Unsupervised Learning of Graph Matching with Mixture of Modes via Discrepancy Minimization

Runzhong Wang, *Student Member, IEEE,* Junchi Yan, *Senior Member, IEEE* and Xiaokang Yang, *Fellow, IEEE*

Abstract—Graph matching (GM) has been a long-standing combinatorial problem due to its NP-hard nature. Recently (deep) learning-based approaches have shown their superiority over the traditional solvers while the methods are almost based on supervised learning which can be expensive or even impractical. We develop a unified unsupervised framework from matching two graphs to multiple graphs, without correspondence ground truth for training. Specifically, a Siamese-style unsupervised learning framework is devised and trained by minimizing the discrepancy of a second-order classic solver and a first-order (differentiable) Sinkhorn net as two branches for matching prediction. The two branches share the same CNN backbone for visual graph matching. Our framework further allows unsupervised learning with graphs from a mixture of modes which is ubiquitous in reality. Specifically, we develop and unify the graduated assignment (GA) strategy for matching two-graph, multi-graph, and graphs from a mixture of modes, whereby two-way constraint and clustering confidence (for mixture case) are modulated by two separate annealing parameters, respectively. Moreover, for partial and outlier matching, an adaptive reweighting technique is developed to suppress the overmatching issue. Experimental results on real-world benchmarks including natural image matching show our unsupervised method performs comparatively and even better against two-graph based supervised approaches.

Index Terms—Unsupervised Learning, Graph Matching, Graph Clustering, Image Matching

1 Introduction

Graph matching (GM) computes the node-to-node correspondences among two or multiple graphs, by utilizing the structural information in graphs. It has wide applications for real-world graph alignment across vision [1], social networks [2], knowledge graph [3], etc. GM is in general NP-hard [4], and the simplest form of graph matching namely **<u>two</u>**-**<u>g</u>**raph **<u>m</u>**atching (GM) has the general formulation namely Quadratic Assignment Problem (QAP) [5], [6], [7]. Other papers focus on the more challenging setting of jointly matching among multiple graphs, known as **<u>m</u>**ulti-**<u>g</u>**raph **<u>m</u>**atching (MGM) [8], [9], [10], [11], and even graphs from different categories (i.e. mixture of modes), known as **<u>m</u>**ulti-**<u>g</u>**raph **<u>m</u>**atching with a **<u>m</u>**ixture of **<u>m</u>**odes (MGM^3) [12]. Considering the rich, popular, and well-developed visual benchmarks and their practical importance towards downstream applications [1], [13], [14], this paper focuses on the application of graph matching on computer vision, by specifying typical models e.g. convolutional neural networks as the main model for learning. To verify the effectiveness of our approach in real-world problems, we further investigate the natural image matching problem which particularly involves the challenges of partial matching and outliers, as will be detailed later in the paper.

Traditional GM algorithms mainly focus on how to efficiently solve the underlying QAP via either continuous relaxation [5], [6], [7], [15], [16] or discrete search algorithms [17], [18], using pre-given affinity metric between nodes and edges, e.g. Gaussian kernels, whose expressiveness power can be restricted due to the limited model capacity and unlearnable nature. In fact, the proper formulation of the optimization problem, i.e. designing the affinity metric between graph nodes and edges, still remains an open problem. Advances in machine learning have inspired learnable affinity functions as well as the node/edge feature representations for graph matching especially in vision, from early shallow models [19], [20], [21] to recent deep neural networks [22], [23], [24].

A popular deep learning GM pipeline in literature usually include perception modules (e.g. VGG16 [25]), structural learning modules (e.g. GCN [26]), and affinity metrics (e.g. vector-space similarity). These deep learning modules have shown superior performance over fixed affinity metrics and representation (e.g. SIFT descriptor [27]) on challenging real-world benchmarks [21], [28], [29].

However, current supervised deep GM requires costly annotation on large-scaled training data, which restricts the real-world application of modern deep graph matching methods. Though appealing, developing an unsupervised learning algorithm on graph matching is non-trivial, because GM methods often involve indifferentiable discretization steps. Besides, unsupervised learning without ground truth labels is desirable yet still challenging for most machine learning tasks. Seeing that the recent success of unsupervised image classification models [30], [31] share a discrepancy minimization pipeline considering two branches of predictions from the same data, in this paper, we present a simple yet effective unsupervised learning pipeline whereby GM solvers offer different matching predictions w.r.t. the differentiable Sinkhorn branch. Both branches share the

- *R. Wang, J. Yan and X. Yang are with Department of Computer Science and Engineering, and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, 200240, P.R. China. R. Wang and J. Yan are also with Shanghai AI Laboratory, Shanghai, China.*
  E-mail: {runzhong.wang,xkyang,yanjunchi}@sjtu.edu.cn
  Correspondence author: Junchi Yan.
  Source code available at https://github.com/Thinklab-SJTU/ThinkMatch.

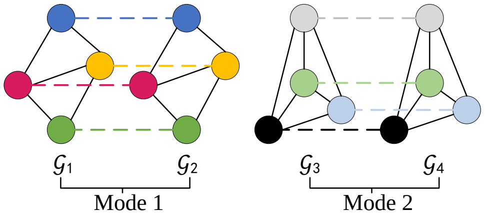
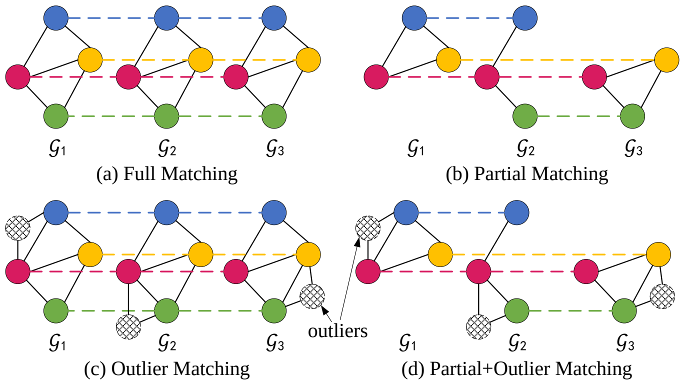


Fig. 1. Example of matching with a mixture of two modes. Within mode 1 or mode 2 (intra-mode) there is full-matching, and between mode 1 and mode 2 (inter-mode) the matching is non-valid.

same CNN backbone (in a Siamese-style [32]), resulting in a discrepancy minimization pipeline. Besides, GM solvers are probably more accurate because they utilize both node-wise and structural information in matching [9]. Without loss of generality, we resort to the classic graduated assignment algorithm dating back to [16] due to its adaptability to various graph matching settings as will be shown later in the paper. In MGM, the joint consideration of multiple graphs utilizes more information and can also be addressed by our unsupervised learning pipeline.

A realistic yet challenging setting is multi-graph matching from a mixture of modes (MGM^3), where graphs may belong to different groups and we need to jointly solve matching and clustering (see Fig. 1)[1]. This problem has seldom been considered apart from one loosely relevant work [12], which is learning-free, and clustering is performed after matching. In contrast, observing the fact that matching and clustering are inherently interleaved to each other, an MGM^3 approach is developed where clustering and matching are performed alternatively with gradually increased confidence. Furthermore, MGM^3 is also covered with our unified unsupervised learning framework, showing the potential of real-world applications of our method.

A preliminary version of this paper has appeared as a conference paper in [33] that deals with multiple graph matching and the mixture situation of graphs[2], and this journal version has further incorporated the classic two-graph matching case, as well as the more challenging and more general partial+outlier matching setting, and makes the following overall contributions:



(a) Full Matching

(b) Partial Matching

(c) Outlier Matching

(d) Partial+Outlier Matching

Fig. 2. Comparison of the four GM settings over multiple graphs (same color denotes correspondence): (a) the classical full matching where all node pairs have a valid matching, (b) partial matching where some inlier nodes are occluded and all nodes partially correspond to a universe (here the universe has 4 nodes), (c) outlier matching where the noisy outliers do not correspond to any nodes in the universe, (d) the mix of both partial and outlier matching, which is the most challenge yet the most realistic setting. Most existing papers [9], [23], [36], [37] only consider (a), some recent efforts [33], [35] also consider (b), and in this paper we propose a robust approach considering (a)(b)(d). (c) is less common because outliers usually appear together with partial matching.

1) We establish an unsupervised framework for deep GM, in contrast to the majority line of works based on supervised learning. Though our experiments are mainly conducted on existing benchmarks that often involve visual matching while the unsupervised framework itself is general and independent of the specific choice of each module.

2) Under the proposed framework, we develop GM solvers based on traditional learning-free solvers for the settings ranging from two-graph matching, multiple graph matching, to the general case that multi-graph matching from a mixture of modes. Moreover, our method benefits from its robustness to the partial matching and outlier matching cases, whereby an adaptive reweighting of matching affinities is proposed to suppress the overmatching issue in the presence of occluded inliers and outliers (see Fig. 2).

3) Evaluation is conducted in a variety of settings and on different benchmarks, showing a strong performance for both matching and clustering over graphs. In particular:

i) We compare our unsupervised method (without any finetuning using additional labels) directly with state-of-the-art supervised GM networks [34], [35], [36] on extensive benchmarks, and show that our method performs on par with and sometimes even outperforms supervised methods.

ii) We adapt the pretraining+finetuning paradigm to GM learning, by combining unsupervised pretraining (by our unsupervised technique) and supervised finetuning. Without bells and whistles, we improve the accuracy and convergence speed of state-of-the-art GM networks: BBGM [35] (from 56.7 to 60.7) and NGMv2 [36] (from 59.0 to 59.4).

iii) We apply our unsupervised technique to the downstream image matching pipeline involving keypoint detectors and stereo estimators, particularly further to the structure-from-motion (SfM) task. Our approach notably improves the state-of-the-art SuperGlue model [1] regarding both matching accuracy and SfM pose-estimation accuracy.

2 Related Works

We discuss three aspects closely related to our methods: i) unsupervised learning with self-correspondence, ii) graph

[1] In this paper we interchangeably use the terms “class”, “cluster” and “mode”, while “class” is mainly used in the context of adopting the cross-entropy loss. Recall that our approach is unsupervised thus we also use the terms “cluster” and “mode”.

[2] Compared with [33], the extensions include: i) an unsupervised framework for graph matching, and particularly for two-graph matching, by minimizing the discrepancy between the fine-grained second-order graduated assignment solver and the Sinkhorn network that only uses the node-wise features; ii) An improved version of graduated assignment algorithm with adaptive reweighting of matching pairs to suppress the overmatching in the presence of occluded inliers and outliers; iii) We propose a new graph matching learning paradigm in combination of unsupervised pretraining (by our presented unsupervised technique) and supervised finetuning; iv) Integration of the unsupervised deep graph matching model into a natural image matching pipeline involving keypoint detectors and downstream stereo estimators, with the new application of structure-from-motion task; v) Comprehensive experiments for our extended methods in comparison with more baselines, more problem settings and more datasets.

matching and its extension with a mixture of modes; iii) the downstream application of natural image matching.

2.1 Unsupervised Learning with Self-Correspondence

Unsupervised deep learning techniques emerge in both vision and graphs. The objective is often defined to minimize the discrepancy of the two branches' predictions from one input sample e.g. an image, whereby various discrepancy metrics are involved. In particular, many efforts have been devoted to preventing the collapse of the solution to a trivial constant one. One popular practice e.g. in SimCLR [30] is introducing negative samples which the contrastive loss tries to repulse. Another way is clustering e.g. in SwAV [38] which incorporates online clustering into contrastive learning. Other techniques like momentum encoder are adopted (see BYOL [31]) to enable negative-free unsupervised learning. Notably, [39] proposes to avoid model collapse by simply stopping the gradient of one out of the two branches for discrepancy minimization. Another line of related works addresses unsupervised learning on structural data e.g. graphs, which can be categorized as generative models exploiting the inherent dependency in graphs as a generative process [40], [41], [42]; and contrastive models learning the universal graph embedding by contrastive learning [43], [44], [45]. In summary, though there exist a considerable amount of works in unsupervised learning, while to our best knowledge, there are few ones for GM.

