Fused Gromov-Wasserstein Alignment for Graph Edit Distance Computation and Beyond (Appendix)

1 Experimental Setup for Graph Alignment

Dataset		# Nodes	# Edges	# Node Types	# Edge Types
Douban [17]	-online	3906	8164	538	2
	-offline	1118	1511	538	2
Movie [6]	-Allmovie	6011	124709	14	1
	-Imdb	5713	119073	14	1
Megadiff changes [6]	-before	2433	4716	17	4
	-after	2488	4835	17	4

Table 1: Statistics of real-world datasets for graph alignment experiments.

Metrics. Four quantitative metrics are used in the evaluation. Given an alignment problem on $G_0 = \{V_0, E_0\}$ and $G_1 = \{V_1, E_1\}$, the ground-truth $\mathcal{A}^*$, and an alignment $\mathcal{A}$ returned by an algorithm, the metrics are defined as follows. **Accuracy (ACC)** is computed as

$$ACC = \frac{|\mathcal{A} \cap \mathcal{A}^*|}{|\mathcal{A}^*|}.$$

Mean Average Precision (MAP) is defined as

$$MAP = \frac{1}{|\mathcal{A}^*|} \cdot \sum_{(v_0,v_1)\in\mathcal{A}^*} \frac{1}{r(v_0,v_1)},$$

where

$$r(v_0,v_1) = 1 + |\{(v_0,u_1) \mid score(v_0,u_1) > score(v_0,v_1)\}|.$$

Edge Correctness (EC) measures the ratio of G_0's edges conserved by $\mathcal{A}$, omitting edge types, and is given by

$$EC = \frac{|\widehat{E}_0(\mathcal{A} \mid G_0, G_1)|}{|E_0|},$$

where

$$\widehat{E}_0(\mathcal{A} \mid G_0, G_1) = \{e_0 \mid (u_0,u_1) \in \mathcal{A}, (v_0,v_1) \in \mathcal{A}, e_0 = (u_0,v_0) \in E_0, e_1 = (u_1,v_1) \in E_1\}.$$

Induced Conserved Structure (ICS) is the ratio of edges in the G_1's induced subgraph on $V_1(\mathcal{A})$ conserved by $\mathcal{A}$, omitting edge types, and is defined as

$$ICS = \frac{|\widehat{E}_1(\mathcal{A} \mid G_0, G_1)|}{|E_1(V_1(\mathcal{A}))|},$$

where

$$\widehat{E}_1(\mathcal{A} \mid G_0, G_1) = \{e_1 \mid (u_0, u_1) \in \mathcal{A}, (v_0, v_1) \in \mathcal{A}^*, e_0 = (u_0, v_0) \in E_0, e_1 = (u_1, v_1) \in E_1\}$$

and

$$E_1(V_1(\mathcal{A})) = \{e_1 \mid e_1 = (u_1, v_1) \in E_1, u_1 \in V_1(\mathcal{A}), v_1 \in V_1(\mathcal{A})\}.$$

Dataset. Three datasets from different domains are used in the experiments; see Table 1 for basic statistics of these datasets.

- **Douban** [17]: This dataset comprises two social networks (online and offline) extracted from Douban. The offline network is a subgraph of the online network. In both networks, vertices represent distinct individuals, and edges represent relationships between them. There are 538 vertex types, each representing a physical area a person comes from. Two edge types exist: ”contacts” and ”friends,” derived from the frequency of two persons’ co-occurrence in offline social activities.
- **Movie** [6]: This dataset contains two film-related heterogeneous graphs, Allmovie and Imdb. The former is extracted from the Rotten Tomatoes website, and the latter from the Imdb website. Vertices represent films, and edges represent relationships between films. There are 14 vertex types, each denoting a genre of film. Only one edge type exists: an edge between two films indicates they have at least one common actor.
- **Megadiff changes** [6]: This software development dataset contains 10 diff files with 5-40 code change lines from Megadiff for evaluation. For each selected file, we transformed the before- and after-change source code into two code property graphs, and manually constructed ground-truth alignments by examining how the before-change code was transformed into after-change code. In these graphs, vertices denote program elements in source code, and edges denote relationships between program elements. The dataset has 17 vertex types and 4 edge types. Performance on Megadiff is measured by averaging results across the 10 samples.

