

Shape-Graph Matching Network (SGM-net): Registration for Statistical Shape Analysis

Shenyuan Liang, Anuj Srivastava

Department of Statistics

Florida State University, Tallahassee, FL

sl20fu@fsu.edu

Mauricio Pamplona Segundo, Sudeep Sarkar

Department of Computer Science

University of South Florida, Tampa, FL

mauriciop@mail.usf.edu

Sathyanarayanan N. Aakur

Department of Computer Science

Auburn University, Auburn, AL

san0028@auburn.edu

Abstract—This paper focuses on the registration problem of shape graphs, where a shape graph is a set of nodes connected by articulated curves with arbitrary shapes. This registration requires optimization over the permutation group, made challenging by differences in nodes (in terms of numbers, locations) and edges (in terms of shapes, placements, and sizes) across graphs. We tackle this registration problem using a neural-network architecture with an unsupervised loss function based on the elastic shape metric for curves. This architecture results in (1) state-of-the-art matching performance and (2) an order of magnitude reduction in the computational cost relative to baseline approaches. We demonstrate the effectiveness of the proposed approach using both simulated data and real-world 2D retinal blood vessels and 3D microglia graphs.

Index Terms—shape graph registration, statistical shape analysis, quadratic assignment solver, matching network.

I. INTRODUCTION

Using shapes to characterize objects in images and videos is critical for object detection, tracking, and recognition tasks. Given tremendous variability between shapes within and across object classes, a **statistical** analysis of shapes becomes essential. There is a rich literature on various facets of shape analysis. The biggest challenge in shape analysis is the registration or matching of parts across objects during shape comparisons. Elastic approaches have addressed this need by incorporating registration as integral to shape comparisons. Such elastic techniques have been developed for several kinds of objects, from scalar functions and Euclidean curves to 2D surfaces, branching trees, networks, and graphs.

In this paper, we focus on specific type of objects called *shape graphs* and address the problem of *registering shape graphs under the elastic shape metric*. Shape graphs are objects that contain sets of Euclidean nodes connected by articulated edges. To compare the shapes of two such graphs, one must account for their node connectivity and the presence and shapes of edges, making registration challenging. Classical optimization tools (see, e.g., Guo et al. [1]) are severely limited by the computational cost that grows exponentially with number of nodes. This paper explores a deep-learning-based paradigm that promises faster solutions and state-of-the-art performance. Several recent papers use deep networks for graph matching, but they are mainly limited to regular (Euclidean) graphs and do not deal with shape graphs. The main **contributions** of this paper are: (1) introducing the

Shape Graph Matching Network (SGM-net) for shape graph registration, (2) handling graphs with arbitrary cardinality, and (3) demonstrating the performance and efficiency gains of our proposed method over current benchmarks.

II. PROPOSED FRAMEWORK

Here we present the proposed deep neural-network, SGM-net, for registering pairs of shape graphs.

A. Shape graph representation

A shape graph is a collection of Euclidean nodes with some pairs connected by articulated curves. We represent a shape graph by a pair of variables: an adjacency matrix to capture its edge attributes and a node matrix to represent its node attributes. If the graph has n nodes, then its adjacency matrix is $A \in \mathcal{S}^{n \times n}$, where $\mathcal{S}$ is the shape space of curves [2], and $A_{i,j}$ denotes the shape of edge starting from node i and ending at node j , $i \neq j$. If there is no edge, then $A_{i,j} = \mathbf{0}$ denoting a null edge ($\mathbf{0} \in \mathcal{S}$). We also define $\tilde{A} \in \mathcal{S}^{n \times n}$, where $\tilde{A}_{i,j}$ is simply a reflection re-parameterization of curve in $A_{i,j}$. Let each node i be assigned a vector attribute $u_i \in \mathbb{R}^k$ and let $u \in \mathbb{R}^{n \times k}$ be the node attributes of all n nodes. Then, the pair (A, u) specifies a graph G .