Cycle-consistency means that the propagated matchings should be consistent if the propagation path is a cycle, which is widely used in computer vision, especially for unsupervised or semi-supervised models [46]. In unsupervised correspondence learning [47], [48], the network is trained with a synthetically warped version of the original image to provide pseudo supervision signals for end-to-end training. While the warping can be limited for real-world image matching, and thus leads to the risk of poor generalization [49]. There are further efforts to create pseudo labels by augmenting images from the dataset [50]. Differently, we do not create any artificial version of the raw graph or image data to enforce correspondence consistency as a learning objective. Instead, we adopt two branches for predicting the matching solutions respectively, both based on the same learned features as input, and cycle-consistency is utilized for jointly matching multiple graphs to improve the prediction quality. Specifically, the loss function refers to the two predictions' discrepancy by cross-entropy.

2.2 Graph Matching and Match from a Mixture of Modes

Traditional methods directly solve the matching problem without a trainable model, and mostly they consider the setting for two graphs. Due to its NP-hard nature, different approximation techniques have been devised ranging from graduated assignment [16], [51], spectral matching [52], random walk [5], to projection-based methods [7], [53], etc. To handle the inherent illness in graph affinity modeling, especially in the presence of noise and outliers, multiple graphs are considered for joint matching [11], [54], [55], which not only mitigates the local ambiguity but also brings the problem into a more realistic setting. In many multi-graph matching works [8], [9], the cycle-consistency has served as an effective regularizer to achieve robustness against outliers and noise. See the survey [56] and the references therein for details about the traditional methods.

Learning for GM has received increasing attention [57], from early shallow models [19], [21] to modern deep neural networks [22]. The motivation is that the learned features can be more informative for matching, as the task is assumed to be different from recognition. Earlier works [58], [59] adopt CNNs and GNNs for visual appearance and structural information extraction, respectively. While the more recent work [60], [61] have shown how to perform joint graph structure learning and matching to lessen the reliance on predefined graph structure. In particular, the graph matching network GLAM [61] adopts a pure attention-based framework for both graph learning and graph matching and it achieves state-of-the-art performance on multiple public datasets. In fact, in existing deep learning-based models, the ground truth correspondences are needed for supervised training, which calls for extensive and even unrealistic labeling. In this paper, we take an initiative towards label-free unsupervised learning of deep GM.

In addition, we consider partial matching in the sense that only a subset of nodes can find their correspondences from other graphs, which is very common due to the existence of ubiquitous outliers and occlusions in visual images. There are a few works addressing this issue directly [54], [55], [62] or indirectly [9], [37]. Matching with outliers is another notable challenge in graph matching while existing approaches [63], [64], [65] mainly exploit certain structural patterns as priors e.g. motion, homography, pose, etc. In this paper, we firstly identify the overmatching issues that leads to partial and outlier-aware matching, and develop an orthogonal approach by reweighting the affinities.

One step further, considering the real-world scenario that the graphs for matching are not always from a single mode, i.e. the graphs are often from a mixture of modes, the seminal work [12] directly solves the joint matching and clustering task for graphs from multiple modes. In our preliminary version, we further devise a learning-based pipeline for this problem with significantly improved performance. In this extended work, we also cover the challenging case of matching from a mixture of modes.

2.3 Natural Image Matching

Graph matching can be used for natural image matching, which is a fundamental problem in computer vision. Its applications include structure-from-motion (SfM), simultaneous localization and mapping (SLAM), image registration, fusion, retrieval, etc. It involves several steps whereby each step has attracted wide attention. In this section, we mainly discuss the image matching pipeline, and readers are referred to the comprehensive survey [66] for more details.

In general, image matching involves 1) keypoint detection; 2) feature description; 3) matching; 4) consensus filtering. Traditional keypoint detectors include corner detectors [67], [68] and blob detectors [27]. Specifically, the well-known SIFT method [27] offers both feature detection and description. Recently, CNN-based feature detectors and descriptors are proposed [69], [70]. For the matching step, the naive nearest neighbor matching used to be popular, and

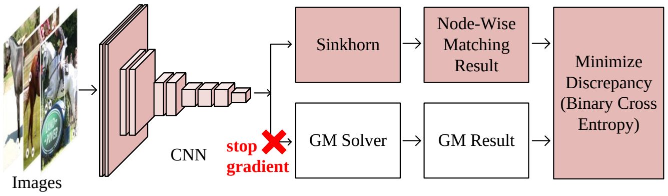

Images
CNN
stop gradient
Sinkhorn
Node-Wise Matching Result
GM Solver
GM Result
Minimize Discrepancy (Binary Cross Entropy)

Fig. 3. An overview of the proposed unsupervised (visual) GM learning pipeline. Here the GM solver in the white box can be any of the solvers for matching either two-graph, multi-graph, or graphs from a mixture of modes. The stop-gradient operation prohibits model degeneration [39].

some recent efforts [1] integrate the differentiable Sinkhorn algorithm [71] for matching. [72] devises a detector-free end-to-end network to tackle the image matching problem. Consensus filtering aims at discarding wrong matches, whereby RANSAC [63] and its variants are still well adopted. The graph matching papers surveyed in Sec. 2.1 are viewed as tackling special scenarios of image matching considering little about the partial and outlier-aware matching. In this paper, we also take a further step to resolve these challenges that arise in image matching.

3 The Discrepancy Minimization Framework for Unsupervised Graph Matching

We present a general unsupervised learning framework of deep graph matching by discrepancy minimization, where the training process requires no manual ground truth matching labels. Specifically, we consider three variants of graph matching problems, namely two-graph matching (GM), multi-graph matching (MGM), and multi-graph matching with a mixture of modes (MGM^3), also covering practical scenarios with partial matching and outliers. These settings cover most existing research works on graph matching problems to our best knowledge.

3.1 The General Paradigm

An overview of our unsupervised learning pipeline is shown in Fig. 3. In this paper, we consider $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m$, and we define n_i as the number of keypoints in $\mathcal{G}_i$ (including the outliers and unmatchable inliers for partial matching). The modules of our framework are as follows:

3.1.1 Feature Extraction

In this paper, our unsupervised GM learning framework is embodied by a visual matching pipeline, which accepts images as input. The images are processed by the feature extractor VGG16 [25] pretrained by ImageNet [73]. In line with peer models [22], [23], [34], the `relu4_2` and `relu5_1` feature maps are extracted from VGG16 and then concatenated. Inspired by ROI-Align in Mask R-CNN [74], the feature vectors at the keypoint positions are obtained by a namely feature align module which performs bi-linear interpolation on the feature map. For each image, $\mathbf{F}_i \in \mathbb{R}^{n_i \times l}$ denotes the l-dimensional feature of n_i nodes extracted by CNN.

3.1.2 Computing Node-Wise Matching via Sinkhorn

Firstly, node-wise matching is computed from the inner-product of node features in graph pair $\mathcal{G}_i, \mathcal{G}_j$:

$$\mathbf{W}_{ij} = \mathrm{Sinkhorn}(\mathbf{F}_i \mathbf{F}_j^\top, \tau_w) \tag{1}$$

where $\mathrm{Sinkhorn}(\mathbf{M}, \tau)$ is the popular Sinkhorn algorithm for matrix normalization [71]. Sinkhorn algorithm is viewed as the differentiable and approximate version of Hungarian algorithm [75]. Hungarian algorithm solves the linear assignment problem at $\mathcal{O}(n^3)$ time complexity:

$$\begin{aligned} &\max_{\mathbf{X}} \mathrm{tr}(\mathbf{X}^\top \mathbf{M}) \\ s.t. \quad &\mathbf{X} \in \{0,1\}^{n\times n}, \mathbf{X}\mathbf{1} = \mathbf{1}, \mathbf{X}^\top \mathbf{1} = \mathbf{1} \end{aligned} \tag{2}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the linear affinity matrix and $\mathbf{X}$ is the discrete assignment matrix. A relaxed projection to the doubly-stochastic matrix is achieved by Sinkhorn algorithm with entropic regularization [76], [77], [78]:

$$\begin{aligned} &\max_{\mathbf{S}} \mathrm{tr}(\mathbf{S}^\top \mathbf{M}) - \tau h(\mathbf{S}) \\ s.t. \quad &\mathbf{S} \in [0,1]^{n\times n}, \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S}^\top \mathbf{1} = \mathbf{1} \end{aligned} \tag{3}$$

where $\mathbf{S}$ is the doubly-stochastic matrix, $h(\mathbf{S}) = \sum_{i,j} \mathbf{S}_{ij} \log \mathbf{S}_{ij}$ is the entropic regularizer and $\tau \in (0, +\infty)$ is the regularization factor. Given any real-valued matrix $\mathbf{M}$, Eq. 3 can be solved by firstly normalizing the regularization factor τ: $\mathbf{S} = \exp(\mathbf{M}/\tau)$. Then $\mathbf{S}$ is row- and column-wise normalized alternatively:

$$\begin{aligned} \mathbf{D}_r = \mathrm{diag}(\mathbf{S}\mathbf{1}), \quad \mathbf{S} = \mathbf{D}_r^{-1}\mathbf{S} \\ \mathbf{D}_c = \mathrm{diag}(\mathbf{S}^\top\mathbf{1}), \quad \mathbf{S} = \mathbf{S}\mathbf{D}_c^{-1} \end{aligned} \tag{4}$$

where $\mathrm{diag}(\cdot)$ means building a diagonal matrix from the input vector. Since Sinkhorn algorithm only involves matrix multiply and element-wise inverse, it is differentiable with automatic gradient techniques provided by deep learning frameworks e.g. PyTorch.

The gap between Sinkhorn and Hungarian algorithm is controlled by τ, viewed as an annealing parameter: Sinkhorn algorithm performs closely to Hungarian if τ is small (at the cost of slowed convergence), and the output of Sinkhorn will become smoother given larger τ [77].

In our unsupervised paradigm, the output of Sinkhorn algorithm ($\mathbf{W}_{ij}$) is treated as the node-wise matching result to compute the unsupervised loss, and also the node-wise similarity matrix which can be used by GM solvers.

3.1.3 Encoding Structural Information

As shown in Fig. 3, the GM solver accepts both CNN features and structural information in images, and the structural information is encoded as edges of graphs. We follow [54] for the construction of edges. For the weighted adjacency matrix of $\mathcal{G}_i$, we firstly compute the Euclidean distance between every pair of keypoints: $l_{ab} = \|p_a - p_b\|$ where p_a, p_b are the coordinates of keypoints. $\mathbf{A}_i \in \mathbb{R}^{n_i \times n_i}$ represents the connectivity matrix of $\mathcal{G}_i$ and the corresponding $\mathbf{A}_i[a,b]$ is computed as:

$$\mathbf{A}_i[a,b] = \exp\left(-\frac{l_{ab}^2}{\sigma \hat{l}^2}\right) \tag{5}$$

where $\hat{l}$ is the median value of all l_{ab}. The diagonal part of $\mathbf{A}_i$ is set as zeros. σ is the scaling factor.

3.1.4 Discrepancy Minimization with GM Solver

Many unsupervised image classifiers share the design of discrepancy-minimization of two different predictions from the same image [30], [31]. To further extend such a paradigm to graph matching, based on node-wise CNN features and additional structural information (and multi-graph consistency information if available), a GM solver offers a matching prediction which is notably different from the node-wise matching result predicted by Sinkhorn.

We develop such a paradigm because the matching quality of Sinkhorn branch is purely based on CNN features, which is probably the weakest matching method available, and the matching prediction by the GM solver should be more accurate than Sinkhorn because it utilizes more information. The unsupervised learning objective is to minimize the discrepancy of these two branches, i.e. the Sinkhorn branch should be as accurate as the GM solver after learning. Since the Sinkhorn branch is purely based on CNN features, a more accurate Sinkhorn matching means improved CNN features, which is also beneficial for the GM solver that shares the same CNN. We empirically find such an unsupervised learning paradigm effective by minimizing the discrepancy between a weaker first-order matching method (Sinkhorn) and a stronger second-order matching method (a GM solver). Finally, gradient is back-propagated through the differentiable Sinkhorn branch, and the gradient through the GM solver is naturally stopped because most GM solvers are non-differentiable.

It is worth noting that we do not restrict the embodiment of the GM solver, and in this paper, we mainly resort to graduated assignment (GA) because this classical approach can fit into all graph matching settings including GM, MGM, and MGM[3]. We are refrained from adopting learning-based GM solvers e.g. [36] because the solver's learnable parameters cannot be updated due to the necessity of stop-gradient of the solver branch (see discussions below).

3.1.5 Remark on the Two-Branch Structure

The two-branch structure in our pipeline can be regarded as a way of Siamese network (while the two branches are not necessarily the same in structure let alone share the same weights) as widely used in literature dating back to [32]. In our case, the input sample refers to a pair of graphs for matching while the output representation denotes the matching matrix instead of an embedding feature as used in many image classification literature. In particular, our approach neither uses negative pairs [30] nor a momentum encoder [31] to prevent collapse into a constant prediction.

