Graph GOSPA similarity function for Gaussian process regression on graphs

Jinhao Gu
Department of Electrical Engineering and Electronics
University of Liverpool
Liverpool, United Kingdom
jinhgu@liverpool.ac.uk

Ángel F. García-Fernández
IPTC, ETSI de Telecomunicación
Universidad Politécnica de Madrid
Madrid, Spain
angel.garcia.fernandez@upm.es

Robert E. Firth
STFC Hatree Centre
Daresbury, United Kingdom
robert.firth@stfc.ac.uk

***Abstract*—In this paper we propose a similarity function between graphs based on a mathematically principled metric for graphs of different sizes: the graph generalised optimal subpattern assignment (GOSPA) metric. The similarity function is based on an optimal assignment between nodes and has an interpretable meaning in terms of similarity for node attribute error, number of unassigned nodes, and number of edge mismatches. The proposed similarity function is computable in polynomial time. We also propose its use in Gaussian processes (GPs) for graphs to predict molecular properties. Experimental results show the benefits of the proposed GP model compared to other GP baselines.**

***Index Terms*—Gaussian process, graph metric, regression.**

I. Introduction

A graph is a type of representation that contains both node features and connectivity information and is able to model structural relationships and domain-specific properties in various fields [1]–[3]. As molecules consist of atoms and bonds, they can be intuitively represented by graphs [4]. Gaussian processes (GPs) are a type of kernel-based method to solve regression and classification problems [5] that are especially suitable for small datasets, since they typically only have a few parameters and also provide the uncertainty quantification on the models. GPs can be used with inputs that graph using a kernel for graphs [6]. Generally, there are three types of kernels or similarity functions for graphs:

(I) Diffusion kernels based on a metric on graphs [7, Chap. 5], such as the graph edit distance (GED) [8]. A drawback of these kernels is that they are computationally intensive to compute because of the matrix exponential in kernel calculation.

(II) Similarity measures based on applying a transformation to the GED such that low metric values are mapped to high similarities, and the other way round [7, Chap. 5]. While these transformations do not define valid kernels, they can be used in practice [9]. A drawback is that the computation of the GED is generally NP-hard [10].

(III) Kernels based on features obtained via pre-processing of the graphs, which can imply a loss of information. Examples of these are the random walk kernel [11], [12], and the Weisfeiler-Lehman (WL) graph kernel [13], [14].

In this paper, we propose a similarity function between graphs, where each node can have certain features, that is based on a mathematically principled metric for graphs, meeting the identity, symmetry and triangle inequality properties. In particular, we propose to use the graph generalized optimal subpattern assignment (GOSPA) metric [15], [16]. The graph GOSPA metric is based on computing an optimal assignment between nodes by penalising node attributes for assigned nodes, the number of unassigned nodes and the number of edge mismatches. Therefore, the graph GOSPA similarity function has an interpretable meaning, inherited from the graph GOSPA metric, which takes into account the whole graph information and can be computed in polynomial time.

Our contributions can be summarised as follows:

(1) We propose a novel similarity measure for graphs, based on the graph GOSPA metric.

(2) We show the decomposition of the graph GOSPA similarity into interpretable components.

(3) We use the graph GOSPA similarity as the kernel function of a GP to predict molecular properties in several datasets. Experimental results demonstrate that Graph GOSPA GP has the best performance compared to other GP baselines in several of the considered datasets. We also show that the decomposition of the kernel can be used to assist with the interpretation of the similarity score.

II. Background

A. Weighted undirected graphs

A weighted, undirected graph is formed by vertices (also called nodes) and weighted edges, each edge connecting two vertices. The set of vertices is $V = \{x_1, \ldots, x_n\}$ with the i-th node feature denoted by $x_i \in \mathbb{R}^N$ [17]. The edges and their weights can be represented by a symmetric adjacency matrix $A \in \mathbb{R}^{n \times n}$, whose (i, j) element $A(i, j)$ indicates the weight between the i-th and j-th node, with $A(i, j) = 0$ indicating no edge.

