OPTIMAL TRANSPORT ON GRAPHS

With Applications to Biomolecules

Kaiwen Shi
Oliver Lab

Jun. 16 2025

Contents

1 Abstract

This is a note prepared by Kaiwen for a chalk talk for the first journal club at CAIPD, Vanderbilt. For any questions, comments, collaborations, please contact Kaiwen Shi.

2 Notations

In this section, we list the normally used notations in this note. Keep in mind, however, that such a notation system might vary to some contextual extent.

1. $\mathcal{X}, \mathcal{Y}$: Spaces.
2. d: Mostly used to denote the distance function, but sometimes it might be slightly abused to appear in differentiation.
3. μ, ν: Measures. Whether they are probability measures can be inferred from context.
4. P, T: Transportation plans, i.e. matrices.
5. δ_x: Dirac delta function whose mass concentrates at point x.
6. $T_{\#}$: Pushforward. Definition can be found at 1.
7. $\langle \cdot, \cdot \rangle$: Inner product. If not specifically mentioned, it is used in the most common case. (e.g. Frobenius for matrices)
8. π: Slightly abused. Sometimes the coupling, sometimes permutation. Can be inferred from context.
9. $supp[\cdot]$: Support of a measure.

Other notations are either explained when used or can be inferred from context.

3 Background

Optimal transport has gained significant attention in recent years because of its effectiveness in deep learning and computer vision. Its descendant metric, the Wasserstein distance, has been particularly successful in measuring distributional similarity in low-dimensional Euclidean spaces. More recently, theories have been developed to generalize the Wasserstein distance to measure similarity between measures in different metric measure spaces, creating more venues to compare more complicated mathematical structures.

In this note, we will cover the optimal transport theory, the Wasserstein distance, the Gromov-Wasserstein distance, and its applications on graphs. We will also include some discussion on how they can be applied to learn the structural information of biological molecules, such as proteins.

3.1 Optimal Transport

The original optimal transport problem can be dated to the work of Gaspard Monge in 1781[11]. It was motivated by the transportation of masses between sources and sinks. Let δ denote the Dirac delta function and δ_x the Dirac delta function whose mass is concentrated at the point x. Then we can let

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}, \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j} \tag{1}$$

denote source and sink, respectively. When μ and ν have equal mass 1, that is, $\sum_{i=1}^{n} a_i = 1 = \sum_{j=1}^{m} b_j$, the Monge formulation aims to find $T : \{x_1, x_2, \dots, x_n\} \to \{y_1, y_2, \dots, y_m\}$ such that $\forall j \in \{1, 2, \dots, m\}, b_j = \sum_{i:T(x_i)=y_j} a_i$.

Here is another formulation of the Monge problem with a definition of pushforward.

Definition 1. *(Pushforward) For T a map from space $\mathcal{X}$ to space $\mathcal{Y}$, we define a corresponding pushforward operator $T_\# : \mathcal{M}(\mathcal{X}) \to \mathcal{M}(\mathcal{Y})$ by $T_\#\mu(B) = \mu\left(\{x : T(x) \in B\}\right), \forall B \subset Y$ a measurable set, where $\mathcal{M}(\mathcal{X})$ stands for the space of measures on $\mathcal{X}$. For discrete measures, the push-forward operation consists simply of moving the positions of all the points in the support of the measure.*

Then, with this definition, the Monge problem is:

$$\min_T \{\sum_i c(x_i, T(x_i)) : T_\#\mu = \nu\} \tag{2}$$

where $c(x, y)$ denotes the cost to transport mass from point x to point y. In Euclidean spaces, it could be for example the Euclidean distance.

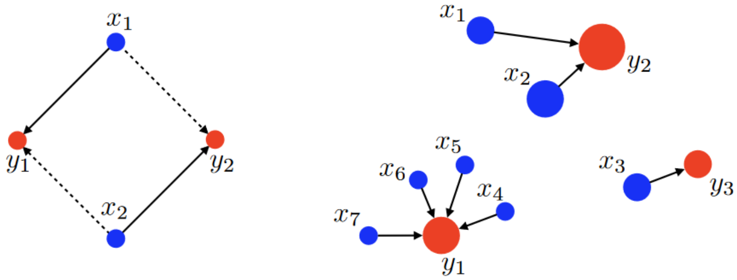

Figure 1: Visualizaion of pushforwards moving the masses at x_i to position y_j[14].