Baselines. We compare FGWAlign with following six graph alignment methods and follow the same experimental protocol as the current state-of-the-art GABoost approach [6].

- **FINAL** [17]: An optimization-based method defining a quadratic objective on graph structure and attributes, solved for its stationary point as the alignment.
- **REGAL** [3]: A spectral approach using matrix factorization to generate vertex embeddings, aligning vertices by Euclidean distance.
- **WAlign** [2]: An unsupervised method that learns vertex embeddings with a lightweight GCN, using Wasserstein distance to evaluate embedding quality.
- **NAME** [4]: A supervised method combining skip-gram and GCN to capture local structure, high-order proximity, and global community features in vertex embeddings.
- **GTCAlign** [16]: An unsupervised method leveraging vertex centrality for similarity computation in GCN, with a refinement step.
- **SLOTAlign** [15]: An unsupervised approach framing alignment via Gromov-Wasserstein distance, alternately optimizing alignment and distance matrices.
- **GABoost** [6]: A parameter-free boosting algorithm that iteratively refines an initial alignment between two heterogeneous graphs. GABoost can be sequentially composed with other alignment methods to enhance their output; here, we apply it to improve SLOTAlign.

Dataset	# Train Graphs)	# Test Graphs)	# Nodes (avg.)	# Edges (avg.)
PROTEINS	360	223	39.1	72.8
ENZYMES	400	120	32.6	62.1
AIDS	1280	400	15.7	16.2
BZR	69	81	35.8	38.4
DD	390	236	284.3	715.7
NCI1	1646	822	29.8	32.3

Table 2: Statistics of datasets for graph-level anomaly detection experiments.

2 Experimental Setup for Graph-level Anomaly Detection

Task Definition. Graph-level anomaly detection (GLAD) aims to recognize anomalous graphs from a set of graphs by predicting an anomaly score for each graph sample that indicates its degree of abnormality [8, 12, 18]. Given a training set $\mathcal{G}_{train} = \{G_1, G_2, \ldots, G_M\}$ consisting only of normal graphs, and a test set $\mathcal{G}_{test} = \{G_{m+1}, G_{m+2}, \ldots, G_{M+N}\}$ containing both normal and anomalous graphs, GLAD seeks to learn a function $f : \mathcal{G} \rightarrow \mathbf{R}$ that maps each graph in $\mathcal{G}_{test}$ to an anomaly score such that anomalous graphs receive higher scores than normal graphs. The goal is to identify graphs whose structural patterns, topological properties, or attribute distributions deviate significantly from the normal graphs observed during training.

To apply FGWAlign on GLAD, we implement a nearest neighbor anomaly detector, which defines the anomaly score of a test graph as its distance to the top-1 nearest neighbor in the training set. We use GED computed by FGWAlign as our distance metric. Since computing GED between a test graph and all training graphs ($|\mathcal{G}_{train}|$ computations) is expensive for large datasets, we employ a pivot selection technique [1] to identify representative graphs in $\mathcal{G}train.$ First, we select k pivot graphs $G_1^p, G_2^p, \ldots, G_k^p$ from $\mathcal{G}_{train}$, where G_1^p is randomly selected and each subsequent G_i^p has the maximum nearest neighbor distance to the previously selected pivots. Then, we pre-compute the GED between each training graph and these k pivot graphs, resulting in a k-dimensional distance vector for each training graph. For each test graph, we similarly compute its k-dimensional distance vector by measuring GEDs to the pivot graphs, identify another k training graphs with the most similar distance vectors (using the ℓ_1 norm), and use the minimum GED to these graphs as the anomaly score.