B. Graph GOSPA metric

The graph GOSPA metric is a mathematically principled metric, as it meets the identity, symmetry, and triangle inequality properties, for graphs of different sizes [15]. Let us consider two graphs X and Y with vertices $V_X = \{x_1, \ldots, x_{n_X}\}$,

$V_Y = \{y_1, \ldots, y_{n_Y}\}$, and adjacency matrices $A_X \in \mathbb{R}^{n_X \times n_X}$ and $A_Y \in \mathbb{R}^{n_Y \times n_Y}$.

The graph GOSPA metric looks for an optimal assignment between nodes in V_X and nodes in V_Y, with the option of leaving some nodes unassigned. The assignments between V_X and V_Y can be represented by an $(n_X + 1) \times (n_Y + 1)$ binary matrix W. The element $W(i,j) = 1$ if x_i is assigned to y_j. If x_i remains unassigned, $W(i, n_Y + 1) = 1$, and if y_j remains unassigned then $W(n_X + 1, j) = 1$. The set of all binary matrices is denoted by $\mathcal{W}_{X,Y}$, which is defined in [15, Eq. (4)-(7)].

If we consider X to be a ground truth graph and Y an estimate (obtained by some algorithm), the unassigned nodes in X and Y are referred to as missed and false nodes, respectively. For $1 < p < \infty$, a scalar $c > 0$, edge mismatch penalty $\epsilon > 0$ and base metric $d(\cdot,\cdot)$ on the node feature space $\mathbb{R}^N$, the graph GOSPA metric $d_p^{(c,\epsilon)}(\cdot,\cdot)$ between two graphs X and Y is [15]

$$d_p^{(c,\epsilon)}(X,Y) = \min_{W \in \mathcal{W}_{X,Y}} \left(\mathrm{tr}\left[D_{X,Y}^{\top} W\right] + e_{X,Y}(W)^p\right)^{1/p} \quad (1)$$

where

$$D_{X,Y}(i,j) = \begin{cases} d\left(x_i, y_j\right)^p & i \le n_X, j \le n_Y, \\ \frac{c^p}{2} & i = n_X + 1, j \le n_Y, \\ \frac{c^p}{2} & i \le n_X, j = n_Y + 1, \\ 0 & i = n_X + 1, j = n_Y + 1, \end{cases} \quad (2)$$

and

$$e_{X,Y}(W)^p = \frac{\epsilon^p}{2} ||A_X W_{1:n_X,1:n_Y} - W_{1:n_X,1:n_Y} A_Y||, \quad (3)$$

where $W_{1:n_X,1:n_Y}$ is the matrix formed by the first n_X rows and the first n_Y columns of matrix W (e.g., removing the last row and column of W) and $||\cdot||$ is the component-wise 1-norm of a matrix.

With the relaxation of the binary constraints $W(i,j) \in \{0,1\}$ to $W(i,j) \ge 0, \forall i,j$, we obtain a relaxed version of the metric, which also satisfies the metric properties and can be computed in polynomial time using linear programming [18]. We also refer to this relaxed version of the metric as the graph GOSPA metric.

The graph GOSPA metric penalises node attribute errors for assigned nodes, the number of unassigned nodes (each with a cost $c^p/2$), and the number of edge mismatches. In particular, for two pairs of assigned nodes (two nodes in X and two nodes in Y), the edge mismatch penalty is ϵ^p multiplied by the absolute difference in the corresponding edge weights. In addition, each edge connecting an assigned node and an unassigned node creates a half-edge mismatch penalty of $\epsilon^p/2$ multiplied by the weight of the edge, see full details in [15].

III. Gaussian processes for graphs with graph GOSPA similarity

A. Gaussian processes

A Gaussian Process (GP) is a non-parametric Bayesian model over functions [5]. A GP can be fully specified by its mean $m(\cdot)$ and covariance function (also called kernel) $k(\cdot,\cdot)$ and can be written as $f \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$.

For a regression task, consider that we have a set of n data points, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{X}$ is the input data point and $y_i \in \mathbb{R}$ is its associated output. In GP regression, we assume that there is additive noise such that $y_i = f(\mathbf{x}_i) + \epsilon_i$, where $f(\mathbf{x}_i)$ is the function value of sample $\mathbf{x}_i$ and ϵ_i is a zero-mean Gaussian noise with variance σ^2, which is independent of other variables. Given the dataset and a new test point $\mathbf{x}_*$, GP regression enables us to estimate the associated output y_*, see [5] for details.