During WWII, the Soviet mathematician Leonid Kantorovich worked extensively on this problem and made huge contributions to this field [6]. Part of his work included a reformulation of the transportation problem. Monge's problem might not have a solution because all the mass in one point from $\mathcal{X}$ will necessarily be transported to another point in $\mathcal{Y}$, with no room for moving the mass to other points. Kantorovich's relaxation of the problem gave a new perspective for splitting the mass: it allows mass from x_i to be transported to more than one location.

Definition 2. *(Transport Plan) Consider the discrete measures μ and ν as before. As they are discrete, we define vectors $\mathbf{a}$ and $\mathbf{b}$ with $\mathbf{a}_i = a_i$ and $\mathbf{b}_j = b_j$, essentially the coefficients of Dirac functions in μ and ν (See eq.* (1)*), to represent the measures μ and ν. Then a transport plan $\mathbf{P}$ can be defined as*

$$\mathbf{P} \in \mathbb{R}_+^{n\times m}: \quad \mathbf{P}\mathbf{1}_m = \mathbf{a}, \quad \mathbf{P}^T\mathbf{1}_n = \mathbf{b} \tag{3}$$

where $\mathbb{R}_+^{n\times m}$ denotes the space of non-negative real-valued matrices of size $n\times m$, and $\mathbf{1}_m$ denotes a column vector of size $m \times 1$ with all entries equal to 1, i.e., $\mathbf{1}_m = [1, 1, \ldots, 1]^T$, and similarly, $\mathbf{1}_n$ denotes a column vector of size $n \times 1$ with all entries equal to 1.

Let $U(\mathbf{a}, \mathbf{b})$ denote the set of all $\mathbf{P}$ satisfying the above condition. The Kantorovich's problem then is:

$$L_c(\mathbf{a}, \mathbf{b}) \quad := \min_{\mathbf{P}\in U(\mathbf{a},\mathbf{b})} \langle \mathbf{C}, \mathbf{P} \rangle \quad := \sum_{i.j} \mathbf{C}_{i,j}\mathbf{P}_{i,j} \tag{4}$$

where the entries $\mathbf{C}_{i,j}$ represent the cost of transporting a unit mass from the i-th element of μ to the j-th element of ν, $\mathbf{P}_{i,j}$ denotes the amount of mass transported from x_i to y_j, and $\langle \mathbf{C}, \mathbf{P} \rangle$ represents the Frobenius inner product between the cost matrix $\mathbf{C}$ and the transport plan $\mathbf{P}$.

For arbitrary measures α and β in space $\mathcal{X}$ and $\mathcal{Y}$, the Kantorovich reformulation can be extended by defining a coupling space $\mathbf{\Pi}(\alpha, \beta) = \{\pi \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y}) : \mathcal{P}_{\mathcal{X}\#}\pi = \alpha \quad \text{and} \quad \mathcal{P}_{\mathcal{Y}\#}\pi = \beta\} \subset \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$. Its element π is called a coupling between α and β, which is a joint distribution over the product space. The superscript one and subscript plus indicate that the coupling is positive and mass preservative.

Definition 3. *(coupling) Let α and β be arbitrary measure in $\mathcal{X}$ and $\mathcal{Y}$, we define the* *coupling set as*

$$\mathbf{\Pi}(\alpha,\beta) = \{\pi \in \mathcal{M}_+^1(\mathcal{X}\times\mathcal{Y}) : \mathcal{P}_{\mathcal{X}\#}\pi = \alpha \quad \textit{and} \quad \mathcal{P}_{\mathcal{Y}\#}\pi = \beta\} \tag{5}$$