To reduce algorithmic complexity, we define *graph connectivity matrices* [3] denoted by $C \in \mathbb{R}^{n \times n_e}$ and $F \in \mathbb{R}^{n \times n_e}$ as follows. Let $s = 1, \dots, n_e$ index the non-zero edges in G . If there is an actual edge $A_{i,j}$ originating from node i , then the entry $(i, n(i-1) + j)$ in C is set to one, and zero otherwise. F is the same, except it uses the edges in $\tilde{A}$ instead of A .

Shape Graph Metric and Registration: Next, we define a metric for comparing two shape graphs $G = (A, u)$ and $G' = (A', u')$. At first, assume that the two graphs have the same number of nodes, say n . (We will relax this condition later in this section). The number of edges in G and G' can differ and are n_e and n'_e , respectively. The edge metric d_e is given by: $d_e(A_{i,j}, A'_{i,j}) = \min\{d_s(A_{i,j}, A'_{i,j}), d_s(\tilde{A}_{i,j}, A'_{i,j})\}$, where d_s is the shape metric between planar curves [2]. Additionally, let the node distance matrix to be $D \in \mathbb{R}^{n \times n}$ such that $D_{ij} = \|u_i - u'_j\|$, for any $u_i, u'_j \in \mathbb{R}^2$, where $\|\cdot\|$ is standard vector norm. The composite distance between two graphs G and G' is defined as:

$$d_g(G, G') = \lambda \left(\sum_{i,j} d_e(A_{i,j}, A'_{i,j}) \right) + (1 - \lambda) \text{Tr}(D),$$

where the parameter $0 < \lambda < 1$ controls the balance between the contributions of the nodes and edges.

Let $\mathcal{P}$ denote the set of all $n \times n$ permutation matrices. For any $P \in \mathcal{P}$, let $G * P$ represent the graph obtained by taking the nodes of G and re-ordering them according to P . The graph registration problem is then given by:

$$\hat{P} = \arg \min_{P \in \mathcal{P}} d_g(G, G' * P). \quad (1)$$

That is, we find a re-ordering of nodes of G' so that they are best matched with the nodes of G in terms of d_g .

Building Affinity Matrices: One can reformulate this problem using two large affinity matrices as follows. Define an edge-affinity matrix $K_e \in \mathbb{R}^{n_e \times n'_e}$ according to $K_e^{s,s'} = \lambda(1 - \frac{d_e(A_s, A'_{s'})}{\alpha})$, $\alpha = \max\{d_e(A_s, A'_{s'})\}$. An element $K_e^{s,s'}$ of this matrix compares the shape of edge s of G with the edge s' of G' . The closer this value is to one, the more similar the two shapes are. Similarly, the node affinity matrix $K_p \in \mathbb{R}^{n \times n'}$ is given by: $K_p^{i,j} = (1 - \lambda)(1 - \frac{D_{ij}}{\eta})$, $\eta = \max\{D_{ij}\}$.

Factorized formulation of QAP: The pairwise graph-matching problem (Eqn. 1) can be rewritten as a quadratic assignment problem (QAP) according to: $\max_{P \in \mathcal{P}} \text{vec}(P)^T K \text{vec}(P)$, where $K \in \mathbb{R}^{n^2 \times n^2}$ is called the *composite* affinity matrix between two graphs. Its diagonal and off-diagonal elements store the node and edge affinities across the graphs. Zhou et al. [3] showed that K can be factorized exactly as:

$$K = \text{diag}(\text{vec}(K_p)) + (C' \otimes C) \text{diag}(\text{vec}(K_e))(F' \otimes F)^T,$$

where C, F and C', F' are the graph connectivity matrices for G and G' , and $\otimes$ denotes the Kronecker product. We note that $(C' \otimes C), (F' \otimes F)^T$ are both of size $(n^2 \times n_e n'_e)$, $\text{diag}(\text{vec}(K_e))$ is of size $(n_e n'_e \times n_e n'_e)$ and $\text{diag}(\text{vec}(K_p))$ is of size $(n^2 \times n^2)$. Plugging the factorized formulation into QAP leads to an equivalent objective function:

$$\max_{P \in \mathcal{P}} \{\text{Tr}(K_p^T P) + \text{Tr}(K_e^T (C^T P C' \circ F^T P F'))\}, \quad (2)$$

where $\circ$ denotes the Hadamard product of matrices.

