts transport plans with competitive loss up to two orders of magnitude faster than classical solvers. Furthermore, the predicted plan can be used as a warm start for classical solvers to accelerate their convergence. Finally, the predicted transport plan is fully differentiable with respect to the graph inputs and FUGW hyperparameters, enabling the optimization of functionals of the ULOT plan.

1 Introduction

Graph alignment In many graph data applications, aligning or matching nodes between two graphs is necessary. Examples include object detection, where semantic correspondences between objects in two images from different domains can make the model more adaptive [12]; graph edit distance [22, 18], where the goal is to compute the distance between two graphs, which requires finding the best correspondence between their nodes; shape matching [20], or brain alignment across subjects [30].
But the problem of graph matching is challenging because of the combinatorial nature of the problem, which can often be reformulated as a Quadratic Assignment Problem (QAP) known to be NP-hard [16]. One strategy that has been proposed recently is to use deep learning to learn the matching in a supervised setting [36], [25]. These methods typically rely on a graph neural network (GNN) to learn a representation of the nodes and then use a matching algorithm to compute the correspondence between the nodes. However, the problem is even more difficult when the graphs are unbalanced, i.e. when they have different numbers of nodes or when some nodes have noisy features or connections.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Optimal transport between graphs In recent years, Optimal Transport (OT) has emerged as a powerful tool for solving the graph matching problem. It can be seen as a continuous relaxation of the QAP with the Gromov-Wasserstein (GW) distance [17, 21, 40], which is a generalization of the classical Wasserstein distance to distributions in different metric spaces. Extensions of the GW distance to labeled graphs have been proposed, such as the Fused Gromov-Wasserstein (FGW) distance [31]. But those OT problems put strong constraints on the transport plan, which makes them very sensitive to noise, outliers and local deformations. This is why Unbalanced GW [27, 5] and Fused Unbalanced GW (FUGW) [30] were proposed to generalize the GW and FGW distances to unbalanced settings with application in positive unlabeled learning and brain alignment.

Complexity of classical optimal transport solvers Quadratic OT problems such as FGW and FUGW are non-convex. Classical solvers rely on a block coordinate descent algorithm to iteratively solve linearized versions of the problem. This linearization is of complexity $O(n_1 n_2^2 + n_2^2 n_1)$, where n_1 and n_2 are the number of nodes in the two graphs [21]. This makes the method unscalable for large graphs, for applications where we need to compute the transport plan for many pairs of graphs or when validation of the hyperparameters is necessary. This is especially problematic in the unbalanced case where the solution is very sensitive to the parameters.
The ML community has recently proposed to use deep learning for accelerating or solving OT problems. For example [11] proposed to estimate the OT mapping using a neural network and extensions to the GW have been proposed in [19, 38]. But the most relevant work for our purpose is Meta OT [2] which proposed to learn to predict the dual potentials of the entropic OT problem with a neural network. But this approach is limited to classical entropy regularized OT and not directly applicable to the quadratic FUGW problem which motivates our proposed approach detailed below.

Contributions We propose in this paper a new method to learn a neural network that predicts the transport plan of the FUGW problem denoted as Unsupervised Learning of Optimal Transport plan prediction (ULOT). We propose a novel architecture based on graph neural networks (GNN) and cross attention mechanisms to predict the OT plan with a complexity of $O(n_1 n_2)$, which is significantly faster than classical solvers. In addition the neural network is conditioned by the parameters of the FUGW problem, which allows to efficiently predict OT plan for all the possible values of the parameters. This is particularly useful in the unbalanced case where the solution is very sensitive to the parameters and validation is often necessary. We show in our experiments, on simulated and real life data, that our method outperforms classical solvers in terms of speed by two orders of magnitude while providing OT plans with competitive loss. The predicted transport plan can also be used as a warm start for classical solvers, which reduces the number of iterations and the overall time of the algorithm. We also show that our method provides a smooth estimation of the transport plan that can be used for numerous applications such as parameter validation for label propagation or gradient descent of a FUGW loss. Our code is available at `https://github.com/smazelet/ULOT`

2 Learning to predict OT plans between graphs

In this section we first introduce the FUGW optimal transport problem and its associated loss function. Next we present our amortized optimization strategy called Unbalanced Learning of Optimal Transport (ULOT) and detail the architecture of the neural network. Finally we discuss the related works in deep learning for optimal transport and graph matching.

2.1 Fused Unbalanced Gromov Wasserstein (FUGW)

Definition of the FUGW loss Consider the two graphs $\{G_k = (\boldsymbol{F}_k, \boldsymbol{D}_k, \boldsymbol{\omega}_k)\}_{k=\{1,2\}}$ with n_1 and n_2 nodes respectively. For $k \in \{1, 2\}$, they are characterized by their node features $\mathbf{F}_k \in \mathbb{R}^{n_k \times d}$, their connectivity matrices $\boldsymbol{D}_k \in \mathbb{R}^{n_k \times n_k}$ (usually adjacency matrix or shortest path distance matrix) and their node weights $\boldsymbol{\omega}_k \in \Delta_{n_k} \triangleq \{(\omega_k^1, ..., \omega_k^{n_k}),\ \sum_{i=1}^{n_k} \omega_k^i = 1\}$ that characterize the node's relative importance [34]. Note that in the following we will assume that these weights are uniform, i.e. $\omega_k^i = 1/n_k$ for $i = 1, ..., n_k$. The goal of FUGW [30] is to learn a positive transport plan $\boldsymbol{P} \in \mathbb{R}^{n_1, n_2}$

between the nodes of G_1 and G_2 that minimizes the following loss function:

$$\mathrm{L}^{\alpha,\rho}(G_1, G_2, \boldsymbol{P}) = (1-\alpha) \sum_{i,j=1}^{n_1,n_2} \left\| (\boldsymbol{F}_1)_i - (\boldsymbol{F}_2)_j \right\|_2^2 \boldsymbol{P}_{i,j} \tag{1}$$

$$+ \alpha \sum_{i,j,k,l=1}^{n_1,n_2,n_1,n_2} |\, (\boldsymbol{D}_1)_{i,k} - (\boldsymbol{D}_2)_{j,l} \,|^2 \boldsymbol{P}_{i,j} \boldsymbol{P}_{k,l} \tag{2}$$

$$+ \rho \left(\mathrm{KL}(\boldsymbol{P}_{\#1} \otimes \boldsymbol{P}_{\#1} | \boldsymbol{\omega}_1 \otimes \boldsymbol{\omega}_1) + \mathrm{KL}(\boldsymbol{P}_{\#2} \otimes \boldsymbol{P}_{\#2} | \boldsymbol{\omega}_2 \otimes \boldsymbol{\omega}_2) \right). \tag{3}$$

The FUGW loss is a combination of the Wasserstein distance (1) that measures the preservation of node features, the Gromov Wasserstein distance (2) that measures the conservation of local geometries and a penalization of the violation of the marginals for the OT plan with the KL divergence (3). The terms (1) and (2) are weighed by the trade-off parameters $\alpha \in [0, 1]$ and the marginal penalization (3) is weighted by ρ. Unbalanced OT is a very general and robust framework that can adapt to differences in the geometry of the graph vertices.