where $\mathcal{P}_{\mathcal{X}\#}$ and $\mathcal{P}_{\mathcal{Y}\#}$ are the pushforwards (See Def. on pushforward 1) of the projections $P_{\mathcal{X}}(x,y) = x$ and $P_{\mathcal{Y}}(x,y) = y$. The constraints on teh couplings are equivalent to imposing that $\pi(\mathrm{A}\times\mathcal{Y}) = \alpha(\mathrm{A})$ and $\pi(\mathcal{X}\times\mathrm{B}) = \beta(\mathrm{B})$. for all sets $\mathrm{A}\subset\mathcal{X}$ and $\mathrm{B}\subset\mathcal{Y}$. The Kantorovich problem is then generalized as

$$\mathcal{L}_c(\alpha,\beta) = \underset{\pi\in\Pi(\alpha,\beta)}{min} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)d\pi(x,y) \tag{6}$$

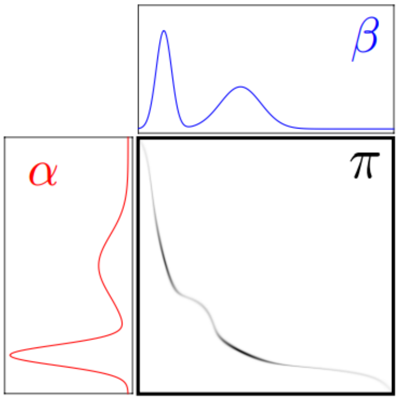

Figure 2: Coupling π between α and β localized around the Monge map $(x, T(x))$ (displayed in black) [14]

3.2 Wasserstein distance

Also known as the Earth Mover's Distance (EMD), the Wasserstein distance is a metric that quantifies the cost of transporting one probability distribution to another. The definition is given as followed.

Definition 4. *(p-Wasserstein Distance) Consider two probability measures μ and ν defined on a metric space $(\mathcal{X}, d)$, where d is the distance function on $\mathcal{X}$. The p-Wasserstein distance, denoted as $W_p(\mu,\nu)$, is defined as:*

$$W_p(\mu,\nu) = \left(\inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^p \, d\pi(x,y)\right)^{1/p}, \tag{7}$$

where $\Pi(\mu,\nu)$ is the set of all couplings of μ and ν. To remind one, a coupling $\pi\in\Pi(\mu,\nu)$ is a joint probability distribution over $\mathcal{X}\times\mathcal{X}$ with marginals μ and ν. A proof of this distance as a metric can be found in [14].

Notice two things:

1. When $p = 1$, the Wasserstein distance recovers the Kontorovich problem formulation between metric space $\mathcal{X}$ and itself.

2. (even though hinted in 1.) The metric is defined on one single metric space $(\mathcal{X}, d)$. This means that the associated metric, structure, topology, geometry, and etc. are shared. We will later see when the spaces are not the same.

One good property of the Wasserstein distance is that the gradient signal carries much greater information than some other choice, thus very suitable for machine learning purposes. The figure below is cited from [8], a previous work by one of my undergraduate advisors, Dr. Soheil Kolouri.

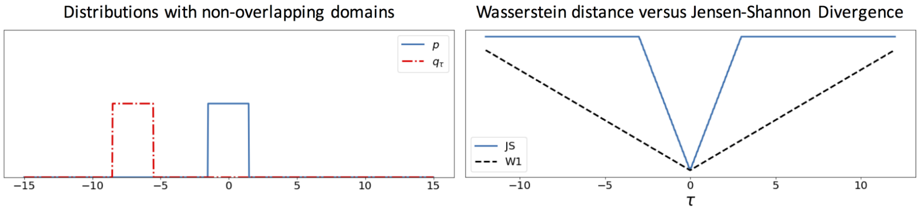

Figure 3: $W_1(p, q_\tau)$ and $JS(p, q_\tau)$ when p is a uniform distribution around zero and $q_\tau = p(x - \tau)$. JS divergence does not provide a usable gradient when distributions are supported on non-overlapping domains.

The major problem with the Wasserstein distance is that computation is not cheap. For discrete measures supported on the same number of points, that is, for $\mu = \sum_{i=1}^{n} \delta_{x_i}$ and $\nu = \sum_{j=1}^{n} \delta_{x_j}$, the problem becomes an assignment problem, which can be solved using the Hungarian Algorithm [9]. The time complexity is $O(n^3)$. In general, the 1-Wasserstein distance problem is solvable by solving a Linear Programming optimization problem, whose many solvers are polynomial-time.