Using Null Nodes to Improve Registration: In case the graphs G, G' have different number of nodes, say n, n' , where $n \leq n'$, we can append G with $m = n' - n$ null nodes to bring them to the same size of n' . We incorporate the null nodes by extending the node distance matrix D to $\tilde{D}$ and node affinity matrix K_p to $\tilde{K}_p$ as follows. Set $\tilde{D} = \begin{bmatrix} D \\ \epsilon \mathbf{1}_{m \times n'} \end{bmatrix}$, where $\mathbf{1}_{(m \times n')}$ is an $(m \times n')$ matrix of ones. We use this $\tilde{D}$ to compute the larger node affinity matrix $\tilde{K}_p \in \mathbb{R}^{n' \times n'}$ as earlier. Here $\epsilon = \text{Tr}(D)/n$ is chosen to penalize matching of real nodes in G' with null nodes in G . We also extend the graph connectivity matrices $C, F \in \mathbb{R}^{n \times n_e}$ to $\tilde{C}, \tilde{F} \in \mathbb{R}^{n' \times n_e}$ accordingly. The entries in the appended parts equal a constant ξ , which is a small positive number that allows taking logarithms later.

B. SGM-net for shape graph matching

Motivated by the factorized formulation of [3], we propose a novel, deep network that solves the vertex classification using an association graph [4]. However, instead of using a very large affinity matrix (as used in [3]), this network takes six smaller matrices: $\tilde{K}_p \in \mathbb{R}^{n' \times n'}$, $K_e \in \mathbb{R}^{n_e \times n'_e}$, $\tilde{C}, \tilde{F} \in \mathbb{R}^{n' \times n_e}$, $C', F' \in \mathbb{R}^{n \times n'_e}$ as inputs and outputs a node registration matrix. This node registration is then used to register points along the matched edges. The overall architecture of the SGM-net is shown in Fig. 1, including both the training and inference procedures.

The proposed architecture provides several conceptual and practical advantages compared to a current SOTA. For instance, the *Neural Graph Matching Network* (NGM) of Wang et al. [4] builds the association graph directly from a computationally expensive (in space and time) composite affinity matrix K , which is of size $O(n'^2 \times n'^2)$. Thus, when applied to shape graphs, the NGM can only handle graphs of a maximum of 70 nodes or so on current hardware. Our approach uses smaller matrices and can handle much larger shape graphs. Besides, NGM primarily registers the node features. Our network takes in the edge shapes as additional information and jointly performs registration using both edge and node features.

1) *Matching aware embeddings of affinity matrices:* We set the initial edge-affinity embedding to be $E^{(0)}$, an $n'^2 \times l_0$ matrix, where each of the l_0 columns is a replicate of $\text{vec}(\log(\tilde{C} K_e C'^T \circ \tilde{F} K_e F'^T))$, l_0 is the initial affinity embedding size, $\circ$ is the Hadamard product, and $\log$ is elementwise. The intuition behind this equation is that if the edges in G and G' are similar, then the corresponding connected nodes in the association graph tend to have a high classification score.

We define an unweighted adjacency matrix $\tilde{A} = \tilde{C} \tilde{F}^T \otimes C' F'^T$, which is an $n'^2 \times n'^2$ sparse matrix of 1s and 0s. An entry in $\tilde{A}$ indicates if there is an edge between the corresponding two vertices in the association graph or not. We adapt the *Graph Attention Networks* (GATConv) [5] to construct the initial node embedding $V^{(0)} = \text{GATConv}(\tilde{K}_p, \tilde{A}, K_e) \in \mathbb{R}^{n'^2 \times l_0}$. The attention operator is applied to each node in the association graph, and the attention coefficients are computed based on K_e . This way different weights are assigned to each node, by aggregating the edge features from each node's neighbors.