Complexity of solving $\min_{\boldsymbol{P} \geq 0} \mathbf{L}^{\alpha,\rho}(G_1, G_2, \boldsymbol{P})$ In order to solve the FUGW problem one needs to minimize the FUGW loss with respect to the optimal transport plan $\boldsymbol{P}$. Because of the Gromov Wasserstein term, the complexity of the FUGW loss or its gradient for a given plan $\boldsymbol{P}$ is theoretically quartic $O(n_1^2 n_2^2)$. In the case of the square loss, Peyré et al [21] showed that the complexity can be reduced to cubic complexity $O(n_1 n_2^2 + n_1^2 n_2)$, which remains computationally intensive for large graphs with typically more than $10k$ nodes.
Existing methods to minimize the FUGW loss use a block coordinate descent scheme on a lower bound of the objective that consists in solving at each iteration a linearization of the quadratic problem [8], [30] and requires at each iteration to compute the cubic gradient. In the following we will compare ULOT to three different types of inner-solvers for the linearized problem: the Majorization-minimization (MM) algorithm [6], the inexact Bregman Proximal Point (IBPP) algorithm [39] and a more classical L-BFGS-B algorithm [4]. We will also compare our approach to the entropic regularization of the FUGW loss for which the inner problem can be solved using the Sinkhorn algorithm. All those methods are iterative and require a $O(n_1 n_2^2 + n_1^2 n_2)$ gradient/linearization computation at each iteration. This is a major bottleneck for the FUGW problem, especially when the graphs are large or FUGW has to be solved multiple times, for instance when computing a FUGW barycenter while selecting the parameters (α, ρ) of the method.

2.2 ULOT optimization problem and architecture

Learning to predict FUGW OT plans We propose to train a model $\boldsymbol{P}_\theta^{\rho,\alpha}(G_1, G_2)$, parametrized by θ, that, given two graphs G_1, G_2 and (ρ, α) parameters can predict a FUGW transport plan, or at least a good solution to the FUGW problem, between them. We want to avoid training the model in a supervised way where for each pair of graphs (G_1, G_2) in the training set we need to pre-compute the corresponding transport plan $\boldsymbol{P}$. This is why we propose to train the model in an unsupervised way using amortized optimization [3]. We do this by sampling pairs of graphs (G_1, G_2) from a training dataset $\mathcal{D}$ and parameters (ρ, α) and minimizing the expected FUGW loss. The ULOT model is trained to minimize:

$$\min_\theta \quad \mathbb{E}_{G_1, G_2 \sim \mathcal{D}^2, \alpha, \rho \sim \mathcal{P}} \left[\mathrm{L}^{\alpha,\rho}(G_1, G_2, \boldsymbol{P}_\theta^{\rho,\alpha}(G_1, G_2)) \right]. \tag{4}$$

Where $\mathcal{P}$ is the distribution of the parameters (ρ, α) chosen in the experiment as $\mathcal{P}_\rho, \mathcal{P}_\alpha$, where $\mathcal{P}_\rho$ is the log uniform distribution between 10^{-7} and 1, and $\mathcal{P}_\alpha$ follows the Beta distribution with parameters $(0.5, 0.5)$. The fact that we optimize the loss over some intervals of the parameters (ρ, α) allows the model to generalize to different values of the parameters to explore or even optimize them.

Encoding the parameters (ρ, α) Since we want the model $\boldsymbol{P}_\theta^{\rho,\alpha}(G_1, G_2)$ to depend on the parameters (ρ, α), we need to encode them in a way that can be used by the neural network. To do that, we add them to the node features of the graphs at each layer of the network. ρ is a positive scalar that impacts the mass of the OT plan and can be included as such. But α is a scalar in $[0, 1]$ fixing the tradeoff between the Wasserstein and Gromov-Wasserstein terms that can have a large impact close to 0 and 1. In order to facilitate its use in the neural network, we propose to encode it in a more

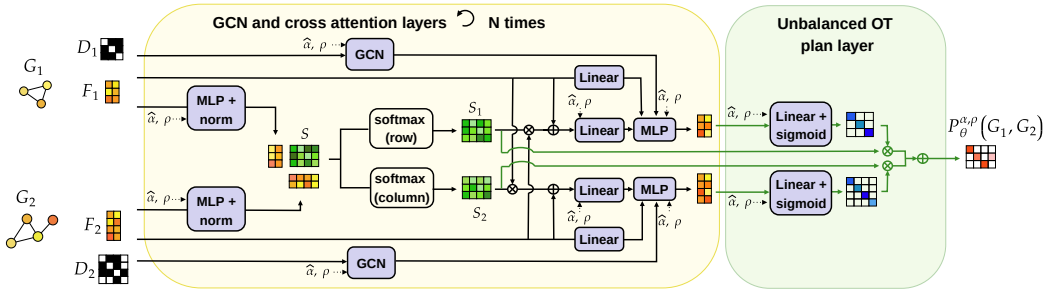

Figure 1: **ULOT architecture for OT plan prediction** The architecture consists of two parts. a node embedding layer repeated N times that relies on cross graph attention and self node updates (GCN), and the final transport plan prediction layer that predicts node weights and the output transport plan.

expressive way with a positional encoding technique. We chose to use the same technique as in [32] using the Fourier basis for encoding time in flow matching applications as follows:

$$\hat{\alpha} = \left[(\cos(k\pi\alpha))_{k=1,\dots,d} | (\sin k\pi(1-\alpha))_{k=1,\dots,d} \right]. \tag{5}$$

where we set $d = 10$ in our experiments. $\hat{\alpha}$ and ρ are concatenated to the node features at each layer of the network as detailed next.

2.3 Proposed cross-attention neural architecture

We now detail the proposed neural network architecture that takes as input two graphs G_1 and G_2 and the parameters (ρ, α) and predicts the transport plan $\boldsymbol{P}_\theta^{\rho,\alpha}(G_1, G_2)$. The proposed ULOT architecture summarized in Figure 1 consists of two main parts. The first part contains N layers of node embedding and attention-based cross-graph interactions that incorporate the geometry of each graph. The final layer predicts the transport plan from the learned node features and interactions with node reweighting. The design choices are further motivated in the ablation studies in Section C.