B. Kernel based on the graph GOSPA metric

A kernel is a function $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ that measures the similarity between elements of the space $\mathbb{X}$. Kernel functions like the radial basis function (RBF) kernel and the Matérn kernels are commonly used [5], but they are designed for a vector input, $\mathbf{x} \in \mathbb{R}^{\mathbb{N}}$. For graphs, there are kernels such as random walk kernels [19] or Weisfeiler-Lehman graph kernels [13]. An alternative is to define similarity functions that can work as kernels, but do not meet the above properties, for instance, a similarity function based on the GED [7, Chap. 5].

Here we introduce the similarity function based on the graph GOSPA metric. Let X and Y be two graphs, $p' \ge 1$, $\ell > 0$, a length scale hyperparameter, and a multiplication factor, $\lambda > 0$. We define the similarity function between two graphs based on the graph GOSPA metric $d_p^{(c,\epsilon)}(\cdot,\cdot)$ as

$$k\left(X,Y\right) = \lambda \exp\left(-\frac{d_p^{(c,\epsilon)}(X,Y)^{p'}}{\ell}\right). \quad (4)$$

As will be shown in Section III-C, this similarity function can be decomposed into its different components to provide clear interpretability of the results. Although similarity functions defined like this are not generally positive semidefinite [20], they can show suitable performance in practice [7], [9]. To improve the stability of the algorithm, we add a small positive number to the diagonal elements of $\mathbf{K}$. In addition, during the training process, if the resulting covariance matrix for a given choice of pre-selected hyperparameters (c, ϵ, p, p', λ) is not positive definite, these hyperparameters are discarded.

C. Decomposition of graph GOSPA similarity function

In this section, we present the decomposition of the graph GOSPA similarity function. We first review the graph GOSPA metric decomposition into different types of costs [15]. We know from the graph GOSPA metric that $D_{X,Y}(i,j)$ represents the following costs:

1) Node attribute (localisation) error for assigned nodes, if $i \le n_X$, $j \le n_Y$.
2) Missed node cost if $i \le n_X$, $j = n_Y + 1$.
3) False node cost if $i = n_X + 1$, $j \le n_Y$.

The sets of indices (i,j) that belong to each of the previously mentioned categories are denoted by $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$. Therefore, for a given assignment matrix W, we have the following costs:

node attribute (localisation) cost, number of missed nodes cost, and number of false nodes cost. They are

$$\mathrm{l}(X,Y,W)^p = \sum_{(i,j)\in\mathcal{S}_1} D_{X,Y}(i,j)W(i,j) \tag{5}$$

$$\mathrm{m}(X,Y,W)^p = \frac{c^p}{2} \sum_{(i,j)\in\mathcal{S}_2} W(i,j) \tag{6}$$

$$\mathrm{f}(X,Y,W)^p = \frac{c^p}{2} \sum_{(i,j)\in\mathcal{S}_3} W(i,j). \tag{7}$$

Let W^* denote the optimal assignment in (1). Then, the graph GOSPA metric can be written as

$$d_p^{(c,\epsilon)}(X,Y) = \Bigg(\mathrm{l}(X,Y,W^*)^p + \mathrm{m}(X,Y,W^*)^p + \mathrm{f}(X,Y,W^*)^p + e_{X,Y}(W^*)^p\Bigg)^{1/p}. \tag{8}$$

Therefore, the graph GOSPA similarity function for $p' = p$ can be written as the product over the similarity functions for node attribute errors, number of missed nodes, number of false nodes and edge mismatches

$$k\left(X,Y\right) = k_{\mathrm{l}}\left(X,Y\right) k_{\mathrm{m}}\left(X,Y\right) k_{\mathrm{f}}\left(X,Y\right) k_{\mathrm{e}}\left(X,Y\right) \tag{9}$$

where

$$k_{\mathrm{l}}\left(X,Y\right) = \lambda \exp\left(-\frac{\mathrm{l}(X,Y,W^*)^p}{\ell}\right) \tag{10}$$

$$k_{\mathrm{m}}\left(X,Y\right) = \lambda \exp\left(-\frac{\mathrm{m}(X,Y,W^*)^p}{\ell}\right) \tag{11}$$

$$k_{\mathrm{f}}\left(X,Y\right) = \lambda \exp\left(-\frac{-\mathrm{f}(X,Y,W^*)^p}{\ell}\right) \tag{12}$$

$$k_{\mathrm{e}}\left(X,Y\right) = \lambda \exp\left(-\frac{e_{X,Y}(W^*)^p}{\ell}\right). \tag{13}$$