Datasets. We evaluate the anomaly detection performance of FGWAlign on six TU datasets [10], following the experimental setup in [7, 8]. Detailed statistics of these datasets are provided in Table 2. Among them, DD, ENZYMES, and PROTEINS represent macromolecular structures. In these datasets, proteins are modeled as graphs where nodes correspond to secondary structure elements (such as helices, sheets, or turns) and are annotated with their respective types as well as various physical and chemical properties. Edges are established between nodes if they are adjacent in the amino acid sequence or are among the three nearest neighbors in three-dimensional space. The ENZYMES dataset is derived from the BRENDA database [13], with each enzyme labeled by one of six EC top-level classes that reflect the type of catalyzed chemical reaction. Similarly, the PROTEINS dataset models proteins as graphs and the task is to predict whether a protein functions as an enzyme. The DD dataset uses the same source data but represents each amino acid as a node, with edges indicating spatial proximity between amino acids. Further details on other datasets can be found at `https://chrsmrrs.github.io/datasets/docs/datasets/`.

Baselines. We compare our approach with four state-of-the-art learning-based GLAD methods: SIGNET [7], OCGIN [18], GLocalKD [8], and OCGTL [12]. All these methods require training a graph neural network encoder to perform predictions. In addition, we include classic non-learning baselines, which combine a graph kernel (either the Weisfeiler-Lehman kernel (WL) [14] or the Propagation kernel (PK) [11]) with a detector (iForest (iF) [5] or one-class SVM (OCSVM) [9]).

Implementation Details. We report the ROC-AUC with respect to anomaly scores and labels [18] for performance evaluation. All experiments are repeated five times, and we report the average performance. For learning-based methods, we follow the hyperparameter search procedure described in [7]. For our proposed FGWAlign anomaly detector, we use the same hyperparameters of FGWAlign$_{\text{fast}}$ in our main experiments and set the pivot number to $k = 20$ without further tuning.

References

[1] B. Bustos, G. Navarro, and E. Chavez. Pivot selection techniques for proximity searching in metric spaces. In *SCCC 2001. 21st International Conference of the Chilean Computer Science Society*, pages 33–40, 2001.

[2] Ji Gao, Xiao Huang, and Jundong Li. Unsupervised graph alignment with wasserstein distance discriminator. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 426–435, 2021.

[3] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. Regal: Representation learning-based graph alignment. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 117–126, 2018.

[4] Thanh Trung Huynh, Chi Thang Duong, Thanh Tam Nguyen, Vinh Tong Van, Abdul Sattar, Hongzhi Yin, and Quoc Viet Hung Nguyen. Network alignment with holistic embeddings. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):1881–1894, 2021.

[5] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[6] Wei Liu, Wei Zhang, Haiyan Zhao, and Zhi Jin. Gaboost: Graph alignment boosting via local optimum escape. *Proc. ACM Manag. Data*, 2(4), September 2024.

[7] Yixin Liu, Kaize Ding, Qinghua Lu, Fuyi Li, Leo Yu Zhang, and Shirui Pan. Towards self-interpretable graph-level anomaly detection. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[8] Rongrong Ma, Guansong Pang, Ling Chen, and Anton van den Hengel. Deep graph-level anomaly detection by glocal knowledge distillation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 704–714, 2022.

[9] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154, 2001.

[10] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML Workshop*, 2020.

[11] Marion Neumann, Roman Garnett, Christian Bauckhage, and Kristian Kersting. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102:209–245, 2016.

[12] Chen Qiu, Marius Kloft, Stephan Mandt, and Maja Rudolph. Raising the bar in graph-level anomaly detection. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2196–2203, 7 2022.

[13] I Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, and D. Schomburg. Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Research*, 32(Database-Issue):431–433, 2004.

[14] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

[15] Jianheng Tang, Weiqi Zhang, Jiajin Li, Kangfei Zhao, Fugee Tsung, and Jia Li. Robust attributed graph alignment via joint structure learning and optimal transport. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1638–1651, 2023.

[16] Chenxu Wang, Peijing Jiang, Xiangliang Zhang, Pinghui Wang, Tao Qin, and Xiaohong Guan. Gtcalign: Global topology consistency-based graph alignment. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[17] Si Zhang and Hanghang Tong. Final: Fast attributed network alignment. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1345–1354, 2016.

[18] Lingxiao Zhao and Leman Akoglu. On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights. *Big Data*, 2021.