Node embedding with cross attention The first block is an adaptation of the Graph Matching Network from [13] that outputs node features that are relevant in a matching context. Intuitively the cross attention mechanism computes node features that are similar for nodes that can be matched and dissimilar for nodes that should not. The node embeddings are computed at each layer using two paths. One path is called the *self* path that simply consists of a GCN treating each graph independently. The use of GCNs is motivated by the Gromov term in the FUGW loss that requires a model taking into account the graph topology. For each graph G_k for $k \in \{1, 2\}$, *self* node features are computed with

$$\boldsymbol{F}_k^{\text{self}} = \text{GCN}(\boldsymbol{F}_k). \tag{6}$$

The second path is the *cross* path that computes a similarity matrix between the node features of both graphs and learns new features that characterize their interactions. In parallel, *cross* node features $\boldsymbol{F}_{1\to2}^{\text{cross}}, \boldsymbol{F}_{2\to1}^{\text{cross}}$ are computed with an attention block, for $k, k' \in \{1, 2\}, k \neq k'$, learned in the following way:

1. Compute embeddings:
$$\boldsymbol{F}_k^{\text{cross}} = \text{MLP}(\boldsymbol{F}_k, \rho, \hat{\boldsymbol{\alpha}}). \tag{7}$$
2. Compute similarity matrix $\boldsymbol{S}$ and the row/column attention matrices $\boldsymbol{S}_1, \boldsymbol{S}_2$ with, for s the cosine similarity and $i \in [1, n_1]$, $j \in [1, n_2]$:
$$S_{i,j} = s\left((\boldsymbol{F}_1^{\text{cross}})_i, (\boldsymbol{F}_2^{\text{cross}})_j\right) \quad \text{and} \quad \begin{cases} \boldsymbol{S}_1 = \text{softmax}_{\text{row}}(a^2\boldsymbol{S}) \\ \boldsymbol{S}_2 = \text{softmax}_{\text{column}}(a^2\boldsymbol{S}) \end{cases} \tag{8}$$
where $a \in \mathbb{R}$ is an hyperparameter.
3. Compute the updated node features $\boldsymbol{F}_{1\to2}^{\text{cross}}$ and $\boldsymbol{F}_{2\to1}^{\text{cross}}$:
$$\boldsymbol{F}_{1\to2}^{\text{cross}} = \text{Linear}\left(\boldsymbol{F}_2^{\text{cross}} - \boldsymbol{S}_2^T \boldsymbol{F}_1^{\text{cross}}\right), \quad \boldsymbol{F}_{2\to1}^{\text{cross}} = \text{Linear}\left(\boldsymbol{F}_1^{\text{cross}} - \boldsymbol{S}_1 \boldsymbol{F}_2^{\text{cross}}\right), \tag{9}$$

At the end of the layer, for $k, k' \in 1, 2$, the learned node features from the self (GCN) and cross-attention paths are merged with the input node features with the following equation:

$$\boldsymbol{F}_k^{\text{final}} = \text{MLP}\left(\text{Linear}(\boldsymbol{F}_k), \text{Linear}(\boldsymbol{F}_k^{\text{self}}), \text{Linear}(\boldsymbol{F}_{k'\to k}^{\text{match}}), \rho, \hat{\boldsymbol{\alpha}}\right), \tag{10}$$

where the inputs to the MLP are concatenated. The use of those two paths allows to learn node features that take into account both the graph geometry and the cross-interactions between the graphs.

Transport plan prediction with cross attention and node scaling The optimal transport plan prediction block is designed so as to predict an unbalanced OT plan that must contain pairwise relationships between nodes but also that can discard (reweight) nodes that cost too much from an OT perspective. This is done by separating the two aspects. We chose to use for the pairwise relationship in the OT plan the average of the cross attention matrices corresponding to row-wise softmax and a column-wise softmax respectively. This matrix gives a good starting point for the plan, which is then refined in the unbalanced optimal transport plan block, where the scaling per node makes it no longer balanced. This is why we also provide in the last block two node reweighting heads that allow scaling on the OT plan and enable the network to discard mass. These heads predict weight vectors $\boldsymbol{v}_k \in \mathbb{R}^{n_k}, k \in \{1, 2\}$ on the nodes of the graphs that represent how much weight is transported from each node with

$$\boldsymbol{v}_1 = \text{sigmoid}(\text{Linear}(\boldsymbol{F}_1^{\text{final}}, \rho, \widehat{\boldsymbol{\alpha}})), \quad \boldsymbol{v}_2 = \text{sigmoid}(\text{Linear}(\boldsymbol{F}_2^{\text{final}}, \rho, \widehat{\boldsymbol{\alpha}})) \tag{11}$$

We learn the transport plan from the similarity matrices S_1, S_2 of the last layer

$$\boldsymbol{P}_\theta^{\rho,\alpha}(G_1, G_2) = \frac{1}{2}\left(\frac{1}{n_1}\boldsymbol{S}_1 \text{diag}(\boldsymbol{v}_1) + \frac{1}{n_2}\text{diag}(\boldsymbol{v}_2)\boldsymbol{S}_2\right). \tag{12}$$

This allows us to learn a transport plan that is unbalanced and separates the problem of finding the node interactions across graphs, done with the cross attention, and the problem of learning the individual node weights that is specific to unbalanced OT.

2.4 Related works

Deep optimal learning for transport Several methods have been proposed for accelerating the resolution of optimal transport problems with deep learning. For instance [26] proposed to model the dual potentials of the entropic regularized OT problem with a neural network. Neural OT [11] learns the classical OT mapping between two distributions using a neural network. Recent Neural OT extensions have also been proposed to solve the GW problem in [19, 38]. This usually allows for solving large OT problems but the resulting neural network is a solution for a specific pair of distributions and needs to be optimized again for new distribution pairs.

Meta OT [2] uses a strategy called amortized optimization [3] to learn an MLP neural network that predicts on the samples the dual potentials of the entropy regularized OT problem. Their model can be used to predict the entropic transport plan between two new distributions using the primal-dual relationship. However, Meta OT cannot be used for Quadratic OT problems such as the Gromov-Wasserstein or FUGW problems because the optimization problem is not convex and the primal-dual relationship is much more complicated [42]. In fact, in order to use an approach similar to Meta OT, one would need to use a linearization of the problem that is $O(n^3)$ which would cancel part of the advantage of using an efficient neural network.

ULOT has been designed to perform OT plan prediction with $O(n^2)$ complexity. Our approach also focuses on graph data. There is a need to go beyond MLP in order to design a neural network that can use the graph structure. In this sense, ULOT can be seen as a generalization of Meta OT to the case of unbalanced OT between graphs. Finally we learn a model that can predict the transport plan conditioned on the parameters of the OT problem, in our case the (α, ρ), which is particularly novel and has not been done before to the best of our knowledge.

Deep learning for graph matching Various neural architectures have been proposed for deep graph matching, which refers to the problem of finding structural correspondence between graphs. However these methods often face notable limitations. Some approaches are limited to predicting a global similarity score between pairs of graphs without predicting node correspondence [13, 14]. Other methods that do provide node-level matching typically rely on supervised training [36, 25, 41], which limits their applications to domains with available ground truth correspondences such as images.