It is also possible to merge the similarities for the missed and false nodes into a single similarity score for unassigned nodes, given by the product of these two similarities $k_{\mathrm{u}}(X,Y) = k_{\mathrm{m}}(X,Y)k_{\mathrm{f}}(X,Y)$. For the cases that $p' \neq p$, the metric property of $d_p^{(c,\epsilon)}$ still holds, but $k\left(X,Y\right)$ cannot be decomposed as forms in (9).

IV. Experimental results

In this section, we first compare the Gaussian process based on the graph GOSPA similarity function with other Gaussian process models to make predictions on molecular properties in real datasets. Then, we illustrate the decomposition of the graph GOSPA similarity function applied to molecules.

A. Experimental setup

Datasets: In the experiments, we use 6 regression datasets, five from MoleculeNet [21], and one from [22]. Specifically, ESOL, FreeSolv, Lipophilicity and Photoswitch are datasets about the physical chemical properties of molecules and there is only one property to predict. QM8 is a dataset consisting of quantum mechanical properties. In this dataset, for numerical tractability of GPs, we only use a subset of the molecules by randomly sampling 2000 molecules from the full dataset and only consider the first 6 properties to predict.

The datasets are split into training and test sets with a ratio of 80/20. The graphs are obtained by converting the SMILES strings [23] into the corresponding molecular graphs.

Baselines: We compare the proposed method with GPs with the following kernels for molecules: Tanimoto kernel [24] using ECFP fingerprints [25], subsequence string kernel (SSK) [26] using SMILES [23] and WL kernel [13] using graphs with atom type as the node attributes. Shortest path kernel [27] for labelled graphs, neighbourhood hash kernel [28], edge histogram kernel and vertex histogram kernel [29].

Evaluation metrics: For ESOL, FreeSolv, Lipophilicity and Photoswitch datasets, we use the root mean square error (RMSE) to evaluate the performance. For QM8, we use mean absolute error (MAE), as this is the common choice in other papers for this dataset [21], [30].

Implementation details: All GP models are single-output GPs and the results are obtained by averaging over 20 random splits of the training and the test set. The node attribute in the GP model based on graph GOSPA similarity is the atom type. The base metric for node attributes of the graph GOSPA metric is $d(x,y) = 0$ if $x = y$, and $d(x,y) = c$ if $x \neq y$. All GPs are trained using the L-BFGS-B optimiser [31], except the graph GOSPA similarity, which is trained with the Adam optimiser [32] on the marginal log-likelihood with 2000 iterations. The learning rate is set to 0.001. The hyperparameters for graph GOSPA metric in the graph GOSPA similarity function are set to $c = 3$, $p = 2$, $p' = 1$, the value of ϵ is set based on the optimal value of marginal likelihood with grid search between [0,3] with step 0.2. The hyperparameter ℓ and λ for the graph GOSPA similarity function are optimised during the GP training process, the initial value is set to $\ell = 1, \lambda = 1$.

The models using graph GOSPA similarity, SSK kernel and Tanimoto kernel are implemented in GPflow[1] [33]. The WL kernel, shortest path kernel, neighbourhood hash kernel, edge histogram kernel and vertex histogram kernel are obtained from functions in the GraKeL library [34]. The GP models for these graph kernels are using the implementation in the library GAUCHE [14], which is implemented in GPytorch [12].

B. Results

Table I shows the results of the proposed methods and the baselines on the molecular datasets. The best results for each task are shown in bold, and the underlined values are the second-best results. From Table I, it can be observed that the proposed graph GOSPA similarity performs the best in the three datasets, and second best in the FreeSolv dataset.

In Table II, which contains the results of the QM8 dataset, the SSK kernel produces the best results followed by graph GOSPA and Tanimoto. Graph GOSPA performs the best among the algorithms that use a molecular graph as input.