To address this computation bottleneck, variants such as Sliced Wasserstein [7] and Sinkhorn-Knopp algorithm [3] are proposed. However, we are not going to dive into this in depth in this note as it is slightly out of context.

4 Gromov-Wasserstein Family

There is yet another problem to apply Wasserstein distance to graph comparison. In previous sections, we emphasized that the Wasserstein distance describes the distance between two measures living in the same metric space. In that space, they most importantly share the same metric, thus making the distance between points quantifiable. However, in comparing graphs, one does not necessarily have this convenience to define a metric across nodes within different graphs. After all, the spaces in which the graphs live in can vary a lot, depending on the number of nodes, the number of edges, and how the edges are placed, etc.

4.1 Gromov-Wasserstein Distance

Below are a few technicalities for the definition of the Gromov-Wasserstein distance. A lot of the texts here are adopted from a seminar note by Dongming(Merrick) Hua. For curious readers, please refer to [12] and [18] for a more formal read.

Definition 5. *(Metric Measure Space) A mm-space is a triple $(\mathcal{X}, d, \mu)$, where $(\mathcal{X}, d)$ is a metric space and μ is a Borel measure on $(\mathcal{X}, d)$ (i.e. a measure on the Borel σ-algebra of $(\mathcal{X}, d)$), with the following properties:*

1. *Metric space $(\mathcal{X}, d)$ is complete (Cauchy sequences converge) and separable (contains a countable dense subset).* [1]

2. *Measure is locally finite: $\forall x \in \mathcal{X}$ and $r > 0$ small, we have $\mu(B_r(x)) < \infty$.*

Definition 6. *(Isomorphism) Two mm-space $(\mathcal{X}_1, d_1, \mu_1)$ and $(\mathcal{X}_2, d_2, \mu_2)$ are isomorphic if $\exists f : supp[\mu_1] \to supp[\mu_2]$ an isometry (preserve distance) such that $\mu_2 = f_{\#}\mu_1$.*

With this definition of isomorphism, we can define a notion of equivalence between mm-spaces. Later, when we define the Gromov-Wasserstein distance, we will see how the notion of isomorphism/equivalence is used.

Definition 7. *(Measure coupling) Let $(\mathcal{X}_1, d_1, \mu_1)$ and $(\mathcal{X}_2, d_2, \mu_2)$ be mm-spaces. A measure μ on the product space $\mathcal{X}_1 \times \mathcal{X}_2$ is a measure coupling of μ_1, μ_2 if $\mu(B \times \mathcal{X}_2) = \mu_1(B)$ and $\mu(\mathcal{X}_1 \times B') = \mu_2(B')$ for all measurable $B \subseteq \mathcal{X}_1$ and $B' \subseteq \mathcal{X}_2$.*

Definition 8. *(Distance coupling) Let $(\mathcal{X}_1, d_1, \mu_1)$ and $(\mathcal{X}_2, d_2, \mu_2)$ be mm-spaces. A metric d on the disjoint union $\mathcal{X}_1 \bigsqcup \mathcal{X}_2$ is a metric coupling of d_1, d_2 if $d(x, y) = d_1(x, y), \forall x, y \in supp[\mu_1]$ and $d(x', y') = d_2(x', y'), \forall x', y' \in supp[\mu_2]$.*

Think of this distance coupling as a shared key between different locks. Notice that this distance coupling still does not define a notion of distance between two points respectively in $\mathcal{X}_1$ and $\mathcal{X}_2$. This is perfectly fine since otherwise the distance coupling can be used to calculate the Wasserstein distance between μ_1 and μ_2 and we will not need Gromov-Wasserstein in this case for a measurement of (dis)similarity.