As shown in Fig. 3, in our pipeline, only the Sinkhorn branch allows gradient backpropagation while the solver branch is non-differentiable. Technically speaking, one can only estimate the gradient (with additional overhead) in this branch e.g. by borrowing the gradient approximation techniques in BBGM [35]. We find that our design can achieve robust unsupervised learning without encountering the solution collapse. We think the reason is that the learning can be more stable by fixing a branch, see Fig. 4 for a case study on the Willow ObjectClass dataset if we differentiate through the solver branch by [35]. Interestingly we also find that a loosely related work [39] after our conference version [33] also takes a similar gradient-stop strategy for image classification with discrepancy-minimization-based unsupervised learning, and their empirical studies also show the effectiveness of this idea. Besides, the stop-gradient seems necessary for either real gradients [39], or approximate gradients [35].

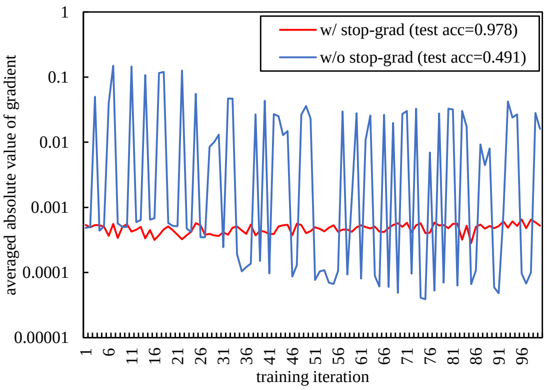


Fig. 4. The necessity of stop-gradient in our unsupervised learning of graph matching, interestingly in line with the conclusions in [39].

3.2 Two-Graph Matching by Discrepancy Minimization

Two-graph matching (GM) is a classic graph matching setting where the algorithm needs to solve the node-to-node correspondence between two graphs $\mathcal{G}_i, \mathcal{G}_j$, and each graph corresponds to an instance in image. The node-to-node correspondence is denoted by the assignment matrix $\mathbf{X}_{ij} \in \{0,1\}^{n_i \times n_j}, s.t. \; \mathbf{X}_{ij}\mathbf{1} \leq \mathbf{1}, \mathbf{X}_{ij}^\top\mathbf{1} \leq \mathbf{1}$. "$\leq$" is element-wise comparison and we allow outliers in both $\mathcal{G}_i$ and $\mathcal{G}_j$ (i.e. partial matching) as the most general case. $\mathbf{1}$ denotes a column vector whose elements are all ones.

Our method works with the popular formulation of pairwise GM, namely Koopmans-Beckmann's Quadratic Assignment Problem [79] (abbreviated as KB-QAP) for $\mathcal{G}_i, \mathcal{G}_j$:

Definition 1. Two-Graph Koopmans-Beckmann's QAP. Solving the matching between $\mathcal{G}_i, \mathcal{G}_j$ requires solving

$$\begin{aligned} &\max_{\mathbf{X}_{ij}} \; \lambda \operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j) + \operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{W}_{ij}) \\ &s.t. \quad \mathbf{X}_{ij} \in \{0,1\}^{n_i \times n_j}, \mathbf{X}_{ij}\mathbf{1} \leq \mathbf{1}, \mathbf{X}_{ij}^\top\mathbf{1} \leq \mathbf{1} \end{aligned} \tag{6}$$

where λ is the weight for the edge-to-edge similarity term, $\mathbf{A}_i, \mathbf{A}_j$ are weighted adjacency matrices of $\mathcal{G}_i, \mathcal{G}_j$, and $\mathbf{W}_{ij}$ represents the node-wise similarity between $\mathcal{G}_i, \mathcal{G}_j$ computed by Sinkhorn from CNN node features.

We minimize the discrepancy of the GM solver $\mathbf{X}_{ij}$ and node-wise matching $\mathbf{W}_{ij}$, measured by cross-entropy:

$$\mathcal{L} = \operatorname{BCE}(\mathbf{X}_{ij}, \mathbf{W}_{ij}) \tag{7}$$

where BCE denotes binary cross-entropy loss (also known as permutation loss in [23], [34]), and $\mathbf{W}_{ij} = \operatorname{Sinkhorn}(\mathbf{F}_i\mathbf{F}_j^\top, \tau_w)$ is node-wise matching composed from individual node features. The gradient is passed through the differentiable Sinkhorn layer for unsupervised learning.

3.3 Multi-Graph Matching by Discrepancy Minimization

For multi-graph matching, we consider all the graphs $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m$ belong to the same category. MGM aims to mitigate wrong matchings obtained by GM solvers, by leveraging the namely cycle-consistency property when multiple graphs are jointly considered.

Definition 2. Cycle-consistency [9]. Under the scenario of partial matching, the matching among $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m$ is cycle-consistent, if and only if

$$\mathbf{X}_{ij} \geq \mathbf{X}_{ik}\mathbf{X}_{kj}, \quad \forall i, j, k \in [m] \tag{8}$$

where $\geq$ is element-wise comparison and $[m]$ denotes the set of graph indices from 1 to m.

One way of enforcing the above cycle-consistency is to decompose pairwise matching by matching to the universe. Universe means the full-set of inlier nodes that co-exist in a multi-matching problem [10]. For example, the MGM problems illustrated in Fig. 2 all have a universe size of 4. The matching between $\mathcal{G}_i$ and the universe of size d is denoted by $\mathbf{U}_i \in \{0,1\}^{n_i \times d}$. Cycle-consistency defined in Eq. 8 is satisfied if all pairwise matchings follow $\mathbf{X}_{ij} = \mathbf{U}_i\mathbf{U}_j^\top$.

Definition 3. Multi-Graph Koopmans-Beckmann's QAP. Multi-graph matching is formulated with KB-QAP, by summing KB-QAP objectives among all pairs of graphs:

$$\max_{\mathbf{X}_{i,j}, i,j \in [m]} \sum_{i,j \in [m]} \left(\lambda \operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j) + \operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{W}_{ij}) \right) \tag{9}$$

where all $\mathbf{X}_{ij}$ are encoded by $\mathbf{X}_{ij} = \mathbf{U}_i\mathbf{U}_j^\top$ and $i, j \in [m]$ means iterating among all combinations of i, j. The constraints in Eq. 9 are omitted for compact illustration.

For the MGM problem, the GM solver predicts a cycle-consistent matching relation. We minimize the cross-entropy between MGM and node-wise matching:

$$\mathcal{L} = \sum_{i,j \in [m]} \operatorname{BCE}(\mathbf{X}_{ij}, \mathbf{W}_{ij}) \tag{10}$$

where $\mathbf{X}_{ij}$ is the cycle-consistent MGM result from GM solver and $\mathbf{W}_{ij} = \operatorname{Sinkhorn}(\mathbf{F}_i\mathbf{F}_j^\top, \tau_w)$ is pairwise matching composed from individual node features.

3.4 Multi-Graph Matching from Mixture of Modes by Discrepancy Minimization

We further consider the more general MGM^3 setting. From now on we further use the term "class" which in this paper is defined as the union of graphs with the same mode, e.g. $\mathcal{C}_1 = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$, and for MGM^3 the set of all graphs is a mixture of graphs from k classes $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$ and $|\mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots \mathcal{C}_k| = m$, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for all $i, j \in [k]$. For MGM^3 problem, the method needs to divide all graphs into k classes, and for all intra-class graph pairs, the node-to-node correspondence is computed for the matching task.

The objective of MGM^3 problem is formulated by slight modification from MGM KB-QAP in Eq. 9 as follows.

Definition 4. Koopmans-Beckmann's QAP for Multi-Graph Matching with a Mixture of Modes. The objective of multi-graph matching with a mixture of modes is modulated by a clustering variable $\mathbb{C}$, where $\mathbb{C}_{ij} = 1$ if $\mathcal{G}_i, \mathcal{G}_j$ are in the same class else $\mathbb{C}_{ij} = 0$.

Algorithm 1: Graduated Assignment for Two-Graph Matching (GA-GM)

```
Input: Visual graphs $\mathcal{G}_i, \mathcal{G}_j$; node-wise similarity $\mathbf{W}_{ij}$;
       initial annealing $\tau_0$; descent factor $\gamma$; minimum
       $\tau_{min}$; universe size $d$; outlier threshold $\phi$.
Randomly initialize $\mathbf{X}_{ij}$; projector $\leftarrow$ Sinkhorn; $\tau \leftarrow \tau_0$;
if enable partial-robust then
    $\mathbf{A}_i = \mathbf{A}_i \times \frac{d}{n_i}$; # handle partial matching
while True do
    while $\mathbf{X}_{ij}$ not converged AND #iter $\leq$ #GMIter do
        $\mathbf{V}_{ij} \leftarrow \lambda \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j + \mathbf{W}_{ij}$; # update $\mathbf{V}_{ij}$
        $\mathbf{X}_{ij} \leftarrow$ projector$(\mathbf{V}_{ij}, \tau)$;
         # project $\mathbf{V}_{ij}$ to (relaxed) feasible space of $\mathbf{X}_{ij}$
        if projector == Hungarian AND $\phi > 0$ then
            # handle outlier matching
            $\mathbf{Q}_{ij} \leftarrow \lambda\ \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j + \mathbf{W}_{ij}$;
            $\mathbf{X}_{ij} \leftarrow (\mathbf{Q}_{ij} < \phi) \odot \mathbf{X}_{ij}$;
    # graduated assignment control
    if projector == Sinkhorn AND $\tau \geq \tau_{min}$ then
        $\tau \leftarrow \tau \times \gamma$;
    else if projector == Sinkhorn AND $\tau < \tau_{min}$ then
        projector $\leftarrow$ Hungarian;
    else
        break;
Output: Matching matrix $\mathbf{X}_{ij}$.
```

$$\max_{\mathbf{X}_{ij}, i,j \in [m]} \sum_{i,j \in [m]} \mathbb{C}_{ij} \left(\lambda \operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j) + \operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{W}_{ij}) \right) \tag{11}$$

with the clustering variable $\mathbb{C}$, only the intra-class graph pairs are counted when summing the objective.

We denote variables related to the mixture of modes by blackboard bold letters. For MGM^3, the GM solver jointly predicts a cycle-consistent matching $\mathbf{X}_{ij}$ and a clustering matrix $\mathbb{C}$. The loss involves a clustering indicator $I(i,j)$:

$$\mathcal{L} = \sum_{i,j \in [m]} I(i,j) \operatorname{BCE}(\mathbf{X}_{ij}, \mathbf{W}_{ij}) \tag{12}$$

where $I(i,j) = \mathbb{C}_{ij}$ under the unsupervised learning setting. There may exist misclassified graphs in the predicted $\mathbb{C}$, so that during training, Eq. 12 will probably involve meaningless matching between graphs from different ground truth classes, yielding challenges for learning. However, experiments show that our unsupervised scheme overcomes this issue by improving matching and clustering simultaneously, outperforming peer learning-free methods.

4 The Graduated Assignment Approach under Discrepancy Minimization Framework

The above unsupervised paradigm has well covered the classic two-graph matching (GM) [5], [6], [7], [15], [16], multi-graph matching (MGM) [8], [9], [10], [11], [80], and the emerging multi-graph matching with mixture of modes (MGM^3) [12]. Now we present detailed algorithmic implementations for these three cases under a unified graduated assignment algorithmic framework, whereby partial matching and outlier-aware matching are also addressed.

There are efforts to develop dedicated solvers for all three graph matching variants including GM, MGM, and MGM^3. However, a unified approach for all three graph matching problems is still missing. In this paper, we resort to graduated assignment (GA) [16] which is a classic algorithm solving hard combinatorial tasks e.g. graph matching. Previous works show the feasibility of graduated assignment in GM [16], MGM [80], and also MGM^3 in our preliminary conference version [33]. In this section, classic graduated assignment algorithms are fit to modern deep learning models, and a unified graduated assignment approach covering all graph matching settings is presented.

In this paper, we name the learning-free version of our graduated assignment approach as GA-GM, GA-MGM, and GA-MGM^3 for GM, MGM, and MGM^3 settings respectively. For those learned with unsupervised learning, they are named graduated assignment neural networks (GANN).