[1]Code: https://github.com/JinhaoGu/GraphGOSPA-similarity.

TABLE I: Molecular property prediction over 4 physical chemical datasets.

	Dataset (RMSE ↓)			
Kernels	ESOL	FreeSolv	Lipophilicity	Photoswitch
SSK	0.66 ± 0.02	**1.34 ± 0.03**	0.73 ± 0.01	26.62 ± 1.07
Tanimoto	1.02 ± 0.02	1.88 ± 0.13	0.76 ± 0.01	23.42± 0.80
WL Kernel	0.75 ± 0.01	1.48 ± 0.04	0.74 ± 0.01	24.02 ± 0.65
Shortest Path Labelled	0.98 ± 0.01	2.41 ± 0.05	1.02 ± 0.02	43.58 ±7.11
Neighbourhood Hash	0.96 ± 0.05	1.82 ± 0.13	1.71 ± 0.18	33.62 ± 5.11
Edge Histogram	2.12 ± 0.02	3.94 ± 0.09	1.19 ± 0.01	66.76 ± 1.10
Vertex Histogram	1.12 ± 0.01	2.93 ± 0.07	1.09 ± 0.01	48.95 ± 1.52
Graph GOSPA	**0.66 ± 0.01**	1.37 ± 0.05	**0.70 ± 0.03**	**21.44 ± 0.68**

TABLE II: Molecular property prediction over a subset of 2000 molecules on the QM8 dataset. MAE values are scaled up by 10^2.

	Dataset (MAE ↓)					
Kernels	QM8 subset (scaled up by 10^2)					
	E1-CC2	E2-CC2	f1-CC2	f2-CC2	E1-PBE0	E2-PBE0
SSK	**1.41 ± 0.01**	**1.20 ± 0.02**	2.46 ± 0.06	4.09 ± 0.04	**1.29 ± 0.01**	2.34 ± 0.05
Tanimoto	1.41 ± 0.01	1.36 ± 0.02	**2.45 ± 0.05**	3.98 ± 0.06	1.47 ± 0.01	**2.29 ± 0.05**
WL kernel	2.76 ± 0.02	1.99 ± 0.01	2.83 ± 0.03	4.23 ± 0.03	3.10 ± 0.02	2.51 ± 0.02
Shortest Path Labelled	2.94 ± 0.02	2.13 ± 0.01	2.93 ± 0.02	4.45 ± 0.26	3.27 ± 0.02	2.67 ± 0.02
Neighbourhood Hash	2.97 ± 0.03	2.26 ± 0.04	3.62 ± 0.26	4.57 ± 0.08	3.29 ± 0.05	2.81 ± 0.09
Edge Histogram	3.61 ± 0.03	2.66 ± 0.02	3.17 ± 0.03	4.66 ± 0.03	3.87 ± 0.03	3.18 ± 0.02
Vertex Histogram	3.24 ± 0.02	2.29 ± 0.02	2.97 ± 0.03	4.55 ± 0.03	3.55 ± 0.02	2.87 ± 0.02
Graph GOSPA	1.48 ± 0.01	1.29 ± 0.01	2.54 ± 0.06	**3.81 ± 0.05**	1.41 ± 0.02	2.44 ± 0.05

C. Decomposition of graph GOSPA similarity example

In this section, we illustrate how the graph GOSPA similarity function can be decomposed into different parts to quantify the similarity of different parts in a graph (node attributes, unassigned nodes and edge mismatches). For demonstration, we choose three molecules from the ESOL dataset, shown in Figure 1. We set the hyperparameters $c = 3$, $p' = p = 1$, $\epsilon = 0.8$, $\lambda = 1$, and ℓ has been set to the optimised value on the ESOL dataset, $\ell = 27.371$, see Section IV-A.