The graph convolutional step updates node and edge embeddings iteratively according to:

$$V_i^{(k)} = f_u \left(\left[f_h([E^{(k-1)}, \tilde{A} V^{(k-1)}]_i), f_v(V^{(k-1)}) \right]_i \right), \quad (3)$$

$$E_i^{(k)} = f_h([E^{(k-1)}, \tilde{A} V^{(k-1)}]_i), \quad i = 1, 2, \dots, n'^2, \quad (4)$$

where $V^{(k)}, E^{(k)} \in \mathbb{R}^{n'^2 \times l_k}$ are node- and edge-affinity embeddings. The notation $[\cdot, \cdot]$ implies a concatenation of matrices. The message passing functions $f_h, f_u : \mathbb{R}^{2l_{k-1}} \rightarrow \mathbb{R}^{l_k}$ and $f_v : \mathbb{R}^{l_{k-1}} \rightarrow \mathbb{R}^{l_k}$ are all implemented by networks with two fully-connected layers and ReLU activation.

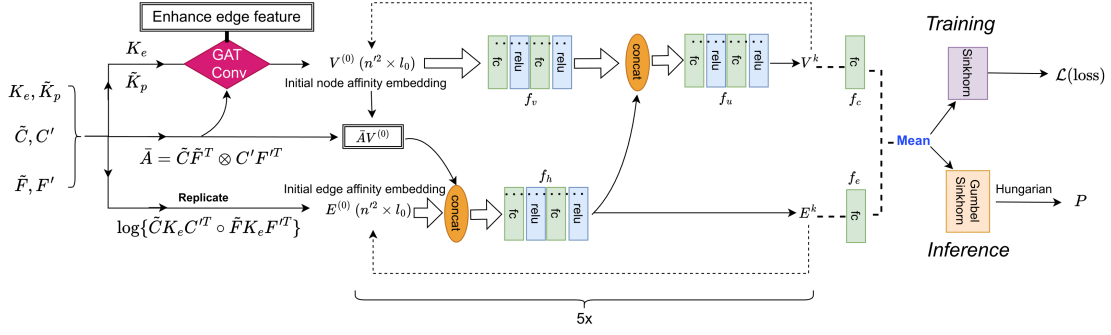


Fig. 1: **Overall architecture** for the SGM-net. With an unsupervised loss function, this architecture uses two flows that discover and aggregate the node- and edge-affinity features separately.

2) *Sinkhorn and Gumbel Sinkhorn network*: We utilize the Sinkhorn layer [6] to perform classification by turning the final embedding matrices $V^{(k)}$ and $E^{(k)}$, obtained in Eqn. 4, into a doubly stochastic matrix as follows: for $i = 1, 2, \dots, n'^2$, we compute $x_i = f_c(V_i^{(k)}) \in \mathbb{R}$, and $y_i = f_e(E_i^{(k)}) \in \mathbb{R}$, and set $S = \exp(\frac{x+y}{2\tau}) \in \mathbb{R}^{n'^2 \times 1}$, where $\exp$ is elementwise. Here $f_c, f_e: \mathbb{R}^{l_k} \rightarrow \mathbb{R}$ are both single fully-connected layers and τ is a normalization constant. After reshaping the classification score matrix S into $\mathbb{R}^{n' \times n'}$, the doubly-stochastic matrix can be obtained by element-wise division on rows and columns: $S = S \oslash (\mathbf{1}_{n'} \mathbf{1}_{n'}^T S)$, $S = S \oslash (S \mathbf{1}_{n'} \mathbf{1}_{n'}^T)$, where symbol $\oslash$ is element-wise division.

Similar to [6], the Gumbel Sinkhorn layer [4] is a post-selection technique that is applied at the inference stage. This layer enables searching over the space of all possible permutation matrices and uses the Hungarian algorithm to select the one with the highest objective score. The Gumbel Sinkhorn layer is the same as the Sinkhorn layer, except that now S is defined as $\exp(\frac{x+y+g}{2\tau})$, where g is sampled from standard Gumbel distribution with CDF $e^{-e^{-x}}$.

3) *Training Loss*: As shape graphs are constructed from real-world 3D images [7, 8], there is no ground truth information available for training. During the training stage, the network performs optimization on the loss function $\mathcal{L}(S)$ as given in Eqn. 2, with S replacing P . The objective score is the negative of the loss function.

III. EXPERIMENTS

Next we provide results on shape graph registration using the proposed SGM-net, and compare them to SOTA methods.