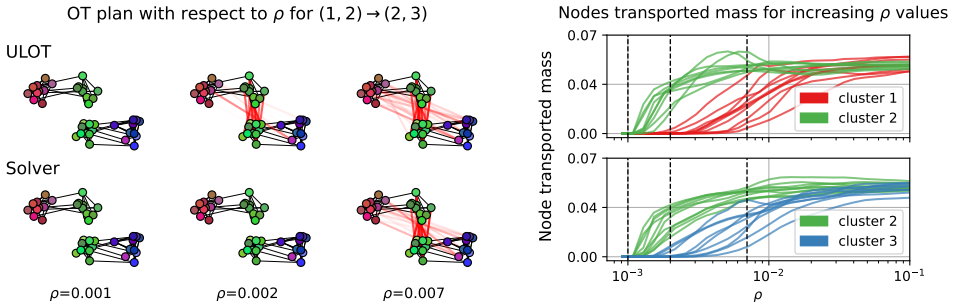

Figure 3: Illustration of OT plans for different values of ρ. (left) OT plan (red lines) predicted by ULOT (top) and estimated by the IBPP solver (bottom). The red lines opacity is proportional to the amount of transported mass. (right) Marginals on the nodes of both graphs for different values of ρ colored by the cluster they belong to.

Unsupervised methods, on the other hand, rely on task-specific objectives such as enforcing cycle consistency [33, 37] or adopt contrastive learning approaches by matching graphs to their augmented copies, for which they know the ground truth [15].
To improve the robustness of the matching between graphs of different sizes, some methods discard a subset of correspondences. Various methods have been proposed such as selecting the top-k most confident matches [35] or adding dummy nodes [10]. However, while these methods have a similar goal to ULOT they perform a hard selection of nodes, which is useful in the case of outliers but not when we need a more continuous way to reduce some node importance. Finally, while our method bears some similarities with neural networks for graph matching, it is important to note that the objective of Unbalanced OT is fundamentally different from graph matching.

3 Numerical experiments

In this section, we evaluate ULOT and compare it to classical solvers on both a simulated dataset of Stochastic Block Models (SBMs) with different numbers of clusters and on the Individual Brain Charting (IBC) dataset [23] of functional MRI activations on brain surfaces.

3.1 Illustration and interpretation on simulated graphs

Dataset and training setup We first train ULOT on a simulated dataset of Stochastic Block Models (SBMs) with 3 linearly connected clusters and 3D node features that are a one hot encoding of the cluster classes $1, 2$ or 3 with additive centered Gaussian noise. To investigate the properties of the transport plans learned by ULOT in comparison to those estimated by classical solvers, we construct three types of graphs with different clusters. The first type includes graphs where all three clusters $(1, 2, 3)$ are present, the second type of graphs has clusters $(1, 2)$ and the third has clusters $(2, 3)$. All graphs in the dataset have a random number of nodes ranging from 30 to 60 and the training dataset consists of 50000 simulated pairs (G_1, G_2, ρ, α).

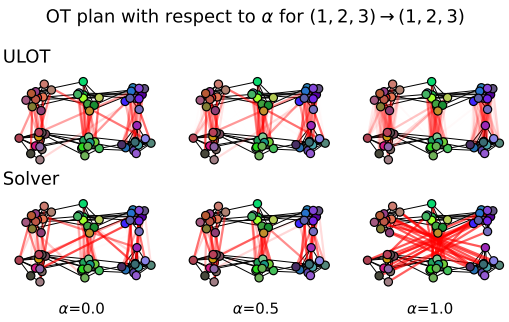

Figure 2: Examples of transport plans (red lines) predicted by ULOT (top) and estimated by the IBPP solver (bottom) for different α values. The red lines opacity is proportional to the amount of transported mass.

With those three types of graphs, unbalanced OT should be able to find a transport plan that matches the clusters if they are present in both graphs, but discard clusters that are not shared (when α, ρ are properly selected).We compare ULOT to the numerical solver IBPP, which provides a good trade off between performance and speed.

Regularization path with respect to the parameters ρ and α We first illustrate the effect of ρ on the predicted transport plan between graphs containing clusters $(1, 2)$ and $(2, 3)$. We see in Figure 3 (left) that the transported mass increases with ρ for both ULOT and the solver but while the solver is stuck in a 0 mass local minimum, ULOT is able to find a plan with mass and lower cost for $\rho = 0.002$. We also provide in Figure 3 (right) the regularization path for the marginals on the nodes of both graphs which shows that the nodes from cluster 2 in green are the first to receive mass when ρ increases finding a proper alignment of the clusters.
Next we illustrate the OT plans between two graphs of type $(1, 2, 3)$ for different values of α in Figure 2. Because the node features are noisy, both the Gromov Wasserstein and Wasserstein terms in the loss are needed to predict accurate plans. We see that while the transport plans from ULOT and the solver are comparable for $\alpha < 1$, the solver wrongly matches the clusters for $\alpha = 1$ because it does not use the node feature information and can permute classes. ULOT does not permute the clusters for $\alpha = 1$ because it is continuous w.r.t. α and has learned to use the node features to find a better plan.

Optimizing the hyperparameter for a prediction task We now consider the task of label propagation between graphs, where node labels are known on a source graph but partly missing on a target graph. To infer the missing labels, we transport the one hot encoding of the source graph node labels onto the target graph, producing label probabilities for each node similarly to what was proposed in [28, 24]. A key challenge of this approach, when the graph types can differ, lies in selecting the appropriate FUGW parameters (ρ, α) to ensure that the plan is relevant for label propagation.
Thanks to our efficient ULOT framework, we can easily compute and visualize the accuracy of the label propagation task as a function of the parameters (ρ, α) for different pairs of graphs. Due to lack of space this is provided in the supplementary material in Figure 11. We find that the accuracy surfaces are relatively smooth and that the optimal parameters greatly depend on the types of graphs transported.
Since ULOT OT plans are by construction fully differentiable with respect to ρ and α, we propose to optimize them, but taking as objective a classical smooth proxy for the accuracy: the Kullback-Leibler divergence between the one-hot encoded target classes and the predicted label scores. We show the optimization trajectories of parameters (ρ, α) for different simulated pairs of graphs in Figure 4. We see that the trajectories vary significantly between the two types of graph pairs. Indeed between different types, ρ has to be small to avoid mass transfer between clusters that are not present in both graphs, while for the same types, ρ can be larger to allow mass transfer between clusters that are present in both graphs. Interestingly, we see that for the same graph pair types, the general trend is similar but the trajectories converge to different values, underlining the necessity of parameter validation for individual pairs. This proof of concept shows that ULOT parameters can be optimized for a given task, allowing for efficient bi-level optimization when the inner optimization problem is a FUGW.

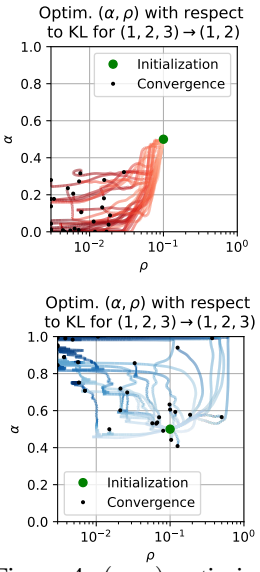

Figure 4: (α, ρ) optimization trajectories for (top) different types $(1, 2, 3) \to (1, 2)$, (bottom) same type $(1, 2, 3)$.