4.1 Unsupervised Learning by Graduated Assignment for Two-Graph Matching (GA-GM)

The application of graduated assignment to graph matching can date back to 1996 when Gold and Rangarajan [16] proposes to solve graph matching by iterative projection with an annealing factor based on the Taylor expansion of graph matching objective function. An illustration of GA-GM algorithm is shown in Alg. 1. The classic graduated assignment method to GM with modern deep learning models is based on the KB-QAP formulation in Eq. 6.

Initialization. Each element in $\mathbf{X}_{ij}$ is initialized by $1/d + 10^{-3}z$, where $z \sim N(0,1)$. A comparison of different initialization techniques can be found in Sec. 5.5.2.

Graduated assignment (GA). The GA algorithm iteratively projects the KB-QAP objective in Eq. 6 to the feasible space, by graduated Sinkhorn and Hungarian assignment. For non-square matrices, minus infinite are padded to form square matrices as the common technique. Recall the discussion in Sec. 3.1.2 that Sinkhorn becomes closer to Hungarian with shrinking τ. Sinkhorn can be adopted to gradually project the input matrix to the feasible assignment matrix, and annealing to the discrete matching result by Hungarian method. In Alg. 1, the annealing speed is controlled by $\gamma < 1$ with $\tau \leftarrow \tau \times \gamma$ and there is a lower limit τ_{min}.

4.2 Unsupervised Learning by Graduated Assignment for Multi-Graph Matching (GA-MGM)

Graduated assignment algorithm on MGM is previously discussed by Solé and Serratosa [80], where a "prototype" graph is required as the anchor (in their paper they use the first graph as the "prototype"). This means they assume the bijection between each pair of graphs (i.e. full-matching) which is hard to satisfy in practice. In contrast, our proposed method is fully decentralized and free from such a constraint, and it can therefore handle the setting of partial matching, and outlier-aware matching. Besides, our method is capable of extending to matching with a mixture of modes. These two settings are more practical.

Algorithm 2: Graduated Assignment for Multi-Graph Matching (GA-MGM)

```
Input: Visual graphs $\{\mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_m\}$; node-wise
       similarity $\{\mathbf{W}_{ij}\}$; initial annealing $\tau_0$; descent
       factor $\gamma$; minimum $\tau_{min}$; universe size $d$; outlier
       threshold $\phi$; clustering weight $\mathbb{B}$ (all $\mathbb{B}_{ij} = 1$ if
       clustering is not considered).
Randomly initialize joint matching $\{\mathbf{U}_i\}$;
 projector $\leftarrow$ Sinkhorn; $\tau \leftarrow \tau_0$;
if enable partial-robust then
    $\mathbf{A}_i = \mathbf{A}_i \times \frac{d}{n_i}$; # handle partial matching
while True do
    while $\{\mathbf{U}_i\}$ not converged AND #iter $\leq$ #GMIter do
        $\forall i \in [m],\ \mathbf{V}_i \leftarrow \mathbf{0}$;
        for $\mathcal{G}_i, \mathcal{G}_j$ in $\{\mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_m\}$ do
            $\mathbf{V}_i \leftarrow \mathbf{V}_i + (\lambda \mathbf{A}_i \mathbf{U}_i {\mathbf{U}_j}^\top \mathbf{A}_j \mathbf{U}_j + \mathbf{W}_{ij} \mathbf{U}_j) \times \mathbb{B}_{ij}$;
             # update $\mathbf{V}_i$
        for $\mathcal{G}_i$ in $\{\mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_m\}$ do
            $\mathbf{U}_i \leftarrow$ projector$(\mathbf{V}_i, \tau)$;
             # project $\mathbf{V}_i$ to (relaxed) feasible space of $\mathbf{U}_i$
        if projector == Hungarian AND $\phi > 0$ then
            # handle outlier matching
            for $\mathcal{G}_i$ in $\{\mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_m\}$ do
                $\mathbf{Q}_i \leftarrow \sum_{j \neq i} \lambda\, \mathbf{A}_i \mathbf{U}_i \mathbf{U}_j^\top \mathbf{A}_j \mathbf{U}_j + \mathbf{W}_{ij} \mathbf{U}_j$;
                $\mathbf{U}_i \leftarrow (\mathbf{Q}_i < \phi) \odot \mathbf{U}_i$;
    # graduated assignment control
    if projector == Sinkhorn AND $\tau \geq \tau_{min}$ then
        $\tau \leftarrow \tau \times \gamma$;
    else if projector == Sinkhorn AND $\tau < \tau_{min}$ then
        projector $\leftarrow$ Hungarian;
    else
        break;
Output: Joint matching matrices $\{\mathbf{U}_i\}$.
```

Our graduated assignment multi-graph matching (GA-MGM) can be viewed as the multi-graph generalization from our GA-GM in Sec. 4.1, and the matching matrix $\mathbf{X}_{ij}$ is replaced by matching-to-universe $\mathbf{U}_i, \mathbf{U}_j$ enforcing cycle-consistency. The proposed method is summarized in Alg. 2. The clustering weight matrix $\mathbb{B}$ has no effect on the MGM problem and is filled with all ones.

Initialization. Each element in $\{\mathbf{U}_i\}$ is initialized by $1/d + 10^{-3}z$ with $z \sim N(0,1)$. In comparison, some MGM peer methods require initialization from pairwise matching [10], [81], [82] or multi-matching [54], leading to additional overhead when computing these matchings.

4.3 Unupervised Graduated Assignment for Multi-Graph Matching with Mixture of Modes (GA-MGM^3)

Our proposed graduated assignment multi-graph matching with a mixture of modes (GA-MGM^3) method is based on the GA-MGM pipeline. It solves the mode-splitting (i.e. clustering) problem and matching problem simultaneously

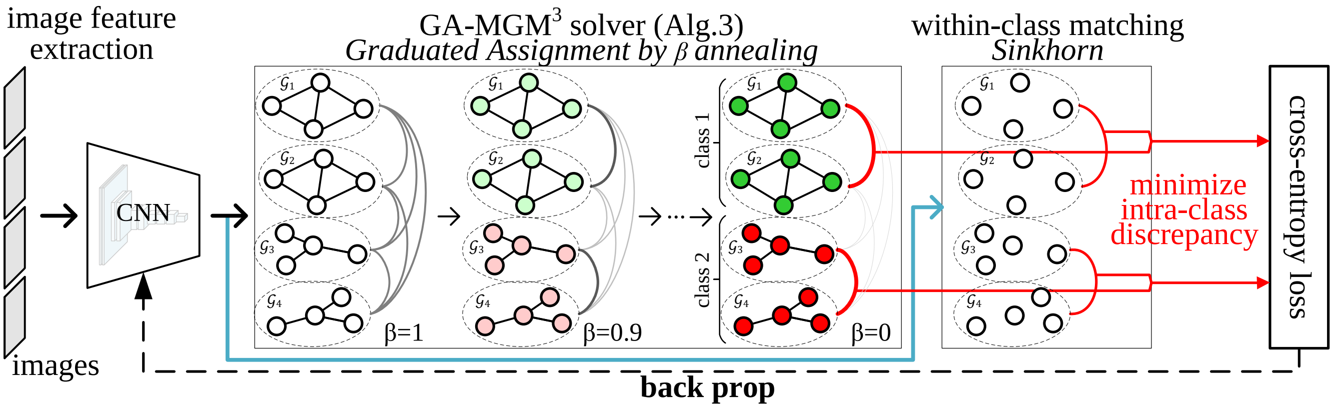


Fig. 5. In the context of visual image matching, it shows a working example of the proposed unsupervised MGM^3 learning pipeline with two classes (each contains two graphs). As Alg. 3 iterates with decreasing annealing parameter β, the graph matching confidence increases (the darker the higher) and so for the clustering confidence as indicated by the brightened node color from white to green/red. As the other parallel branch, CNN and Sinkhorn net can form a node-wise matching network (connected by the blue arrows) whose samples are those within each class determined by Alg. 3. The cross-entropy loss is computed to minimize the discrepancy between the above matching network and Alg. 3, only considering the graphs from the same class. As such, the CNN weights can be trained via back-propagation along the flow in black dashed lines.

Algorithm 3: Graduated Assignment for Multi-Graph Matching over Mixtures of Modes (GA-MGM^3)

```
Input: Visual graphs $\{\mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_m\}$; node-wise
       similarity $\{\mathbf{W}_{ij}\}$; clustering weights $\{\beta\}$.
All $\mathbb{B}_{ij} \leftarrow 1$; # initialize clustering weight matrix
for $\beta$ in $\{\beta\}$ do
    while $\{\mathbf{U}_i\}, \mathbb{C}$ not converged AND #iter $\leq$ #ClsIter do
        $\{\mathbf{U}_i\} \leftarrow$ GA-MGM($\{\mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_m\}, \{\mathbf{W}_{ij}\}, \mathbb{B}$);
         # multi-graph matching
        for $\mathcal{G}_i, \mathcal{G}_j$ in $\{\mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_m\}$ do
            build $\mathbf{X}_{ij}$ from $\mathbf{U}$;
            $\mathbb{A}_{ij} \leftarrow$
              $\lambda_c \exp(-\|\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} - \mathbf{A}_j\|) + \mathrm{tr}(\mathbf{X}_{ij}^\top \mathbf{W}_{ij})$
              # graph-to-graph similarity
        $\mathbb{C} \leftarrow$ spectral clustering with k-means++ on $\mathbb{A}$;
        $\mathbb{B} \leftarrow \mathbb{C} \times (1-\beta) + \beta$; # clustering weight
Output: Joint matching matrices $\{\mathbf{U}_i\}$; clustering
        matrix $\mathbb{C}$.
```

with two annealing parameters tailored for matching and clustering, respectively. We refer to classic clustering algorithms [83] for the clustering step. Motivated by the idea that more precise multi-graph matching will improve the clustering accuracy and vice versa, GA-MGM^3 performs clustering and multi-graph matching alternatively with gradually increased clustering confidence until convergence. GA-MGM^3 is summarized in Alg. 3.

Matching-based clustering. The key challenge of handling graphs with a mixture of modes is finding a reasonable measurement for graph-wise similarity, after which the common spectral clustering technique can be applied. We tackle this problem by proposing a matching-based graph-wise similarity measure for clustering.

Given a batch of graphs from multiple modes, a multi-graph matching relationship can be achieved by out-of-box solvers e.g. our proposed GA-MGM. For graphs in the same category, there should be a higher agreement in their structural and node-wise alignment, compared to graphs from different categories. Therefore, the matching information among graphs can be adopted as the similarity measure. For $\mathcal{G}_i, \mathcal{G}_j$, their similarity is computed from their structural and node-wise agreement:

$$\mathbb{A}_{ij} = \underbrace{\lambda_c \exp(-\|\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} - \mathbf{A}_j\|)}_{\text{structural agreement}} + \underbrace{\mathrm{tr}(\mathbf{X}_{ij}^\top \mathbf{W}_{ij})}_{\text{node-wise agreement}} \tag{13}$$

where the first entry measures the similarity of aligned adjacency matrices and the second entry encodes the agreement on node-wise alignment. A weighting factor λ_c is also considered here. Spectral clustering with k-means++ [83] is further performed on $\mathbb{A}$ as the common technique. Based on Eq. 13, for intra-class graphs $\mathcal{G}_i, \mathcal{G}_j$, a more accurate $\mathbf{X}_{ij}$ will result in larger $\mathbb{A}_{ij}$, therefore increased matching accuracy will lead to more accurate clustering.

Clustering-aware matching. Following the MGM^3 KB-QAP formulation in Eq. 11, the clustering weight $\mathbb{B}_{ij}$ is multiplied to the projection step of GA-MGM (L8 in Alg. 2). If we assign $\mathbb{B} = \mathbb{C}$, the projection step meets the objective function in Eq. 11, assuming 100% clustering accuracy. In realistic conditions with non-optimal clustering, we further apply an annealing parameter β for the clustering weight matrix $\mathbb{B}$: $\mathbb{B}_{ij} = \begin{cases} 1 & \text{if } \mathbb{C}_{ij} = 1 \\ \beta & \text{if } \mathbb{C}_{ij} = 0 \end{cases}$ which is equivalent to $\mathbb{B} = \mathbb{C} \times (1-\beta) + \beta$. The annealing parameter $\beta \in [0,1]$ can be viewed as the confidence of clustering, and the MGM^3 problem is solved by gradually declining β from 1 to 0.