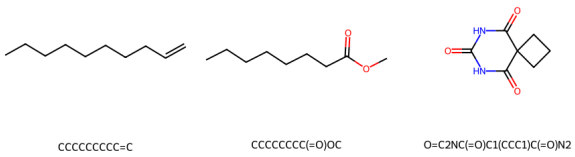
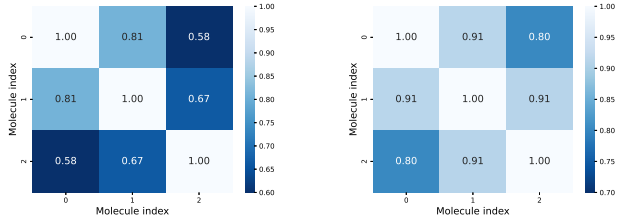


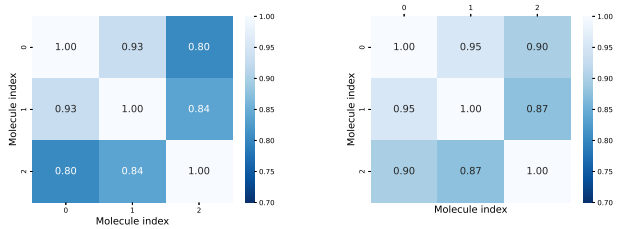
Fig. 1: Example molecules with their SMILES strings [23].

In Figure 2, we show the decomposition of the graph GOSPA similarity. Figure 2a shows the similarity matrix between the molecular graphs of the molecules in Figure 1. The indices 0, 1 and 2 in Figure 2 represent the molecules from left to right in Figure 1.

As can be seen in Figure 1, intuitively, the molecules become more different from left to right, being molecules 0 and 1 more similar than molecule 2. Therefore, the similarity decreases from molecule 0 to molecule 2. Figures 2b, Figure 2c, and Figure 2d show the decomposition of total similarity. In Figure 2b, the matrix shows the similarity in the node elements. By looking at the first row, we can see that molecule 0 is more similar in node elements to molecule 1 than to molecule 2. Figure 2c shows the similarity of the unassigned nodes. Again, molecule 0 is more similar to molecule 1 than to molecule 2 since they have a higher number of assigned nodes. Finally, Figure 2d shows the decomposition for edge similarity between graphs. Again, molecule 0 is more similar



(a) Similarity matrix of graphs. (b) Similarity matrix of node elements.



(c) Similarity matrix of unassigned nodes. (d) Similarity matrix of edges.

Fig. 2: Plots of the decomposition of the graph GOSPA similarity ($\lambda = 1$) across the three molecules in Figure 1. (a) Similarity of graphs; (b) similarity of node elements; (c) similarity of unassigned nodes; (d) similarity of edges.

to molecule 1 than to molecule 2, since they have fewer edge mismatches.

V. Conclusion

In this paper, we have proposed a Graph GOSPA similarity function, which measures graph similarity in an interpretable manner based on the graph GOSPA metric. The interpretability that comes from the similarity decomposition is an important characteristic, as it helps identify the similar/different aspects between two graphs. We have also introduced a GP model based on the Graph GOSPA similarity, which is able to learn both node and structural features in graphs by measuring differences in node attributes, number of unassigned nodes, and edge mismatches.

Finally, we have evaluated the proposed Graph GOSPA GP on various molecular property prediction datasets. Experimental results demonstrate that Graph GOSPA GP has better performance than the baselines in a number of datasets, and closely follows the best-performing algorithms when it does not provide the best results.

Acknowledgement

This work was supported by the EPSRC Centre for Doctoral Training in Distributed Algorithms EP/S023445/1. This work made use of the Barkla High Performance Computing facilities at the University of Liverpool.

References

[1] A. Christidis, C. Papaioannidis, I. Mademlis, and I. Pitas, "Lightweight human gesture recognition using multimodal features," in *2024 32nd European Signal Processing Conference (EUSIPCO)*, 2024, pp. 977–981.

[2] E. Rodrigo-Bonet and N. Deligiannis, "Physics-guided graph convolutional deep equilibrium network for environmental data," in *2024 32nd European Signal Processing Conference (EUSIPCO)*, 2024, pp. 987–991.

[3] H. Tang, X. Liu, Y. Rong, and D. Bliss, "Graph based method for enhanced heart rate trace estimation using uwb impulse radar," in *2023 31st European Signal Processing Conference (EUSIPCO)*, 2023, pp. 775–779.

[4] A. T. Balaban, "Applications of graph theory in chemistry," *Journal of chemical information and computer sciences*, vol. 25, no. 3, pp. 334–343, 1985.