Optimizing a graph wrt the FUGW loss One very interesting aspect of our method is that the predicted transport plan is fully differentiable with respect to the graph structure and features. We illustrate this by optimizing a functional of a graph. Given a target graph $G^\star$, we optimize the function $F(G) = L^{\alpha,\rho}(G, G^\star, \boldsymbol{P}_\theta^{\rho,\alpha}(G, G^\star))$ where we expect the graph G to converge to or close to the graph $G^\star$. Starting from $G_0 = G$, at each time step t, we predict the transport plan $\boldsymbol{P}_\theta^{\rho,\alpha}(G_t, G^\star)$ between G_t and $G^\star$, compute the associated FUGW loss and update the nodes features and shortest path distance matrix of G_t using backpropagation. We recover the adjacency matrix at time step $t + 1$ by thresholding the shortest path distance matrix. The influence of the tradeoff parameters α is shown on Figure 5. The trajectory for $\alpha = 0.5$ shows that the graph converges to a two-cluster graph with proper labels. In contrast, the trajectory for $\alpha = 1$ shows that the graph converges to a two-cluster

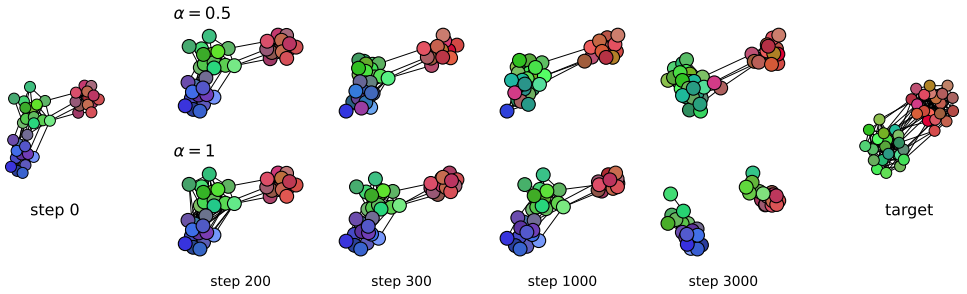

Figure 5: Gradient descent steps of the minimization of of the ULOT FUGW loss between a source graph (step 0) and a target graph for $\alpha = 0.5$ (top) and $\alpha = 1$ (bottom).

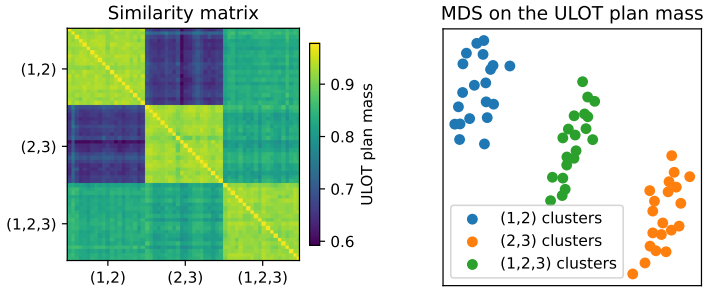

Figure 6: (left) Similarity matrix of the ULOT transport mass between simulated SBM graphs with respective clusters $(1, 2)$, $(2, 3)$ and $(1, 2, 3)$ sorted by type. (left) MDS of the similarity matrix.

graph with wrong labels. This occurs because only the Gromov-Wasserstein term is optimized, so the node features are not updated.

Using the ULOT transport plan mass The FUGW loss computed with the ULOT plan provides a meaningful distance between graphs, but its computation is $O(n_1^2 n_2 + n_2^2 n_1)$, where n_1, n_2 are the number of graph nodes. In contrast, computing the ULOT transport plan only has a quadratic time complexity. In unbalanced OT, the total mass of the OT plan $m(\boldsymbol{P}) = \sum_{i,j} P_{i,j}$ will decrease when the two graphs are very different, since in this case the marginal violation will cost less than the transport cost. This is why we propose to use the ULOT transport plan mass as a graph similarity measure (positive and between 0 and 1).
We evaluate this approach on synthetic SBM drawn from three cluster configurations: $(1, 2)$, $(2, 3)$ and $(1, 2, 3)$. We compute the ULOT transport plan using fixed hyperparameters $\alpha = 0.5$ and $\rho = 0.01$. As shown in Figure 6 (left), the resulting similarity matrix reveals the structure of the dataset. This similarity matrix can naturally be used for (spectral) graph clustering, or even for dimensionality reduction and visualization as illustrated in Figure 6 (right) with multidimensional scaling of the similarity matrix where the relation between the types of graphs is clearly recovered.

3.2 Solving FUGW for Functional MRI brains

Dataset description We now evaluate ULOT on the Individual Brain Charting (IBC) dataset [23] which is a dataset of functional MRI activations on brain surfaces. The dataset is made of surface meshes with $160k$ vertices associated with different fMRI activations. As the dataset focuses on individual information, it includes subject-specific brain geometrical models, associated with individual fMRI activations. This justifies the necessity of the unbalanced framework to adapt mass between regions that can vary in size between subjects [30].

Experimental setup The high dimensionality of the meshes in the dataset makes it particularly interesting for data augmentation. In order to obtain more training graphs, we perform a parcellation using Ward algorithm [29], and reduce the graph size to 1000 nodes, where the node features are the fMRI activations averaged over the grouped vertices. Data augmentation consists of generating 10 different graphs for each of the 12 subjects, as the results of the Ward clustering using randomly

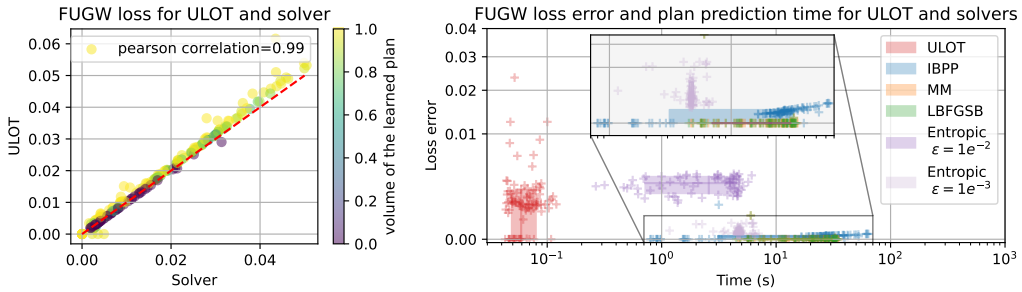

Figure 7: (left) Comparison of the loss obtained with ULOT and IBPP solver, the dashed lines correspond to equality. (right) Plot of loss error VS time for ULOT and other solvers. Colored squares correspond to the 20-80% quantiles for both measures.

sampled activations. The geometric information matrix D is the shortest path distance matrix and the 3D node positions are concatenated with the node activation features to provide a positional encoding. We construct a dataset of the 14400 different graph pairs and train the network using a $60/20/20$ train/val/test split. We provide more details on the experimental setup in Section E.