Definition 9. *(Gromov-Wasserstein p-Distance) Let $(\mathcal{X}_1, d_1, \mu_1)$ and $(\mathcal{X}_2, d_2, \mu_2)$ be mm-spaces. The Gromov-Wasserstein p-Distance is defined as*

$$d_{GW,p}((\mathcal{X}_1, d_1, \mu_1), (\mathcal{X}_2, d_2, \mu_2)) = \inf(\int_{\mathcal{X}_1 \times \mathcal{X}_2} d(x, y)^p d\mu)^{\frac{1}{p}} \tag{8}$$

where μ is a measure coupling of μ_1, μ_2 and d is a metric coupling of d_1, d_2.

We will use $p = 2$ in this note and thus drop the subscript. An equivalent definition of this distance is:

$$d_{GW}((\mathcal{X}_1, d_1, \mu_1), (\mathcal{X}_2, d_2, \mu_2)) = \inf d_W(\phi_{\#}\mu_1, \phi'_{\#}\mu_2) \tag{9}$$

where $\phi : supp[\mu_1] \to \mathcal{X}$ and $\phi' : supp[\mu_2] \to \mathcal{X}$ are isometric embeddings, and $d_W(\cdot, \cdot)$ denotes the Wasserstein distance.

The second definition can be interpreted easily. Basically, one first embeds isometrically the two measures into a space, and within that space, one computes the Wasserstein distance. With this definition, we now have a theoretical ground on which we define a metric between two mm-spaces. But problems persist as to how to compute the distance empirically.

[1] Dense subset: we say $\mathcal{A}$ is a dense subset of $\mathcal{X}$ if $\forall x \in \mathcal{A}, \exists y \in \mathcal{X}$ such that $x = y$ or $x \in B_r(y, \epsilon)$. In other words, for every point in $\mathcal{A}$, either it is an element of $\mathcal{X}$, or is arbitrarily close to an element in $\mathcal{X}$.

4.2 Gromov-Wasserstein Discrepancy

In this work [15], Peyré et al. defined a pseudometric called the Gromov-Wasserstein Discrepancy that made computation possible, and proposed a fast algorithm to compute it.

Definition 10. *(Gromov-Wasserstein Discrepancy) Let $(\mathcal{X}_1, d_1, \mu_1)$ and $(\mathcal{X}_2, d_2, \mu_2)$ be mm-spaces, with $(\mathcal{X}_1, d_1)$ is compact and μ_1 is a Borel probability measure on $\mathcal{X}_1$. (So is $(\mathcal{X}_2, d_2, \mu_2)$ defined.) The Gromov-Wasserstein Discrepancy is defined as*

$$d_{GWD}(\mu_1, \mu_2) = \inf_{\pi \in \Pi(\mu_1,\mu_2)} \int_{\mathcal{X}_1 \times \mathcal{X}_2} \int_{\mathcal{X}_1 \times \mathcal{X}_2} L(x_1, x_2, x_1', x_2') d\pi(x_1, x_2) d\pi(x_1', x_2') \tag{10}$$

where $L(x_1, x_2, x_1', x_2') = |d_1(x_1, x_1') - d_2(x_2, x_2')|$.

This definition is adopted from [15] and [20]. It should be easy to verify that $L(x_1, x_2, x_1', x_2') = |d_1(x_1, x_1') - d_2(x_2, x_2')|$ is a metric coupling. In the setting where the measures are discrete, such as in graphs where masses are concentrated on nodes, we define

Definition 11. *(GWD on graphs) Given two graphs $G_1(V_1, E_1, C_1, \mu_1)$ and $G_2(V_2, E_2, C_2, \mu_2)$, where for $s \in \{1, 2\}$, V_s is the set of vertices, E_s is the set of edges, $C_s = [c^s_{i,j}] \in \mathbb{R}^{|V_s| \times |V_s|}$ is the cost matrix, and μ_s is the measure associated with the graph G_s, the Gromov-Wasserstein Discrepancy is defined as*

$$\begin{aligned} d_{GWD}(\mu_1, \mu_2) &= \min_{T \in \mathcal{T}(\mu_1,\mu_2)} \sum_{i,j,i',j'} L(c^1_{i,j}, c^2_{i',j'}) T_{i,i'} T_{j,j'} && (11) \\ &= \min_{T \in \mathcal{T}(\mu_1,\mu_2)} \langle \mathcal{L}(C_1, C_2, T), T \rangle && (12) \end{aligned}$$

where $\mathcal{L}(C_1, C_2, T)$ is the loss matrix $[L_{j,j'}] \in \mathbb{R}^{|V_1| \times |V_2|}$ such that $L_{j,j'} = \sum_{i,i'} L(c^1_{i,j}, c^2_{i',j'} T_{i,i'})$.