[5] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, Mass: MIT Press, 2006.

[6] G. Nikolentzos, G. Siglidis, and M. Vazirgiannis, "Graph kernels: A survey," *Journal of Artificial Intelligence Research*, vol. 72, pp. 943–1027, 2021.

[7] M. Neuhaus and H. Bunke, *Bridging the gap between graph edit distance and kernel machines*. World Scientific, 2007, vol. 68.

[8] A. Sanfeliu and K.-S. Fu, "A distance measure between attributed relational graphs for pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 3, pp. 353–362, 1983.

[9] S. Boughorbel, J.-P. Tarel, and F. Fleuret, "Non-Mercer kernels for SVM object recognition." in *BMVC*. Citeseer, 2004, pp. 1–10.

[10] Z. Zeng, A. K. Tung, J. Wang, J. Feng, and L. Zhou, "Comparing stars: On approximating graph edit distance," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 25–36, 2009.

[11] H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 321–328.

[12] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "Gpytorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration," in *Advances in Neural Information Processing Systems*, 2018.

[13] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-Lehman graph kernels," *Journal of Machine Learning Research*, vol. 12, no. 77, pp. 2539–2561, 2011.

[14] R.-R. Griffiths, L. Klarner, H. Moss, A. Ravuri, S. Truong, Y. Du, S. Stanton, G. Tom, B. Rankovic, A. Jamasb *et al.*, "GAUCHE: A library for Gaussian processes in chemistry," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[15] J. Gu, A. F. García-Fernández, R. E. Firth, and L. Svensson, "Graph GOSPA metric: A metric to measure the discrepancy between graphs of different sizes," *IEEE Transactions on Signal Processing*, vol. 72, pp. 4037–4049, 2024.

[16] A. S. Rahmathullah, A. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *2017 20th International Conference on Information Fusion (Fusion)*, 2017, pp. 1–8.

[17] R. J. Trudeau, *Introduction to Graph Theory*. Dover Publications, Inc., 1993.

[18] L. Khachiyan, "Polynomial algorithms in linear programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 1, pp. 53–72, 1980.

[19] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," *The Journal of Machine Learning Research*, vol. 11, pp. 1201–1242, 2010.

[20] J.-P. Vert, "The optimal assignment kernel is not positive definite," *arXiv preprint arXiv:0801.4061*, 2008.

[21] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: a benchmark for molecular machine learning," *Chemical science*, vol. 9, no. 2, pp. 513–530, 2018.

[22] R.-R. Griffiths, J. L. Greenfield, A. R. Thawani, A. R. Jamasb, H. B. Moss, A. Bourached, P. Jones, W. McCorkindale, A. A. Aldrick, M. J. Fuchter *et al.*, "Data-driven discovery of molecular photoswitches with multioutput gaussian processes," *Chemical Science*, vol. 13, no. 45, pp. 13 541–13 551, 2022.

[23] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[24] L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi, "Graph kernels for chemical informatics," *Neural networks*, vol. 18, no. 8, pp. 1093–1110, 2005.

[25] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, 2010.

[26] H. B. Moss, D. Beck, J. González, D. S. Leslie, and P. Rayson, "BOSS: Bayesian optimization over string spaces," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[27] K. Borgwardt and H. Kriegel, "Shortest-path kernels on graphs," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, 2005, pp. 8 pp.–.

[28] S. Hido and H. Kashima, "A linear-time graph kernel," in *2009 Ninth IEEE International Conference on Data Mining*, 2009, pp. 179–188.

[29] M. Sugiyama and K. Borgwardt, "Halting in random walk kernels," *Advances in neural information processing systems*, vol. 28, 2015.

[30] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," in *International Conference on Learning Representations*, 2020.

[31] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, 1989.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman, "GPflow: A Gaussian process library using TensorFlow," *Journal of Machine Learning Research*, vol. 18, no. 40, pp. 1–6, apr 2017.

[34] G. Siglidis, G. Nikolentzos, S. Limnios, C. Giatsidis, K. Skianis, and M. Vazirgiannis, "Grakel: A graph kernel library in Python," *Journal of Machine Learning Research*, vol. 21, no. 54, pp. 1–5, 2020.