4.4 Solving Partial and Outlier-aware Matching with Affinity Reweighting under Overmatching Perspective

Partial matching and outlier matching are two important scenarios yet less addressed by existing deep graph matching methods, which are yet not explicitly considered in our conference version [33]. We firstly discuss the overmatching issue in Sec. 4.4.1, which is regarded as the major challenge faced for partial and outlier matching, and then present our reweighting techniques to suppress overmatching for partial matching in Sec. 4.4.2 and outlier matching in Sec. 4.4.3.

4.4.1 Unifying Partial Matching and with Outliers from the Overmatching Perspective

The overmatching issue arises given the existence of partial matching (i.e. occluded inliers) and outliers. Since GM algorithms are in general designed to maximize the objective score, and incorrect matching still contributes to the value of the objective score (though the contribution is usually smaller than correct matching), GM algorithms tend to match as many nodes as possible to maximize the objective. We call such a phenomenon as "overmatching" whereby the unexpected partial nodes and outlier nodes are matched, leading to decreased matching accuracy. Though MGM shows some ability of discovering the co-existence of common inliers by enforcing cycle consistency, the overmatching issue still exists because matching additional but incorrect nodes can still increase the objective scores.

4.4.2 Affinity Reweighting for Robust Partial Matching

Partial matching means some inliers are occluded, and the input graphs contain only a partial set of the universe. For the MGM case, the co-existence of nodes can be discovered by leveraging cycle consistency, but the overmatching issue still exists, and the MGM solver may still tend to match as many nodes as possible to maximize Eq. 9.

It is observed that the overmatching issue in partial multi-graph matching occurs when equal weights are given to all graphs with different sizes. For $\mathbf{A}_i$ of size $n_i \times n_i$ and $\mathbf{A}_j$ of size $n_j \times n_j$, the corresponding term in Eq. 9 is

$$J_{ij} = \operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j) \tag{14}$$

where the summed number of elements is proportional to $n_i n_j$ if Eq. 9 is maximized by matching as many pairs in $\mathbf{X}_{ij}$ as possible. These extra nodes may be incorrect matchings. To mitigate this issue, we propose to balance the values of J_{ij} for different i, j by reweighting the edge weights. Since we have J_{ij} proportional to $n_i n_j$, we normalize $\mathbf{A}_i$ by n_i, $\mathbf{A}_j$ by n_j so that the values of J_{ij} are balanced.

Specifically, for $\mathcal{G}_i$ with n_i nodes and the universe size of d, we normalize $\mathbf{A}_i$ as $\mathbf{A}_i = \mathbf{A}_i \times \frac{d}{n_i}$. We multiply the universe size so that the hyper-parameters do not need significant modification. Such a reweighting technique is found empirically effective in our experiments. It is equivalent to reweighting the edge affinities in the matching objective (Eq. 9). Note that such a reweighting technique is also applicable for $\mathbf{W}_{ij}$, but we do not consider it because $\mathbf{W}_{ij}$ is already normalized by the Sinkhorn algorithm.

4.4.3 Proposal Reweighting for Robust Outlier Matching

Outliers are ubiquitous e.g. by the false alarm detection of visual keypoint detectors which cannot find their correspondence from any other views. It also leads to the overmatching issue when two outliers are forced to be matched. Outlier matching is different from partial matching (see Fig. 2), and requires different reweighting techniques. Unfortunately, it has not been specifically studied in previous GM learning works [5], [9], [33], [35], [36], [37].

We resort to image matching pipelines [1], whereby matching proposals are filtered and reweighted by a predefined confidence threshold. Since all elements in the doubly-stochastic matrix output by the Sinkhorn algorithm can be viewed as the confidence of node matching, [1] performs thresholding over the doubly-stochastic matrix and reweights the below-threshold elements to zero. In this paper, we propose a confidence measurement that naturally arises in our graduated assignment algorithm: the contributed objective score (i.e. affinity) of each matching candidate. Starting from the two-graph objective (Eq. 6) as an example, the contribution to the objective score is:

TABLE 1
The averaged inference time of learning-free GA-MGM and GA-MGM^3 w/ or w/o the compact matrix form, on the Willow ObjectClass dataset in our experiment (in line with Table 3 and the right half of Table 5).

	compact matrix	inference time (s)
GA-MGM	✓	**1.1**
	×	345.6
GA-MGM^3	✓	**107.2**
	×	252.0

$$\mathbf{Q}_{ij} = \lambda\ \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j + \mathbf{W}_{ij} \tag{15}$$

where $\mathbf{Q}_{ij} \in \mathbb{R}^{n_i \times n_j}$ measures the confidence of all matching candidates for $\mathcal{G}_i, \mathcal{G}_j$, and $\mathbf{X}_{ij}$ is the matching matrix proposed in the current iteration. The objective score is $\operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{Q}_{ij})$. At each iteration, we reweight matchings whose corresponding score is lower than a given threshold ϕ:

$$\mathbf{X}_{ij} = (\mathbf{Q}_{ij} < \phi) \odot \mathbf{X}_{ij} \tag{16}$$

where $\odot$ means element-wise product.

For the more general multi-graph scenario (Eq. 9), we exploit a matching-to-universe representation: $\mathbf{X}_{ij} = \mathbf{U}_i \mathbf{U}_j^\top$. We also derive the equivalent version for multi-graphs:

$$\mathbf{Q}_i = \sum_{j \neq i} \lambda\ \mathbf{A}_i \mathbf{U}_i \mathbf{U}_j^\top \mathbf{A}_j \mathbf{U}_j + \mathbf{W}_{ij} \mathbf{U}_j \tag{17}$$

where $\mathbf{Q}_i \in \mathbb{R}^{n_i \times d}$, d is the universe size. The thresholding and reweighting at each iteration becomes

$$\mathbf{U}_i = (\mathbf{Q}_i < \phi) \odot \mathbf{U}_i \tag{18}$$

In the implementation, reweighting is only considered if the projector is the Hungarian algorithm. For the Sinkhorn projection steps, the matching relations are still unclear and the confidence scores are less reliable, thus we encourage the algorithm to find as many matchings as possible (i.e. we allow overmatching). With more matchings and higher recalls after the Sinkhorn steps, the outliers can be discarded without significantly poisoning the recall.

4.5 Implementation Details and Further Discussions

We discuss our GPU-friendly implementation of graduated assignment algorithms in Sec. 4.5.1, and we provide the theoretical analysis of graduated assignment in Sec. 4.5.2. The initialization techniques of Sinkhorn and Hungarian projectors are discussed in Sec. 4.5.3.

4.5.1 Compact Matrix Form Implementation

The multi-graph KB-QAP objective in Eq. 9 can be equivalently written in a compact matrix form, whereby the computational efficiency can be easily benefited from GPU

TABLE 2
Hyper-parameter configurations over different datasets to reproduce our reported results in this paper.

parameter	Willow ObjectClass		CUB2011		Pascasl VOC Keypoint		PhotoTourism	description
	GM & MGM	MGM^3	GM & MGM	MGM^3	GM & MGM	MGM^3	GM & MGM	
lr	10^{-3}	10^{-3}	10^{-3}	10^{-4}	10^{-3}	10^{-4}	10^{-6}	learning rate
lr-steps	{200,1000}	{100,500}	{2000}	{1000}	{2500}	{500}	{5000}	lr=lr×0.1 at these steps
λ	1	1	0.1	0.1	0.005	0.005	0.001	edge weight (Eq. 6&9)
λ_c	-	1	-	0.1	-	0.1	-	clustering weight (Eq. 13)
τ_w	0.05	0.05	0.05	0.05	0.05	0.05	-	Sinkhorn's τ for $\mathbf{W}_{ij}$
β	-	$\{1, 0.9, 0\}$	-	$\{1, 0.9\}$	-	$\{1, 0.9\}$	-	clustering annealing
τ_0	0.1	$\{0.1, 0.1, 0.1\}$	0.05	$\{0.05, 0.05\}$	0.05	$\{0.05, 0.05\}$	5×10^{-4}	init τ (match annealing)
τ_{min}	10^{-2}	$\{10^{-2}, 10^{-2}, 10^{-3}\}$	10^{-3}	$\{10^{-2}, 10^{-2}\}$	0.005	$\{0.005, 0.005\}$	10^{-4}	min τ (match annealing)
γ	0.8	0.8	0.8	0.8	0.8	0.8	0.9	τ's decay factor
#SKIter	10	10	100	100	50	50	100	Sinkhorn's loop number
#GMIter	500	500	500	500	500	500	100	Alg. 1&2's loop number
#ClsIter	-	10	-	10	-	10	-	Alg. 3's loop number
σ	1	1	1	1	2	2	0.01	scaling factor (Eq. 5)
partial-robust	×	×	×	×	✓	✓	✓	enable $\mathbf{A}_i = \mathbf{A}_i \times \frac{d}{n_i}$
outlier-robust(ϕ)	-1	-1	-1	-1	-1	-1	1.1	outlier threshold

parallelization concerning multiple graphs. Inspired by [54], the matching objective in Eq. 9 can be written as:

$$\max_{\mathbf{U}} \lambda \operatorname{tr}(\mathbf{U}\mathbf{U}^\top\mathbf{A}\mathbf{U}\mathbf{U}^\top\mathbf{A}) + \operatorname{tr}(\mathbf{U}\mathbf{U}^\top\mathbf{W}) \tag{19}$$

where $\mathbf{U}$ is the joint matching matrix by stacking all $\mathbf{U}_i$ at their first dimension. $\mathbf{A}$ is the joint adjacency matrix by placing $\mathbf{A}_i$ at its diagonal, and $\mathbf{W}$ is the joint node-to-node similarity matrix:

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_0 \\ \vdots \\ \mathbf{U}_m \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_m \end{pmatrix} \tag{20}$$
$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_{00} & \cdots & \mathbf{W}_{0m} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{m0} & \cdots & \mathbf{W}_{mm} \end{pmatrix}$$

The update step of GA-MGM (L8 of Alg. 2) is replaced with

$$\mathbf{V} \leftarrow \lambda\mathbf{A}\mathbf{U}\mathbf{U}^\top\mathbf{A}\mathbf{U} + \mathbf{W}\mathbf{U} \tag{21}$$

The clustering weight can also be fused with this compact matrix form, by modifying Eq. 11 as

$$\max_{\mathbb{M},\mathbf{U}} \lambda \operatorname{tr}(\mathbf{U}\mathbf{U}^\top\mathbf{A}(\mathbf{U}\mathbf{U}^\top \circ \mathbb{M})\mathbf{A}) + \operatorname{tr}(\mathbf{U}\mathbf{U}^\top(\mathbf{W} \circ \mathbb{M})) \tag{22}$$

where $\mathbb{M}_{n_i:n_{(i+1)},n_j:n_{(j+1)}} = \mathbb{C}_{ij}$ introduces the clustering information of MGM^3 objective and $\circ$ denotes element-wise multiplication. Therefore, considering additionally clustering, the update step in GA-MGM^3 can be replaced with

$$\mathbf{V} \leftarrow \lambda\mathbf{A}(\mathbf{U}\mathbf{U}^\top \circ \mathbb{M})\mathbf{A}\mathbf{U} + (\mathbf{W} \circ \mathbb{M})\mathbf{U} \tag{23}$$

where $\mathbb{M}_{n_i:n_{(i+1)},n_j:n_{(j+1)}} = \mathbb{B}_{ij}$ encodes the clustering weight. The handling of outlier matching in Eq. 17 also has its compact matrix form:

$$\mathbf{Q} \leftarrow \lambda\ \mathbf{A}\mathbf{U}\mathbf{U}^\top\mathbf{A}\mathbf{U} + \mathbf{W}\mathbf{U} \tag{24}$$

Table 1 shows a significant acceleration with this compact matrix form, because the benefits of our compact matrix form can be automatically leveraged by the GPU support of PyTorch. Please note that this compact matrix form is tailored for multiple graphs and has no effect on GA-GM.