Comparison to solvers in terms of loss and computational time We first compare the FUGW loss of OT plans predicted by ULOT and the IBPP solver. We find in Figure 7 (left) that both losses are very close and highly correlated with a Pearson correlation of 0.99.
Next we compare the loss error (wrt the best among all solvers) of ULOT and the other solvers introduced in section 2.1, namely IBPP [39], MM [6], LBFGSB and Sinkhorn for the entropic regularized FUGW [8, 30] using the Python library POT [9]. We find in figure 7 (right) that even though ULOT makes errors, it is up to 100 times faster than classical solvers and 10 times faster than Sinkhorn for a smaller error.
This computational gain on graphs of size 1000 is very important as the solvers have a cubic time complexity with respect to the number of nodes, while ULOT has a quadratic time complexity as shown in Figure 8 (left) where computation time is plotted against the number of nodes.

ULOT as warmstart. Finally when high precision is needed, we can use ULOT as a very efficient warm start for the IBPP solver. We find in Figure 8 (right) that using ULOT as a warm start allows the solver to converge much faster. This means that if high precision is required, using ULOT as a warmstart for a solver is an efficient alternative.

fMRI activations prediction using ULOT plans We now illustrate the use of ULOT transport plans on fMRI data for activation prediction between brain graphs. We use the same experimental setup as in [30] where fMRI activations from a source brain graph G_1 are transported to predict activations on a target brain graph G_2. First, we compute the ULOT transport plan $P_\theta^{\alpha,\rho}$ between the two graphs using the model trained above. Then, we predict for a new mental task, *colorless_auditory* contrast from MathLanguage, the fMRI activations $\widehat{F}_2$ on the nodes of G_2 by transporting the activations F_1 from graph G_1:

$$\widehat{F}_2 = \text{diag}\left(\frac{1}{(P_\theta^{\alpha,\rho})_{\#2}}\right)(P_\theta^{\alpha,\rho})^\top F_1. \tag{13}$$

We visualize in Figure 9 the activations on the parcellated brain regions and observe that the general fMRI trend is conserved through the transportation. While this is only a qualitative experiment, this opens doors for future use of OT in large scale experiments on fMRI data.

4 Conclusion, limits and future work

We have introduced ULOT, a new unsupervised deep learning approach for predicting optimal transport plans between graphs, trained by minimizing the FUGW loss. We have shown that ULOT is able to predict transport plans with low error on both simulated and real datasets, up to 100 times faster than classical solvers. Its low complexity and differentiability make it naturally efficient for

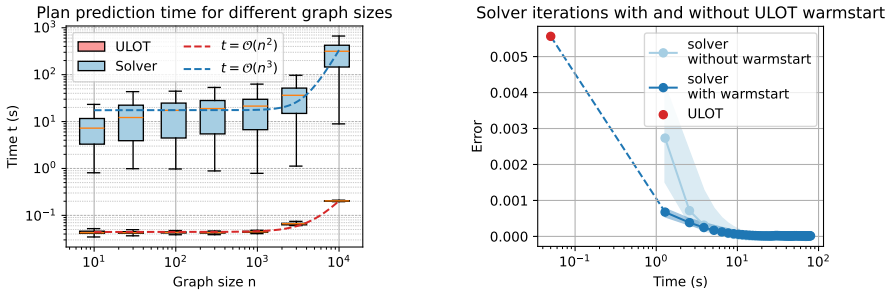

Figure 8: (left) FUGW transport plan prediction time for ULOT and IBPP solver for different graph sizes. (right) IBPP solver loss along iterations with and without ULOT warmstart, reported with the $20\% - 80\%$ quantiles.

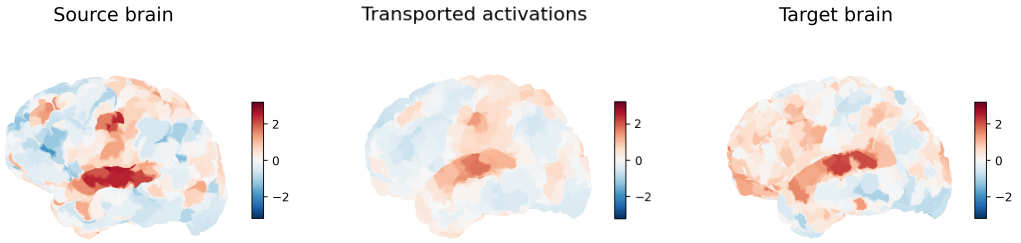

Figure 9: Example of transporting *colorless_auditory* activations from the MathLanguage contrast between two subjects from the IBC dataset. Each brain is equipped with its individual geometry

.

minimizing functionals of optimal transport plans and performing FUGW parameter selection. ULOT also allows for discovering novel ways to use FUGW OT plans such as using their total mass as a measure of similarity of complexity $O(n^2)$ between graphs of different sizes.
While we believe that ULOT is a very promising step towards the use of deep learning for optimal transport, we also acknowledge its limitations and propose research directions for addressing them. The very fast prediction comes at the cost of a small error in the predicted transport plans, which can limit its applications in a context where high precision is needed. While this can be avoided by using ULOT as a warmstart for a solver, there is still room for improvement for directly predicting even more accurate plans. Also we were limited in our experiments to graphs of size $n \leq 10000$ due to GPU memory constraints and going further might require dedicated developments such as lazy tensors or other memory-efficient techniques [7] for cross-attention. In the future, we plan to apply this method to large-scale applications such as activation prediction on high-resolution brain surfaces and computation of graph barycenters. While these applications require large training datasets, which is not the norm for fMRI data, we plan to further investigate our random parcellation data augmentation technique to train a more general model that can be effective across subjects.

Acknowledgments and Disclosure of Funding

This work was granted access to the HPC resources of IDRIS under the allocation 2025-AD011016350 made by GENCI. This work is supported by Hi! PARIS and ANR/France 2030 program (ANR-23-IACL-0005). This research was also supported in part by the French National Research Agency (ANR) through the MATTER project (ANR-23-ERCC-0006-01). This work benefited from state aid managed by the Agence Nationale de la Recherche under the France 2030 programme, reference ANR-22-PESN-0012 and from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement HORIZON-INFRA-2022-SERV-B-01, grant agreement number 10.3030/101147319. Finally, it received funding from the Fondation de l'École polytechnique. We thank Quang Huy Tran for providing code for the FUGW solvers and for helpful discussions.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019.

[2] Brandon Amos, Samuel Cohen, Giulia Luise, and Ievgen Redko. Meta optimal transport. *arXiv preprint arXiv:2206.05262*, 2022.

[3] Brandon Amos et al. Tutorial on amortized optimization. *Foundations and Trends® in Machine Learning*, 16(5):592–732, 2023.

[4] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

[5] Laetitia Chapel, Mokhtar Z Alaya, and Gilles Gasso. Partial gromov-wasserstein with applications on positive-unlabeled learning. *Advances in Neural Information Processing Systems*, 2020.

[6] Laetitia Chapel, Rémi Flamary, Haoran Wu, Cédric Févotte, and Gilles Gasso. Unbalanced optimal transport through non-negative penalized linear regression. *Advances in Neural Information Processing Systems*, 34:23270–23282, 2021.

[7] Benjamin Charlier, Jean Feydy, Joan Alexis Glaunes, François-David Collin, and Ghislain Durif. Kernel operations on the gpu, with autodiff, without memory overflows. *Journal of Machine Learning Research*, 22(74):1–6, 2021.