There is a lot to digest. Let us make sense of this definition. First, compare (8) and (11), one might wonder why the integral becomes a double integral over the same product space. An informal explanation is try to see the formulation as an extension of the Quadratic Assignment Problem [10], where the cost is defined not only by the weights defined between points in the source, but also the distances between the points being transported into the target. Thus, the inner integral can be seen as the distance calculation given a soft assignment π, and the outter integral aggregates the cost based on the weights induced by the soft assignment π.

Second, the summation over four variables in (11) seems daunting, but again, it would be less intimidating to see it as in (12). What (12) does is basically Wasserstein distance calculation with the cost matrix $\mathcal{L}(C_1, C_2, T)$, with one caveat that the cost matrix is a function of the transport plan T. This indeed traces back to what we cover in the first point above, where the QAP formulation allows the distance to be a function of the assignment.

Notice that in some literature, such as [15], the cost function is written as a tensor product:

$$\mathcal{L} \otimes T \overset{\text{def.}}{=} \left(\sum_{k,\ell} \mathcal{L}_{i,j,k,\ell} T_{k,\ell} \right)_{i,j} \tag{13}$$

Therefore, (11) can be written as:

$$d_{GWD}(\mu_1, \mu_2) = \min_{T \in \mathcal{T}(\mu_1, \mu_2)} \langle \mathcal{L}(C_1, C_2) \otimes T, T \rangle, \tag{14}$$

5 Application on Biomolecules

We now have a framework for comparing relational data, many of which can be represented with graphs. Thus, a natural target is biomolecules, such as proteins or DNA, whose building blocks are connected by physical forces and hence can be modeled by graphs. Previous research has done much analysis on the graphical nature of such biomolecules. For example, in [2], K. V. Brinda et al. found that the degree distribution of graphs generated by protein sidechain-linkage surpassing an interaction strength threshold fits well to a Poison distribution, much frequently observed in Erdõs–Rényi model. They also found that the largest cluster(connected components) as a function of the interaction strength threshold fits well with the same model by adjusting the edge formation probability.

5.1 Protein Structures as Graphs

The problem now remains what is the most useful graphical representation for a protein? [2] Or rather, with some points in the 3D Euclidean space as representations of the amino acids (the featurization of the point cloud can be customized), how are the edges formed? There are many propositions, but researchers have not considered any one the best practice. For instance, [4] uses a KNN search to define the neighborhoods for nodes in the graph, [16] is concerned with the H-bond linkage between the residues detectable by a spherical probe over the van der Waals surface, [2] defines an interaction strength between side chains, [5] relaxes the Delaunay Tessellation and achieves a much sparser graph representation, and many others just make a neighborhood cut-off based on the distances between residues.

The choice of edge formation would affect the semantics of the protein graphs, of course. So, it is an important design choice. However, once a good graph design protocol is established for the protein structures, various downstream tasks can be performed.

5.2 Graph Matching

A natural question to pose when studying proteins is that, given two protein structures, how similar are they? One heuristic, but inaccurate, way to measure it is by aligning the sequences and calculating the mismatch. However, there are some roadblocks. First of all, we may not have sequence information even when the structures are known. Second, aligning the sequence is a computationally non-trivial task. Third, even if the sequences are the same, conformational perturbations can make two structures different. Thus, a way to compare relational structured data is critical for answering such question.

The problem can be framed as a graph matching problem. When applied in a strict sense, it is a graph isomorphism problem.

Definition 12. *(Graph Isomorphism) Given two graphs G and H, we say they are isomorphic if there is a bijection between the vertex sets of G and H, $f : V(G) \to V(H)$ such that $(f(v), f(u)) \in E(H), \forall (v, u) \in E(G)$. $E(\cdot)$ denotes the edge set.*

[2] We will restrict our discussion to proteins for now. The RNA structures can be analyzed in a similar way with some twists.