4.5.2 Theoretical Analysis on Graduated Assignment

In this section, we discuss the theoretical groundings of our graduated assignment algorithms taking GA-MGM as an example. GA-GM can be viewed as a special case of GA-MGM where the number of graphs is set to 2, and the derivation of GA-MGM^3 can be achieved by simply multiplying the clustering term. The multi-graph KB-QAP objective in Eq. 9 is abbreviated as J. By setting $\mathbf{X}_{ij} = \mathbf{U}_i^\top\mathbf{U}_j$, for a set of feasible multi-graph matching solutions $\{\mathbf{U}_i^0\}$, J can be rewritten in its Taylor series:

$$\begin{aligned} J = &\sum_{i,j\in[m]} \lambda\operatorname{tr}(\mathbf{U}_j^0\mathbf{U}_i^{0\top}\mathbf{A}_i\mathbf{U}_i^0\mathbf{U}_j^{0\top}\mathbf{A}_j) + \operatorname{tr}(\mathbf{U}_j^0\mathbf{U}_i^{0\top}\mathbf{W}_{ij}) \\ &+ \sum_{i\in[m]} \operatorname{tr}(\mathbf{V}_i^\top(\mathbf{U}_i - \mathbf{U}_i^0)) + \ldots \end{aligned} \tag{25}$$

where

$$\mathbf{V}_i = \left.\frac{\partial J}{\partial \mathbf{U}_i}\right|_{\mathbf{U}_i=\mathbf{U}_i^0} = \sum_{j\in[m]} \left(2\lambda\mathbf{A}_i\mathbf{U}_i^0\mathbf{U}_j^{0\top}\mathbf{A}_j\mathbf{U}_j^0 + \mathbf{W}_{ij}\mathbf{U}_j^0\right) \tag{26}$$

where all terms are constants except $\mathbf{V}_i$. Approximating J with its first-order Taylor expansion, the maximization of J is equivalent to maximizing $\sum_{i\in[m]} \operatorname{tr}(\mathbf{V}_i^\top\mathbf{U}_i)$, which is equivalent to solving m independent linear assignment problems. Therefore, given initial $\mathbf{U}_i$, our GA-MGM works by considering the first-order Taylor expansion of the multi-graph KB-QAP objective and computing $\mathbf{V}_i$ by Eq. 25. The above linear assignment problems are solved via either Sinkhorn algorithm (controlled by τ) [77] or Hungarian algorithm [75]. The linear assignment solver is controlled by the annealing parameter τ, ensuring $\mathbf{U}_i$ gradually converges to a high-quality discrete solution. Readers are referred to [84] for the convergence analysis of graduated assignment. We omit the constant 2 in our implementation because it can be absorbed by the parameter λ.

4.5.3 Initialization of Projectors

For the GM and MGM problem, our graduated algorithms in Alg. 1 and 2 firstly work with coarse linear assignment solvers, i.e. Sinkhorn method with large τ, then gradually converge to a fine-grained solution with shrinking τ, finally getting the discrete solution via Hungarian algorithm. Initializing the projector in Alg. 2 with the coarse Sinkhorn

TABLE 3
Matching accuracy with both learning-free MGM methods and supervised learning peer methods on Willow ObjectClass dataset (mean and std by 50 trials). Compared results are quoted from the original papers without knowing the std results. Our learning-free GA-GM and GA-MGM surpass previous learning-free methods, and our unsupervised learning GANN-MGM best performs among supervised learning methods.

method	learning	car	duck	face	mbike	wbottle	mean
MatchLift [82]	free	0.665	0.554	0.931	0.296	0.700	0.629
MatchALS [81]	free	0.629	0.525	0.934	0.310	0.669	0.613
MSIM [11]	free	0.750	0.732	0.937	0.653	0.814	0.777
MGM-Floyd [37]	free	**0.850**	0.793	**1.000**	0.843	0.931	0.883
HiPPI [54]	free	0.740	0.880	**1.000**	0.840	0.950	0.882
VGG16+Sinkhorn	free	0.776±0.238	0.732±0.253	0.990±0.044	0.525±0.205	0.824±0.190	0.769
GA-GM (ours)	free	0.814±0.300	0.885±0.197	**1.000±0.000**	0.809±0.243	**0.954±0.113**	0.892
GA-MGM (ours)	free	0.746±0.153	**0.900±0.106**	0.997±0.021	**0.892±0.139**	0.937±0.072	**0.894**
HARG-SSVM [21]	supervised	0.584	0.552	0.912	0.444	0.666	0.632
GMN [22]	supervised	0.743	0.828	0.993	0.714	0.767	0.809
PCA-GM [23]	supervised	0.840	0.935	**1.000**	0.767	0.969	0.902
DGMC [85]	supervised	0.903	0.890	**1.000**	0.921	0.971	0.937
CIE-H [34]	supervised	0.822	0.812	**1.000**	0.900	0.976	0.902
BBGM [35]	supervised	0.968	0.899	**1.000**	0.998	**0.994**	0.972
NMGM [36]	supervised	**0.973**	0.854	**1.000**	0.860	0.977	0.933
GANN (with HiPPI [54])	unsupervised	0.845±0.313	0.868±0.1573	0.992±0.010	0.911±0.172	0.954±0.090	0.914
GANN-GM (ours)	unsupervised	0.854±0.261	0.898±0.226	**1.000±0.000**	0.886±0.186	0.964±0.104	0.920
GANN-MGM (ours)	unsupervised	0.964±0.058	**0.949±0.057**	**1.000±0.000**	**1.000±0.000**	0.978±0.035	**0.978**

method is adopted under GM and MGM settings. For the MGM^3 problem, where GA-MGM^3 (Alg. 3) repeatedly calls GA-MGM (Alg. 2) in its loop, a more cost-efficient initialization strategy is proposed. For each distinct value of β, the projector is initialized by Sinkhorn with a large τ when GA-MGM is called for the first time. For later iterations with the same β, the projector is initialized with the Hungarian algorithm, because only relatively small changes will occur in the clustering result, and the corresponding matching result should not change violently. A projection with the Hungarian algorithm should be adequate. We empirically find such an initialization technique improves both speed and stability of our GA-MGM^3 method.

5 Experiments

In Sec. 5.1 we introduce our evaluation protocol. Results are reported for two separate settings: GM with single mode given two or more graphs (GM/MGM) in Sec. 5.2 and GM with a mixture of modes (MGM^3) in Sec. 5.3. We also test our techniques on the natural image matching pipeline with application to structure-from-motion (SfM) in Sec. 5.4. Further experiments and ablation studies are given in Sec. 5.5.

5.1 Evaluation Protocol

Our implementation is built in line with deep GM peer methods [23], [85], and we use the Matlab code released by [12] for MGM^3 peer methods. All images are resized to 256×256 and normalized before being passed to the VGG16 network. The raw RGB values are firstly divided by 256 (normalized to [0, 1)), and normalized by mean $[0.485, 0.456, 0.406]$ and STD $[0.229, 0.224, 0.225]$ which are collected from ImageNet statistics. We implement Alg. 1, 2 and 3 that supports GPU parallelization. Experiments are conducted on our Linux workstation with Xeon-3175X@3.10GHz, RTX8000, and 128GB memory. The parameter configurations are listed in Table 2.

5.1.1 Evaluation Metrics

For both matching with single mode (GM and MGM) and matching with a mixture of modes (MGM^3), we consider the following matching accuracy metrics. Given one predicted assignment $\mathbf{X}$ and its ground truth $\mathbf{X}^{gt}$, $\mathrm{precision} = \frac{\mathrm{tr}(\mathbf{X}^\top \mathbf{X}^{gt})}{\mathrm{sum}(\mathbf{X})}$, $\mathrm{recall} = \frac{\mathrm{tr}(\mathbf{X}^\top \mathbf{X}^{gt})}{\mathrm{sum}(\mathbf{X}^{gt})}$ and the corresponding f1-score are considered. Mean and STD are reported from all possible pairs of graphs. Note that $\mathrm{precision} = \mathrm{recall} = \mathrm{f1}$ if there exists no partial matching or outliers, and we denote this metric as **matching accuracy** inline with [23], [34], [85].

For the MGM^3 task, matching accuracy is evaluated with intra-class graphs. The following clustering metrics are also considered: **1) Clustering Purity (CP)** [86] where $\mathcal{C}_i$ represent the predicted class i and $\mathcal{C}_j^{gt}$ is ground truth class j: $\mathrm{CP} = \frac{1}{m}\sum_{i=1}^{k} \max_{j\in\{1,\dots,k\}} |\mathcal{C}_i \cap \mathcal{C}_j^{gt}|$; **2) Rand Index (RI)** [87] computed by the number of graphs predicted in the same class with the same label n_{11} and the number of graphs predicted in separate classes and with different labels n_{00}, normalized by the total number of graph pairs n: $\mathrm{RI} = \frac{n_{11}+n_{00}}{n}$; **3) Clustering Accuracy (CA)** [12] where $\mathcal{A}, \mathcal{B}, \dots$ are ground truth classes and $\mathcal{A}', \mathcal{B}', \dots$ are predicted classes and k is the number of classes: $\mathrm{CA} = 1 - \frac{1}{k}\left(\sum_{\mathcal{A}}\sum_{\mathcal{A}'\neq\mathcal{B}'} \frac{|\mathcal{A}'\cap\mathcal{A}||\mathcal{B}'\cap\mathcal{A}|}{|\mathcal{A}||\mathcal{A}|} + \sum_{\mathcal{A}'}\sum_{\mathcal{A}\neq\mathcal{B}} \frac{|\mathcal{A}'\cap\mathcal{A}||\mathcal{A}'\cap\mathcal{B}|}{|\mathcal{A}||\mathcal{B}|}\right)$.

5.1.2 Graph Matching Datasets

We consider the following datasets: Willow ObjectClass, CUB2011, Pascal VOC Keypoint. These datasets are evaluated under the setting of GM with single mode (GM and MGM) and with a mixture of modes (MGM^3).

- *Willow ObjectClass* dataset[3] [21] has received wide attention which contains 304 images collected from Caltech-256 [88] (208 faces, 50 ducks and 66 winebottles) and Pascal VOC 2007 [89] (40 cars and 40 motorbikes). This dataset is relatively small for deep learning models, and it creates an ideal setting whereby each image

[3] http://www.di.ens.fr/willow/research/graphlearning/WILLOW-ObjectClass_dataset.zip

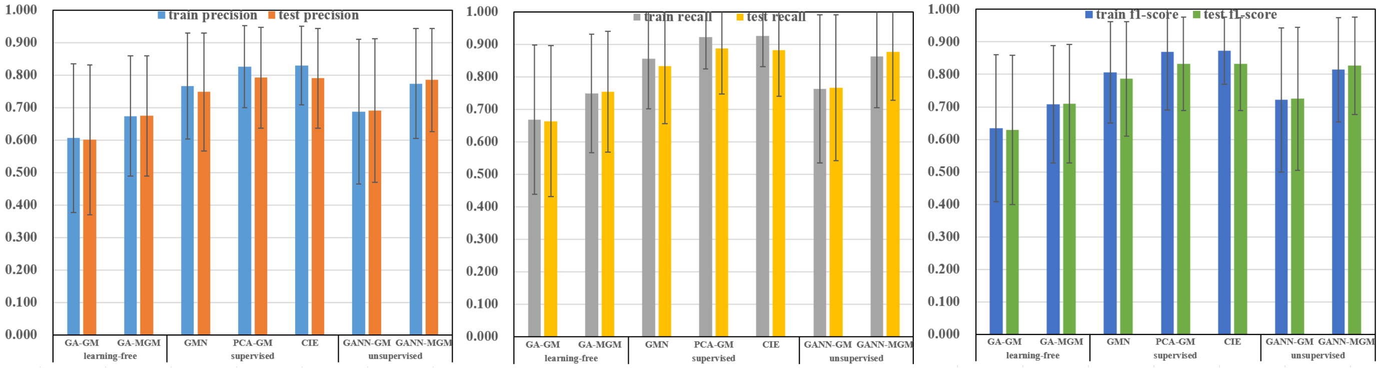


Fig. 6. Mean and STD of precision, recall, and f1-score of graph matching with a single mode on CUB2011 dataset. MGM statics are computed from all graph pairs in each category, and two-graph matching statics are computed from 1000 random graph pairs. Improvement can be seen from our learning-free methods to our unsupervised methods, which are comparative with novel supervised learning methods on the testing set.

contains one object and is labeled with 10 common semantic keypoints without outliers. It is worth noting that there are originally 209 face images in Willow ObjectClass, but the face image No. 0160 is labeled with only 8 keypoints (which is probably a mistake), and we exclude this image during evaluation.