[8] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123, 2018.

[9] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[10] Zheheng Jiang, Hossein Rahmani, Plamen Angelov, Sue Black, and Bryan M Williams. Graph-context attention networks for size-varied deep graph matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2343–2352, 2022.

[11] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. *arXiv preprint arXiv:2201.12220*, 2022.

[12] Wuyang Li, Xinyu Liu, and Yixuan Yuan. Sigma: Semantic-complete graph matching for domain adaptive object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5291–5300, 2022.

[13] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In *International conference on machine learning*, pages 3835–3845. PMLR, 2019.

[14] Xiang Ling, Lingfei Wu, Saizhuo Wang, Tengfei Ma, Fangli Xu, Alex X Liu, Chunming Wu, and Shouling Ji. Multilevel graph matching networks for deep graph similarity learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2):799–813, 2021.

[15] Chang Liu, Shaofeng Zhang, Xiaokang Yang, and Junchi Yan. Self-supervised learning of visual graph matching. In *European Conference on Computer Vision*, pages 370–388. Springer, 2022.

[16] Eliane Maria Loiola, Nair Maria Maia De Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690, 2007.

[17] Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11:417–487, 2011.

[18] Aldo Moscatelli, Jason Piquenot, Maxime Bérar, Pierre Héroux, and Sébastien Adam. Graph node matching for edit distance. *Pattern Recognition Letters*, 184:14–20, 2024.

[19] Maksim Nekrashevich, Alexander Korotin, and Evgeny Burnaev. Neural gromov-wasserstein optimal transport. *arXiv e-prints*, pages arXiv–2303, 2023.

[20] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 31(4):1–11, 2012.

[21] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pages 2664–2672. PMLR, 2016.

[22] Chengzhi Piao, Tingyang Xu, Xiangguo Sun, Yu Rong, Kangfei Zhao, and Hong Cheng. Computing graph edit distance via neural graph matching. *Proceedings of the VLDB Endowment*, 16(8):1817–1829, 2023.

[23] Ana Luísa Pinho, Alexis Amadon, Torsten Ruest, Murielle Fabre, Elvis Dohmatob, Isabelle Denghien, Chantal Ginisty, Séverine Becuwe-Desmidt, Séverine Roger, Laurence Laurier, et al. Individual brain charting, a high-resolution fmri dataset for cognitive mapping. *Scientific data*, 5(1):1–15, 2018.

[24] Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In *The 22nd International Conference on artificial intelligence and statistics*, pages 849–858. PMLR, 2019.

[25] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[26] Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*, 2017.

[27] Thibault Séjourné, François-Xavier Vialard, and Gabriel Peyré. The unbalanced gromov wasserstein distance: Conic formulation and relaxation. *Advances in Neural Information Processing Systems*, 34:8766–8779, 2021.

[28] Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *International Conference on machine learning*, pages 306–314. PMLR, 2014.

[29] Bertrand Thirion, Gaël Varoquaux, Elvis Dohmatob, and Jean-Baptiste Poline. Which fmri clustering gives good brain parcellations? *Frontiers in neuroscience*, 8:167, 2014.

[30] Alexis Thual, Quang Huy Tran, Tatiana Zemskova, Nicolas Courty, Rémi Flamary, Stanislas Dehaene, and Bertrand Thirion. Aligning individual brains with fused unbalanced gromov wasserstein. *Advances in neural information processing systems*, 35:21792–21804, 2022.

[31] Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pages 6275–6284. PMLR, 2019.

[32] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

[33] Siddharth Tourani, Muhammad Haris Khan, Carsten Rother, and Bogdan Savchynskyy. Discrete cycle-consistency based unsupervised deep graph matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5252–5260, 2024.

[34] Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Fused gromov-wasserstein distance for structured objects. *Algorithms*, 13(9):212, 2020.

[35] Runzhong Wang, Ziao Guo, Shaofei Jiang, Xiaokang Yang, and Junchi Yan. Deep learning of partial graph matching via differentiable top-k. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6272–6281, 2023.

[36] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Learning combinatorial embedding networks for deep graph matching. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3056–3065, 2019.

[37] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Graduated assignment for joint multi-graph matching and clustering with application to unsupervised graph matching network learning. *Advances in neural information processing systems*, 33:19908–19919, 2020.

[38] Tao Wang and Ziv Goldfeld. Neural entropic gromov-wasserstein alignment. *arXiv preprint arXiv:2312.07397*, 2023.

[39] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in artificial intelligence*, pages 433–453. PMLR, 2020.

[40] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*, pages 6932–6941. PMLR, 2019.

[41] Andrei Zanfir and Cristian Sminchisescu. Deep learning of graph matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2684–2693, 2018.

[42] Zhengxin Zhang, Ziv Goldfeld, Youssef Mroueh, and Bharath K Sriperumbudur. Gromov–wasserstein distances: Entropic regularization, duality and sample complexity. *The Annals of Statistics*, 52(4):1616–1645, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the contributions by presenting the experimental results explained throughout the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 4 and in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper is experimental and does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental setup and hyperparameters and presented in the main paper and Section A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer:[Yes]

Justification: The code will be provided in the supplemental materials and at `https://github.com/smazelet/ULOT`. The data is public.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental details and hyperparameters and presented in Section A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Quantitative results are reported with medians and $20\% - 80\%$ quantiles

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computer resources are presented in Section A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The fMRI IBC dataset respects the NeurIPS code of conduct in terms of privacy and content. To the best of our knowledge, this paper does not have potential harmful consequences.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper investigates fast approximation of solutions of optimal transport problems. There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We are the creators of all new assets in the paper, and use code under MIT license. We use the fMRI IBC dataset which is under license CC0.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code details are provided to reproduce the simulated SBM dataset.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involved crowdsourcing nor new research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

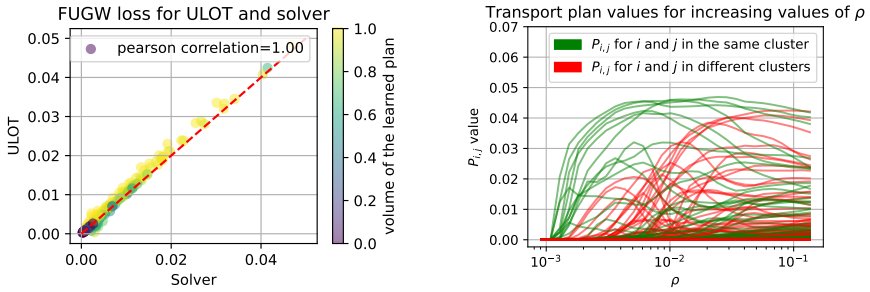

Figure 10: (left) Comparison of the loss obtained with ULOT and IBPP solver for the simulated graphs, the dashed line corresponds to the equality. (right) Transport plan values for increasing values of ρ, colored by whether they link common or different clusters.

A Training setup details

Compute resources We trained our network on an NVIDIA V100 GPU for 100 hours on the IBC dataset and a few hours on the smaller simulated dataset. Note that while the network is $O(n^2)$ with n the number of graph nodes, the main training bottleneck comes from the need to compute the $O(n^3)$ FUGW loss for each transport plan at every epoch.