A graph isomorphism problem thus is asking, given two graphs, are they isomorphic? In protein structures, such restrictive correspondence can hardly be found. The problem thus needs to be relaxed. A graph matching problem hence aims to find a matching between the vertex sets, with some minimization/maximization on an objective. For instance, given two graphs with the same number of vertices and undirected unweighted edges, the graph matching can be defined as:

Definition 13. *(Graph Alignment) Given graphs $G(V_g, E_g)$ and $H(V_h, E_h)$ with $|V_g| = |V_h| = N$, we want to find π^* an optimal permutation of indices, such that the aligned graphs G, H_{π^*} (by permuting H) share the greatest number of common edges.*

$$\pi^* = \arg\max_{\pi \in P_N} \sum_{i,j \in \{1,\ldots,N\}} A^g_{i,j} A^h_{\pi(i),\pi(j)} = \arg\max_{\pi \in P_N} \langle A^g, \Pi A^h \rangle \tag{15}$$

where P_N is the permutation of N indices, A is the adjacency matrix, Π is the permutation matrix corresponding to π, and $\langle \cdot, \cdot \rangle$ is the Frobenius norm.

Note that in this construction, graphs are "simple" (no complex vertex features and edge features) and the objective is "simple" (Frobenius norm), but the computation cost is already high. In fact, because the permutation set is combinatorial in nature, the search space is combinatorial. It is not tractable without a search strategy.

The case will be worse in proteins. For protein graphs, the graphs are complex, as the vertices and edges have their own features, the sequence length might be different, and we do not have a clear objective to optimize. Here, the power of the Gromov-Wasserstein Discrepancy kicks in. The Gromov-Wasserstein method provides a means to compare two graphs, with the cost function customizable to reflect the edge features. Variants of the GWD, such as fused Gromov-Wasserstein[19], allow incorporation of vertex features into the comparison agenda. Computational approximations have been proposed to address the NP-Hard challenge, so we can have an approximate GWD within polynomial time complexity[15][20]. Moreover, in cases where a protein snippet needs to be matched to a whole protein, or a certain location within (people coin this problem subgraph matching), partial optimal transport metrics, such as partial Gromov-Wasserstein[1] or partial fused Gromov-Wasserstein[13] can help.

5.3 Graph Barycenter and the "Average" Protein

The previous section gives us a taste on pair-wise comparisons between two protein structures. Another good question to ask is, given a collection of protein graphs, is there an "average" graph? Such a question is valuable in two senses. On one hand, if a good mean is defined, we can model the protein dynamics with perturbations around this mean. That is, we can, to some extent, capture conformational changes with a single structure [3]. On the other hand, with the definition of an average protein, it becomes easier to characterize a collection of proteins with this average being a meta-datum.

Such average is called a barycenter in Optimal Transport theory. The intuition is to find a measure that minimizes the average distance to all the measures in the dataset.

[3] This is a somewhat bold claim and we have not validate such thoughts. There are also some technical challenges with pre-definition of barycenter parameters.

The definition follows, and since this note talks primarily about Gromov-Wasserstein, the definition would be framed under GWD.

Definition 14. *(GWD Barycenter) Given a set of mm-spaces* $(\mathcal{X}_i, d_i, \mu_i).i \in \{1, \ldots, N\}$, *the* *Gromov-Wasserstein Barycenter* *is a mm-space* $(\mathcal{Y}, d_y, \mu_y)$ *with*

$$\mu_y = \underset{admissible\ \nu}{\arg\min} \sum_{i=1}^{N} \lambda_i d_{GWD}(\nu, \mu_i) \tag{16}$$

where admissible means the measure is properly defined with any prior constraints that we want to impose, λ_i *is the weight for aggregation, and* d_{GWD} *follows the definition as in* (10).