- *CUB2011* dataset[4] [29] is a large-scale image matching dataset including 11,788 bird images from 200 categories with about 50%/50% split of training/testing images for each category. The keypoints may be self-occluded which results in the partial matching challenge (about 12 out of 15 keypoints for each image), and the poses of birds vary from flying, standing, and swimming, and the images may contain different illumination and background situations. All these factors yield more challenges, compared to Willow ObjectClass.
- *Pascal VOC Keypoint* dataset contains natural images from 20 classes in VOC 2011[5] [89] with additional keypoint labels[6] provided by [28]. To adapt the raw dataset to graph matching, specific filtering rules (see [22]) are developed to retain 7,020 training images and 1,682 testing images to obtain a new benchmark, and the size of the graph for each image ranges from 6 to 23. This protocol has been widely followed by subsequent learning-based solvers [23], [34], [35]. So far only two-graph matching learning is considered. The large variations in appearance, scale, illumination, and pose make the Pascal VOC Keypoint so challenging that state-of-the-art supervised algorithms still struggle on this dataset. However, the mainstream of existing evaluations [23], [34] on this dataset limits the problem setting to GM and pre-filter nodes to bypass the challenge of partial matching except for a recent attempt [35].

5.2 Evaluation on Graph Matching with Single Mode

Graph matching with single mode is the classic setting, where all graphs belong to the same category. We evaluate our unsupervised learning GANN-GM and GANN-MGM against learning-free MGM solvers [11], [37], [54], [81], [82] and supervised learning-based GM [21], [22], [23], [34], [35], [85] and MGM [36] methods.

5.2.1 Results on Willow ObjectClass

Our evaluation protocol follows [36], [37], where both learning-free and learning-based methods are compared, as shown in Table 3. Our MGM models are trained in an unsupervised manner with 8 graphs randomly drawn per category, and tested by jointly matching the entire category since this dataset is relatively small. Most compared learning graph matching methods are two-graph matching since there is little effort in learning MGM except NMGM [36]. Among learning-free methods, our GA-MGM performs comparatively with state-of-the-art MGM solvers. Most importantly, under the learning setting, our unsupervised learning GANN-MGM surpasses all supervised two-graph matching learning peer methods and best performs in terms of mean accuracy. Since we do not restrict the embodiment of the MGM solver in Fig. 3, we also implement and compare another unsupervised learning method by replacing GA-MGM with HiPPI [54], whose matching accuracy also surpasses several learning baselines.

5.2.2 Results on CUB2011

CUB2011 is more challenging compared to Willow ObjectClass because the graphs are larger and we need to deal with partial matching. Since the learning-free GA-MGM is comparative with the recent learning-free peer MGM methods [11], [37], [54] on Willow ObjectClass, we mainly compare with the available supervised deep learning methods (only for two graphs, since multi-graph learning NMGM [36] does not support partial matching). For our MGM method, during training, we randomly draw 8 graphs per category for cost-efficiency, and during evaluation, all graphs from the same category are matched jointly. We follow the train/test original split of the dataset, and our unsupervised models are learned on the training set. As shown in Fig. 6, for the MGM task, our GANN-MGM performs comparatively on the testing set against state-of-the-art PCA-GM [23] and CIE [34] which are trained by ground truth supervision. Also, our unsupervised learning scheme generalizes soundly from training samples to unseen testing

[4] http://www.vision.caltech.edu.s3-us-west-2.amazonaws.com/visipedia-data/CUB-200-2011/CUB_200_2011.tgz

[5] http://host.robots.ox.ac.uk/pascal/VOC/voc2011/index.html

[6] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/shape/poselets/voc2011_keypoints_Feb2012.tgz

TABLE 4
F1 scores (%) on Pascal VOC Keypoint dataset (without filtering). The bold statistics denote the best-performing method. The Pascal VOC Keypoint dataset seems to be challenging for unsupervised models without ground truth labels, and we develop a new graph matching learning paradigm for state-of-the-art models NGMv2 and BBGM in combination of unsupervised pretraining (by GANN-MGM) and supervised learning.

model	learning	backbone	#params	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbkie	person	plant	sheep	sofa	train	tv	mean
GMN [22]	sup.	VGG16	12.9M	28.0	55.0	33.1	27.0	79.1	52.2	26.0	40.2	28.4	36.0	29.8	33.7	39.4	43.0	22.1	71.8	30.8	25.9	58.8	78.0	41.9
PCA-GM [23]	sup.	VGG16	41.7M	27.5	56.5	36.6	27.7	77.8	49.2	23.9	42.3	27.4	38.2	38.7	36.5	39.3	42.8	25.6	74.3	32.6	24.7	51.5	74.3	42.4
GANN-GM (ours)	unsup.	VGG16	12.4M	12.6	19.5	16.6	18.5	41.1	32.4	19.3	12.3	24.3	17.2	38.0	12.2	15.9	18.2	19.4	35.5	14.8	15.4	41.5	60.8	24.3
GANN-MGM (ours)	unsup.	VGG16	12.4M	18.1	33.4	20.2	28.2	71.7	33.9	22.0	24.7	23.5	22.4	50.2	19.6	20.9	27.7	19.5	74.5	19.3	26.5	39.8	72.7	33.4
		ResNet34	21.3M	16.4	37.4	23.2	29.5	80.8	33.6	24.1	17.8	26.9	22.6	53.4	16.9	21.8	30.1	23.3	86.0	19.7	31.7	39.0	71.5	35.3
NGMv2 [36]	sup.	VGG16	71.4M	45.9	66.6	**57.2**	**47.3**	87.4	64.8	50.5	59.9	39.7	60.9	42.1	58.3	58.5	61.9	44.6	94.5	50.1	35.2	73.1	82.1	59.0
		ResNet34	77.9M	45.1	65.5	52.7	44.0	87.3	69.4	56.1	62.2	45.7	63.6	61.9	59.6	59.2	67.8	54.4	96.9	57.0	45.9	74.3	83.6	62.6
BBGM [35]	sup.	VGG16	71.4M	41.3	65.7	55.0	43.4	86.8	61.1	35.5	59.0	40.2	60.0	29.7	57.1	57.5	65.9	37.7	95.8	52.6	30.3	74.4	84.1	56.7
		ResNet34	77.9M	38.1	69.1	54.2	45.0	87.0	**74.7**	43.3	**62.3**	**48.3**	63.7	63.8	60.9	60.4	65.4	50.2	**97.1**	56.2	45.9	78.4	82.2	62.3
GANN-MGM (ours)	unsup.	VGG16	71.4M	**47.0**	**69.4**	53.7	46.3	85.7	67.6	**59.0**	60.2	45.9	61.0	29.9	57.9	59.5	63.2	47.4	92.2	51.5	39.9	71.6	78.3	59.4 (+0.4)
+NGMv2 [36]	+sup.	ResNet34	77.9M	46.5	66.2	56.5	46.5	85.9	73.8	57.4	61.4	47.3	65.7	**63.9**	59.4	60.1	**70.6**	**54.7**	94.3	57.0	51.8	74.9	82.4	**63.8 (+1.2)**
GANN-MGM (ours)	unsup.	VGG16	71.4M	43.0	69.2	55.3	46.3	85.4	66.7	53.3	61.4	46.7	64.0	33.7	61.7	60.7	64.0	43.9	94.0	54.9	**52.3**	78.0	80.0	60.7 (+4.0)
+BBGM [35]	+sup.	ResNet34	77.9M	39.6	68.9	56.2	46.7	**87.4**	71.9	44.5	62.0	47.9	**65.8**	50.8	**62.6**	**61.6**	62.9	49.4	96.9	**59.6**	49.7	**80.6**	**85.3**	62.5 (+0.2)

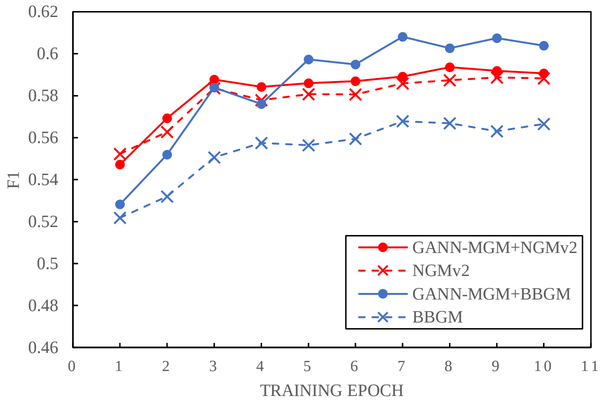
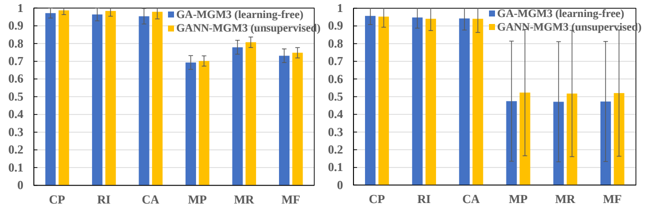


Fig. 7. Experiment result on Pascal VOC Keypoint dataset for initializing VGG16 weights of NGMv2 [36] and BBGM [35] by the proposed unsupervised learning model GANN-MGM. Without modifying any model architectures, the unsupervised pretraining by GANN-MGM leads to faster convergence and better performance compared to the versions initialized by ImageNet classification weights.



(a) CUB2011

(b) Pascal VOC Keypoint

Fig. 8. Mean and STD results of our learning-free GA-MGM^3 and unsupervised learning GANN-MGM^3 on the MGM^3 problem. Since the performance of our learning-free GA-MGM^3 is comparative to other learning-free baselines [9], [12] on Willow ObjectClass, we compare unsupervised learning-based GANN-MGM^3 with GA-MGM^3. For the CUB2011 dataset, both matching and clustering performances can be elevated by unsupervised learning. For the Pascal VOC Keypoint dataset, the clustering performance is nearly saturated, but the matching accuracy can be improved by unsupervised learning.

samples, while the supervised learning methods suffer from overfitting on training data. Besides, the reason why our unsupervised GANN-MGM performs better on testing data than training data is probably that the testing set is slightly easier than the training set, as our learning-free version GA-MGM also performs better on testing data.

5.2.3 Results on Pascal VOC Keypoint

In line with our previous experiments, we adopt the original dataset labels without filtering the keypoints i.e. partial matching setting, which is also known as the "without filtering" setting in [35]. We jointly consider 5 graphs for multi-matching. We reimplement learning-based peer methods [22], [23], [35], [36] to this challenging setting, and results are reported in Table 4. We discover that training labels are important for the challenging Pascal VOC Keypoint dataset, and our unsupervised learning methods are inferior to supervised learning models [22], [23]. Existing deep graph matching models are all based on the VGG16 [25] CNN for the purpose of fair comparison. In this paper, we also consider the ResNet34 [90] CNN backbone. Improved matching accuracy is achieved with the higher model capacity offered by ResNet. We further develop a new graph matching learning paradigm by firstly pretraining the CNN weights (by our devised unsupervised learning approach), and then learning by supervised labels. In experiments, we consider state-of-the-art graph matching models [35], [36]. For GANN-MGM+NGMv2 and GANN-MGM+BBGM, the VGG16/ResNet34 backbones are initialized by our unsupervised learning model, and the other layers (SplineConv [91] layers and NGM's solver layers) are randomly initialized. In contrast, the original training pipeline adopts VGG16/ResNet34 weights pretrained on ImageNet [73] classification task, which seem to suffer from certain domain gaps to the graph matching task. In Fig. 7, we plot the test accuracy for the training process (VGG16 backbone), and the models initialized by GANN-MGM converge faster and better compared to the baselines.

5.3 Evaluation under the Setting of Mixture of Modes

We then test the case with graphs belonging to a mixture of underlying modes, and the solver needs to handle matching and clustering simultaneously. Since there is no learning-based solution to MGM^3 so far to our best knowledge, the learning-free method DPMC [12] is compared whose evaluation protocol is also adopted here.