Real Datasets: The first is 2D Retinal Blood Vessel (RBV) dataset for networks extracted from colored fundus images [7, 8]. The second set is for the 3D Microglia dataset available on NeuroMorpho.Org [9]. **Synthetic Dataset**: The synthetic data is created by starting with actual 2D RBV networks and introducing noise, clutter, and distortions to generate paired shape graphs [1]. A total of 900 (registered) pairs are simulated for training. The synthetic data for 3D Microglia datasets is simulated similarly.

For each experiment, we collect simulated and real data to form 1800 pairs for training the network. For testing, we include 100 pairs of synthetic graphs and 100 pairs of real graphs. Note that for the real data, there is no ideal pairing available; we randomly pair these graphs. The training and testing data have no overlap.

Evaluation metrics. For quantitative evaluation, we use the graph shape metric d_g defined in Section II-A. The smaller the distance, the better the registration. Additionally, we compute the node objective score $\text{Tr}(\tilde{K}_p^T S)$, where a larger objective points to a better registration. For qualitative results, we visualize geodesics under estimated registrations. Given the availability of known ground truth in synthetic experiments, we naturally compare results with the ground truth.

State-of-the-art baselines: There are no existing DNNs to perform shape graph registration under the shape metrics. We have adapted three recent DNNs for shape-graph matching to enable comparisons. These are: **1. PCA-GM** [6] approximates the distribution of node embeddings using cross-graph convolutions. We adapt this approach to also learn how to weigh neighbors in convolutions based on the shape of edges. **2. CIE-H** [10] introduced a Hungarian Attention module that dynamically constructs a structured and sparsely connected layer, taking into account the most contributing matching pairs as hard attention during training. **3. NGM** [4] learns with Lawler's QAP with given affinity matrices. For a learning-free method, we also include a SOTA method **Factorized Graph Matching (FGM)** [3] with details as presented in [1].

Quantitative Results: We compute the shape graph distance before d_g^B and after registration d_g^R , and compute a relative graph-distance reduction as $\frac{d_g^B - d_g^R}{d_g^B}$. We also compute the node objective score which is defined earlier. The larger the distance reduction and node objective score, the better the registration. Table I shows the comparison between our proposed SGM-net and other recent graph-matching networks for 2D and 3D shape graphs, respectively. We can see that the SGM-net outperforms other deep neural networks in all experiments in terms of graph distance reduction and node objective score. Besides, the SOTA QAP solver NGM failed

2D Retinal Blood Vessels (RBVs)							
Type	Size		Method	Graph Dist. Reduc.(%)	Node Score	Time (s)	Speed Gain over FGM
	Node	Graph					
Real	45-65	1250-1850	CIE	39.34	21.11	0.07	2385 times
			PCA-GM	40.59	21.28	0.63	265 times
			NGM	43.25	21.30	1.25	133 times
			FGM	45.58	20.30	167	–
			Ours	47.20	21.31	0.7	240 times
Real	120-200	3500-4500	CIE	32.54	57.10	1.1	1018 times
			PCA-GM	34.24	57.66	3.31	338 times
			FGM	44.72	57.09	1120	–
			Ours	42.83	59.27	5.31	211 times
Synthetic	120-200	3500-5500	CIE	45.23	50.92	0.09	–
			PCA-GM	49.12	49.99	3.13	–
			Gnd.Truth	50.09	50.93	–	–
			Ours	52.84	50.93	5.18	–
			3D Microglia datasets				
Synthetic	80-120	1000-2000	CIE	53.43	33.31	0.09	–
			PCA-GM	53.31	33.31	0.21	–
			Gnd.Truth	53.63	33.33	–	–
			Ours	55.26	33.34	0.98	–
Real	80-120	1000-2000	CIE	25.95	29.41	0.08	3625 times
			PCA-GM	19.05	28.52	0.22	1318 times
			FGM	35.62	30.02	290	–
			Ours	34.93	31.08	1.26	230 times

TABLE I: Performance comparisons on 2D RBV and 3D Microglia datasets: (For graph distance reduction and node scores the larger, the better). The best results are in blue bold. For graphs with more than 70 nodes, **the NGM fails**.

to finish on the original Retinal Blood Vessel [7, 8], and Microglia datasets [9] (graphs with around 100 nodes) with an Nvidia RTX A100 (80GB) GPU. However, our method SGM-net finishes the experiments on the original datasets with any 24GB GPU, while improving performance. Between our SGM-net and the SOTA learning-free method FGM [3], they achieve similar performances in terms of graph distance reduction and node objective score. However, the biggest justification for using SGM-net comes in form of improved computational speed. SGM-net incurs much lower cost as listed in Table I. We can see after the training stage, the time gains for SGM-net over FGM are substantial. SGM-net is almost 220 times faster than FGM.