Hyperparameters The hyperparameters used for training ULOT on both the simulated graphs and the IBC dataset are reported in Table 1. All the MLP and GMN have one hidden layer, with weights shared between the two graph branches. Moreover, the first MLP in the node embedding layer preserves the dimensionality of the input node features. Hyperparameter optimization was performed on a subset of the training data using the Optuna library [1]. The code is available in the supplementary materials and will be released on github upon publication. We will also share the pre-trained model weights for both datasets.

Table 1: ULOT hyperparameters

Hyperparameter	**Simulated dataset**	**IBC dataset**
Learning rate	0.001	0.0001
Batch size	256	64
Optimizer	Adam	Adam
Number of node embedding layers N	5	3
Embedding dimension for α	10	10
Node embedding layer final out dimension	256	256
MLP hidden dimension	64	256
GCN hidden dimension	16	128
Temperature value a	3	5

B Additional experiments on the simulated graphs

Comparison of FUGW loss for ULOT and solver on the simulated graphs We train ULOT on the dataset of simulated SBMs introduced in Section 3.1 and test it on new pairs of simulated graphs sampled from the same distribution. We find in Figure 10 (left) that similarly to the IBC dataset ULOT finds transport plans that have a FUGW loss perfectly correlated with the FUGW loss obtained with the IBPP solver.

Regularization path of the transport plan component $P_{i,j}$ with respect to ρ We predict ULOT transport plans between a pair of graphs with cluster configurations $(1, 2)$ and $(2, 3)$ for different values of ρ and show the regularisation path of each transport plan entry $(P_\theta^{\alpha,\rho})_{i,j}$ for $i \in [1, n_1]$ and $j \in [1, n_2]$ in Figure 10 (right). We observe that transport plan values corresponding to nodes in the shared cluster 2 increase more rapidly with ρ compared to values between nodes in non-overlapping clusters. Moreover, we find that there exists an optimal value around $\rho \simeq 0.05$, for which transport

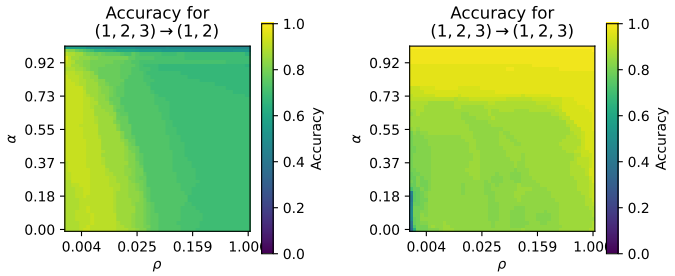

Figure 11: Label propagation accuracy of the ULOT FUGW transport w.r.t. (ρ, α) between (left) different types $(1, 2, 3) \to (1, 2)$, (right) same type $(1, 2, 3)$.

plan mass is predominantly assigned to entries corresponding to the common cluster, effectively discarding irrelevant correspondences.

Visualization of the accuracy for label propagation We visualize the accuracy surfaces on Figure 11 for the label propagation task introduced in Section 3.1 for a range of (α, ρ) values. We consider two different types of pairs: pairs with clusters $(1, 2, 3)$ and $(1, 2)$ and pairs with clusters $(1, 2, 3)$. We see that the accuracy surfaces are smooth and that the optimal parameter values differ across pair types. We optimize ρ and α using gradient descent on each pair of graphs by minimizing the KL divergence between the predicted class probabilities obtained with the ULOT transport plan and the ground truth target one hot encodings of the classes on 50% of the nodes. We obtain an accuracy of 0.87 ± 0.092 on pairs with similar clusters $(1, 2, 3)$ and 0.74 ± 0.11 on the pairs with different clusters.

C Ablation studies

We provide an ablation study of the impact of each step of the OT plan block on the SBM dataset in Table 2. As a first ablation we remove the node scaling in the OT plan head, as a second ablation we replace the whole OT plan head by a much simpler nonlinear transformation of the node embedings to recover an OT plan with $(P_\theta^{(\alpha,\rho)})_{i,j} = \frac{1}{1+||(F_1^{\text{final}})_i - (F_2^{\text{final}})_j||_2^2}$, where F_1^{final} and F_2^{final} are the outputs of the GCN and cross attention block for the first and second graphs respectively.

Table 2: Ablation studies

Ablation	**Relative error**	**Pearson correlation**
None	0.19 ± 0.29	1.0
Remove node scaling	4.1 ± 9.5	0.89
Replace OT plan block by nonlinearity	27 ± 49	0.48

We can see that node scaling is essential for recovering unbalanced OT plans especially for small ρ parameters (unbalanced plans that do not sum to one), and removing them increases the error (but remains surprisingly well correlated). Using a simpler output head for the OT plan leads to a failure to predict OT plans.

D Robustness to distribution shifts

We tested the following distribution shift: we trained ULOT on a dataset of Stochastic Block models with balanced clusters (every cluster has the same number of nodes), and tested the model on a dataset of SBMs with unbalanced clusters, where the cluster ratio is sampled following a Dirichlet distribution. We report the results Table 3 and find that even though the relative loss error increases, the Pearson correlation between the ground truth losses and the ULOT losses remains stable.

Table 3: Distribution shifts

Distribution	Relative error	Pearson correlation
Balanced clusters (train distribution)	0.19 ± 0.29	1.0
Unbalanced clusters (distribution shift)	0.27 ± 0.69	0.98

E Illustration of the fMRI alignement

Experimental details on the IBC dataset We use brain cortical surfaces from the IBC dataset consisting of approximately 160k vertices, each associated with fMRI contrasts obtained as subjects perform specific tasks. To train ULOT, we use the 239 contrasts from the tasks in Table 4. The predicted activation visualized on Figure 9 is selected from the 30 left out contrasts from the MathLanguage task. For each subject, we perform a parcellation of the surface to form 1000 brain regions, using the Ward's hierarchical clustering algorithm. Each contrast is then averaged over the vertices in every brain region. From this parcellation, we construct 1000 node graphs, where edges connect spatially adjacent regions. Each node has a 242 dimensional feature vector composed of the region's contrasts concatenated to its 3D node position.
To augment the dataset, we generate multiple parcellations for each subject by randomly selecting 20% to 40% of the tasks, which produces variability in the geometries and the activations across all generated brains. The final dataset is constructed from all possible graph pairs, which we randomly split into 60% training, 20% validation, and 20% test sets.

Table 4: IBC tasks used for alignment

Task
ArchiEmotional
ArchiSocial
ArchiSpatial
ArchiStandard
HcpEmotion
HcpGambling
HcpLanguage
HcpMotor
HcpRelational
HcpSocial
HcpWm
RSVPLanguage
PreferenceFaces
PreferenceHouses
PreferenceFood
PreferencePaintings
MCSE
Moto
Visu
Audi
MVEB
MVIS
Lec1
Lec2
TheoryOfMind
PainMovie
EmotionalPain
Enumeration
VSTM
Self