To the best of my knowledge, this using barycenter to represent an average of proteins is largely unexplored. The effectiveness cannot be guaranteed as a result, but conceptually it is a promising direction. Also, the measure space of protein graphs are designable, i.e. you could define your own desirable number of nodes like the mean of the sequence lengths, your own cost matrix based on vertex features and edge features, and your own measures such as uniform measures[13] or degree-based measures[20]. Consequently, the flexibility is immense and expressive power can be high for strong data representations.

6 Ending Note

As an ending note, I want to thank a few people and give a few more applications that I do not have time to cover. I am sincerely grateful for the trust Dr.Carlos Oliver has in me to make me the first speaker of the journal club. Also, a huge shout-out to Dr.Soheil Kolouri for leading me into the door of Optimal Transport and all those helpful discussions.

Optimal Transport is such a powerful framework with which a lot of real-world events can be modeled. Due to the time limit, I will not have the leisure to discuss them all. However, I do want to point out a few more directions people are exploring with Optimal Transport in graph learning:

1. Graph Partitioning aims to define equivalence classes for a graph such that every node can be assigned to a bucket. With this partition, a classification of the vertices would be made easier. Also, such a partition can be used to match graphs. The intuition is to match the nodes that are assigned to the same bucket[20].

2. Graph Editing cares for finding an optimal editing agenda for transforming a source graph into a target graph. The editing actions include adding or deleting a vertex, adding or deleting a node, and substituting the vertices and edges, each associated with a cost[17]. People defined a distance between two graphs called Graph Editing Distance with the least amount of cost to edit one graph into another and this can be modeled as a Linear Assignment Problem, solvable by the Hungarian Algorithm[9].

References

[1] Yikun Bai, Rocio Diaz Martin, Abihith Kothapalli, Hengrong Du, Xinran Liu, and Soheil Kolouri. Partial gromov-wasserstein metric, 2025.

[2] K. V. Brinda, Saraswathi Vishveshwara, and Smitha Vishveshwara. Random network behaviour of protein structures. *Mol. BioSyst.*, 6:391–398, 2010.

[3] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[4] Zhangyang Gao, Cheng Tan, Pablo Chacón, and Stan Z. Li. Pifold: Toward effective and efficient protein inverse folding, 2023.

[5] Jun Huan, Wei Wang, Deepak Bandyopadhyay, Jack Snoeyink, Jan Prins, and Alexander Tropsha. Mining protein family specific residue packing patterns from protein structure graphs. In *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology*, RECOMB '04, page 308–315, New York, NY, USA, 2004. Association for Computing Machinery.

[6] L. V. Kantorovich. On translation of mass (in russian). *C. R. Doklady. Proceedings of the USSR Academy of Sciences*, 37:199–201, 1942.

[7] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced wasserstein distances. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[8] Soheil Kolouri, Phillip E. Pope, Charles E. Martin, and Gustavo K. Rohde. Sliced-wasserstein autoencoder: An embarrassingly simple generative model, 2018.

[9] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955.

[10] Eliane Loiola, Nair Abreu, Paulo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey of the quadratic assignment problem. *European Journal of Operational Research*, 176:657–690, 01 2007.

[11] G. Monge. *Mémoire sur la théorie des déblais et des remblais.* De l'Imprimerie Royale, 1781.

[12] Facundo Mémoli. Gromov-wasserstein distances and the metric approach to object matching. *Found. Comput. Math.*, 11(4):417–487, 2011.

[13] Wen-Xin Pan, Isabel Haasler, and Pascal Frossard. Subgraph matching via partial optimal transport, 2024.

[14] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[15] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2664–2672, New York, New York, USA, 20–22 Jun 2016. PMLR.

[16] Ofer Rahat, Uri Alon, Yaakov Levy, and Gideon Schreiber. Understanding hydrogen-bond patterns in proteins using network motifs. *Bioinformatics*, 25(22):2921–2928, 09 2009.

[17] Alberto Sanfeliu and King-Sun Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(3):353–362, 1983.

[18] Karl-Theodor Sturm. On the geometry of metric measure spaces. i. *Acta Math.*, 196(1):65–131, 2006.

[19] Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Optimal transport for structured data with application on graphs, 2019.

[20] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-Wasserstein learning for graph matching and node embedding. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6932–6941. PMLR, 09–15 Jun 2019.