Geodesic deformations. the SGM-net can achieve performance similar to that with ground truth registration. At the halfway point (the bottom row), the ground truth and SGM-net registration largely preserve the salient structures and display a more natural deformation. However, The PCA-GM [6] and CIE-H [10] suffer from bad registration and provide distortion paths as shown in the red dashed box. NGM failed to run the experiments due to large memory needs.

IV. CONCLUSION AND DISCUSSION

This paper proposes a deep learning-based registration framework for handling shape graphs that appear in biological and anatomical data. This method uses unsupervised training and learning-based registration to provide state-of-the-art results and order-of-magnitude improvements in computational costs. The key distinction from the past methods is that it

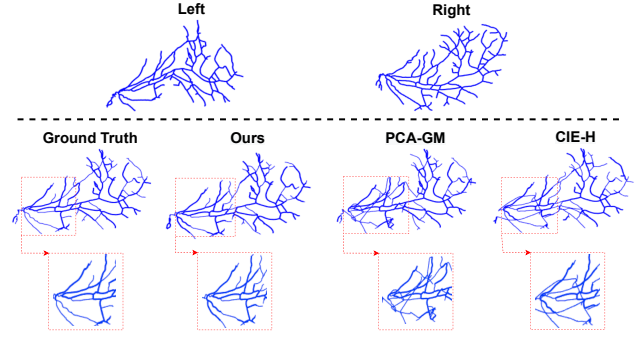


Fig. 2: Geodesics examples: The bottom row illustrates the halfway points of geodesic paths between the left and right shape graphs, as computed by different methods. Our approach shows a natural deformation of blood vessels.

incorporates the shape metrics for curves in comparing edges across graphs. It obtains speed gains of $\approx 200 - 300$ times the SOTA technique FGM.

COMPLIANCE WITH ETHICAL STANDARDS

This is a numerical simulation study for which no ethical approval was required.

ACKNOWLEDGMENT

This research was supported in part by the NSF grants NSF-IIS-1955154, IIS 2348689, IIS 2348690, and IIS 1956050. The authors are thankful to the producers of the datasets used here.

REFERENCES

- [1] X. Guo *et al.*, “Statistical shape analysis of brain arterial networks (ban),” *The Annals of Applied Statistics*, vol. 16, no. 2, pp. 1130–1150, 2022.
- [2] A. Srivastava *et al.*, *Functional and shape data analysis*. Springer, 2016, vol. 1.
- [3] F. Zhou *et al.*, “Factorized graph matching,” in *IEEE CVPR*. IEEE, 2012, pp. 127–134.
- [4] R. Wang *et al.*, “Neural graph matching network: Learning lawler’s quadratic assignment problem with extension to hypergraph and multiple-graph matching,” *IEEE TPAMI*, 2021.
- [5] P. Veličković *et al.*, “Graph attention networks,” in *ICLR*, 2018.
- [6] R. Wang *et al.*, “Learning combinatorial embedding networks for deep graph matching,” in *ICCV*, 2019.
- [7] J. Staal *et al.*, “Ridge-based vessel segmentation in color images of the retina,” *IEEE TMI*, vol. 23, pp. 501–509, 2004.
- [8] A. Hoover *et al.*, “Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response,” *IEEE TMI*, vol. 19, no. 3, pp. 203–210, 2000.
- [9] M. A. Akram *et al.*, “An open repository for single-cell reconstructions of the brain forest,” *Scientific data*, vol. 5, no. 1, pp. 1–12, 2018.
- [10] T. Yu *et al.*, “Learning deep graph matching with channel-independent embedding and hungarian attention,” in *ICLR*, 2020.