5.3.1 Results on Willow ObjectClass

In line with [12], matching and clustering are simultaneously solved for graphs from 3 Willow categories (car, duck, motorbike). Apart from the dedicated MGM^3 solver [12],

TABLE 5

Multiple graph matching with mixture of modes (MGM^3) evaluation (with inference time in second) on Willow ObjectClass dataset. Our learning-free version GA-MGM^3 is slightly inferior to DPMC [12], but both matching and clustering accuracies can be elevated by unsupervised learning, and our unsupervised GANN-MGM^3 surpasses all peer methods.

method	learning	8 Cars, 8 Ducks, 8 Motorbikes					40 Cars, 50 Ducks, 40 Motorbikes				
		CP	RI	CA	MA	time (s)	CP	RI	CA	MA	time (s)
RRWM [5]	free	0.879	0.871	0.815	0.748	**0.4**	0.962	0.949	0.926	0.751	**8.8**
CAO-C [9]	free	0.908	0.903	0.860	0.878	3.3	0.971	0.960	**0.956**	0.906	1051.5
CAO-PC [9]	free	0.887	0.883	0.831	0.870	1.8	0.971	0.960	**0.956**	0.886	184.0
DPMC [12]	free	0.931	0.923	0.890	0.872	1.2	0.969	0.959	0.948	**0.941**	97.5
GA-MGM^3 (ours)	free	0.921	0.905	0.893	0.653	10.6	0.890	0.871	0.850	0.669	107.8
GANN-MGM^3 (ours)	unsup.	**0.976**	**0.970**	**0.963**	**0.896**	5.2	**0.974**	**0.968**	**0.956**	0.906	80.7

TABLE 6

Stereo reconstruction (SfM) accuracy and matching accuracy (%) on the validation set of PhotoTourism dataset. "AUC@X" means Area Under Curve if the pose error is smaller than X degrees. "Prec" means matching precision i.e. $\frac{\#\text{correct match}}{\#\text{predicted match}}$, and "MScore" means $\frac{\#\text{correct match}}{\#\text{keypoints}}$.

	label-free finetune	AUC@5	AUC@10	AUC@20	Prec	MScore
SuperGlue [1]	no	16.57	32.40	48.86	65.61	24.50
SuperGlue [1]+GA-GM (ours)	no	16.60	32.74	49.79	65.71	25.52
SuperGlue [1]+GA-MGM (ours)	no	17.13	32.76	49.79	64.90	26.19
SuperGlue+GA-MGM (no partial/outliers handling)	no	7.75	20.01	39.96	21.11	18.58
SuperGlue [1]+GA-GM (ours)	yes	26.18	46.99	66.34	68.69	**32.36**
SuperGlue [1]+GA-MGM (ours)	yes	**26.70**	**47.19**	**66.70**	**71.13**	27.51

two popular solvers for two-graph matching [5] and multi-graph matching [9] are also compared, where matching is firstly solved among all graphs, followed by spectral clustering. Table 5 reports the clustering metrics including CP, RI, CA, and intra-ground-truth class matching accuracy (MA). Both smaller-scaled (24 graphs) and larger-scaled (130 graphs) MGM^3 problems are tested, and GANN-MGM^3 outperforms on the smaller-scaled problem and scales up soundly. For unsupervised learning on the MGM^3 task, clustering accuracy might be more important than matching accuracy. Our learning-free GA-MGM^3 achieves accurate clustering but is not as comparative in matching, however, both matching and clustering performances are improved with supervision from GA-MGM^3. Besides, our GANN-MGM^3 is relatively slow on the smaller-scaled problem, but runs comparatively fast with DPMC [12] on the larger-scaled problem, probably because the overhead on VGG16 is more significant on smaller problems. We also find unsupervised learning improves the convergence speed of our graduated assignment method.

5.3.2 Results on CUB2011

For the MGM^3 task, evaluation is conducted on 10 Horned Grebe, 10 Baird Sparrow, and 10 Caspian Tern for 50 trails. This MGM^3 problem on CUB2011 is more challenging than the Willow ObjectClass because all involved images belong to different subcategories of birds and are more difficult to categorize. For the MGM^3 problem, clustering metrics and matching precision (MP), recall (MR), and f1-score (MF) are reported on the testing set, as shown in Fig. 8(a), where unsupervised learning helps to improve all matching and clustering metrics compared to the learning-free version.

5.3.3 Results on Pascal VOC Keypoint

We formulate the MGM^3 problem by a mix of three modes: 5 bicycles, 5 bottles, and 5 tvmonitors randomly sampled from the dataset. Mean and STD are reported with 200 trials in Fig. 8(b). The improvement of clustering metrics brought by unsupervised learning is not as significant as the improvement of matching accuracies on Pascal VOC Keypoint, probably because images within the same category of Pascal VOC still vary greatly in pose and appearance, yielding challenges to the clustering step.

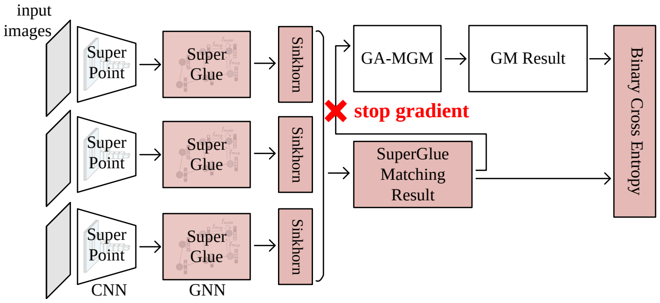
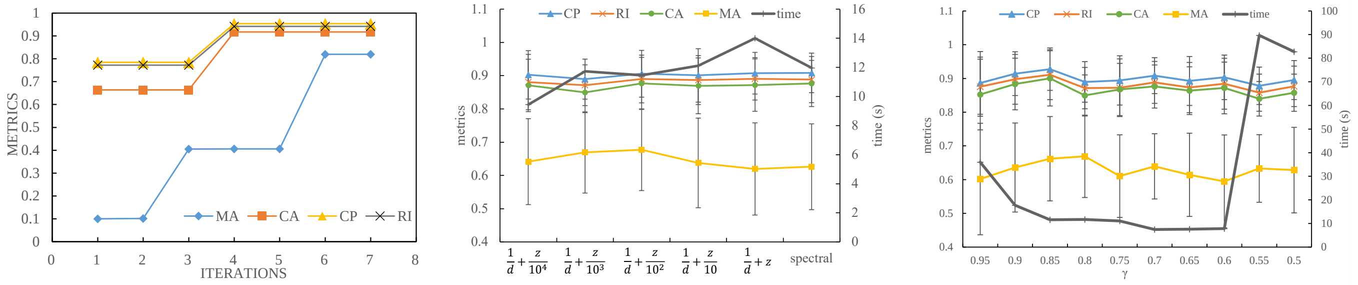


Fig. 9. Our unsupervised learning image matching pipeline. The brown modules support gradient back-propagation and are learned by minimizing the discrepancy between SuperGlue [1] and our GA-MGM.

5.4 Extending Unsupervised Graph Matching to Image Matching and SfM Applications

5.4.1 Unsupervised Natural Image Matching Pipeline

We try to apply our technique to this realistic and popular setting. It involves the following steps: keypoint detection, feature extraction, keypoint matching, and consensus filtering. The filtered matching result is further utilized to estimate the stereo pose or measure the similarity between images. Existing deep graph matching papers focus on the keypoint matching step, assuming that the other upstream and downstream tools are ready. However, it leads to certain ideal assumptions such as not considering partial matching and outliers [9], [23], [36], [37]. With our proposed treatments for partial and outlier matching, we manage to bridge the gap between deep graph matching and natural image matching. Specifically, we follow the novel SuperPoint [69] and SuperGlue [1] pipeline whereby SuperPoint is a pre-



(a) Effect of shrinking β at iterations 3 and 6

(b) Different initialization techniques

(c) Different γ configurations

Fig. 10. Ablation study results of the proposed MGM^3 method on WillowObject Class dataset (in line with the right half of Table 5).

TABLE 7
Matching accuracy on Willow dataset (50 tests), where the backbone net of PIA+GA-MGM is built by extending the VGG16 with PIA-GM [23].

method	learning	car	duck	face	mbike	wbottle
GA-MGM (ours)	free	0.746±0.153	0.900±0.106	0.997±0.021	0.892±0.139	0.937±0.072
PIA+GA-MGM [23]+(ours)	free	0.380±0.103	0.434±0.153	0.484±0.102	0.450±0.161	0.421±0.064
GANN-MGM (ours)	unsup.	0.964±0.058	0.949±0.057	1.000±0.000	1.000±0.000	0.978±0.035
PIA+GANN-MGM [23]+(ours)	unsup.	0.394±0.009	0.493±0.026	0.501±0.009	0.426±0.023	0.478±0.006

TABLE 8
MGM^3 on Willow w/ 40 Cars, 50 Ducks, 40 Motorbikes with PIA + GA-MGM^3 (mean and STD by 50 tests).

method	learning	CP	RI	CA	MA	time (s)
GA-MGM^3 (ours)	free	0.890±0.060	0.871±0.061	0.850±0.061	0.669±0.122	11.7
PIA+GA-MGM^3 [23]+(ours)	free	0.607±0.102	0.645±0.068	0.514±0.102	0.261±0.060	17.9
GANN-MGM^3 (ours)	unsup.	0.974±0.034	0.968±0.035	0.956±0.039	0.906±0.047	9.2
PIA+GANN-MGM^3 [23]+(ours)	unsup.	0.567±0.061	0.633±0.029	0.451±0.044	0.255±0.023	19.2

trained CNN keypoint detector and SuperGlue is the matching module composed of graph attention networks [92] and Sinkhorn [71]. Our GA-GM and GA-MGM algorithms are built on top of SuperGlue's output, to leverage the domain knowledge of tackling image matching problems. Since the final layer of SuperGlue is a Sinkhorn layer, we can treat the SuperGlue network as the CNN backbone in our previous graph matching pipeline and apply the framework for image matching, as shown in Fig. 9. During inference, we adopt the RANSAC algorithm [63] for consensus filtering.

5.4.2 *The Photo Tourism Dataset and Results*

The Photo Tourism dataset is part of the annual image matching challenge [93] composed of 25061 images, collected from photos at 16 tourism attractions around the world[7] from Yahoo Flickr[8]. It focuses on the stereo pose estimation task for structure-from-motion (SfM). The (pseudo) ground truth camera poses are established with the off-the-shelf SfM software colmap [94], [95] by jointly matching all images from the same scene. Participant models are tested on a harder problem of matching images and estimating the relative poses given only a small subset of data. Besides, this dataset suffers from partial matching due to occlusion and view-point changes, and there exist unavoidable outliers found by the keypoint detector. Therefore, this dataset is a wonderful testbed for validating the robustness of our approach and the performance of real-world downstream tasks. We adopt the pretrained "outdoor" weights released by the SuperGlue [1] authors and tune the SuperGlue network on the training set by unsupervised learning.

Table 6 shows the evaluation metrics and results. Our learning-free GM and MGM algorithms improve the matching precision and MScore, such that the SuperGlue model is guided to more accurate matching during the discrepancy-minimization procedure. Increased precision and MScore also lead to higher accuracy on the downstream SfM task, suggesting the potential of further applying our method to other natural image matching tasks.

5.5 Additional Results and Further Discussion

5.5.1 *Root Analysis of Graduated Assignment on MGM^3*

The motivation of our GA-MGM^3 method is that more precise multi-graph matching will improve the clustering accuracy and vice versa. In Fig. 10(a), we plot the matching and clustering metrics for the learning-free GA-MGM^3 model on Willow ObjectClass (in line with right half of Table 5). In Fig. 10(a), β drops from 1 to 0.9 at iteration 3, and drops from 0.9 to 0 at iteration 6. We observe that more accurate clustering improves matching accuracy and vice versa, finally reaching a satisfying matching and clustering result. Our motivation of developing GA-MGM^3 is validated in this root analysis.

5.5.2 *Ablation Study for Algorithmic Configurations and Hyper-parameters*

Ablation studies are conducted on the MGM^3 problem on Willow dataset in line with the right half of Table 5. For the initialization of $\{\mathbf{U}_i\}$, we experiment with other initialization methods, e.g. initialize by spectral multi-matching [10].

[7] https://www.cs.ubc.ca/~kmyi/imw2020/data.html
[8] https://www.flickr.com/

As shown in Fig. 10(b), different denominator configurations are compared with the spectral multi-matching technique [10], and our method seems to be insensitive to different initialization methods. Thus we adopt random initialization for its cost-efficiency in practice.

We also test different configurations of γ from 0.5 to 0.95 at the interval of 0.05. As shown in Fig. 10(c), the inference time drops as γ grows, and the matching accuracy (MA) peaks at $\gamma = 0.8$. For $\gamma < 0.6$, the graduated assignment method becomes hard to converge, resulting in relatively lower accuracy and slower inference. The clustering metrics do not change significantly with γ. Our selected $\gamma = 0.8$ achieves the best average MA and moderately good clustering results with satisfying inference time.

5.5.3 Feature Extraction with GNN

Many recent efforts in deep GM involve learning with graph neural networks (GNNs) [23], [34], [35]. Here we experiment our method with graph convolutional network (GCN), which can be viewed as extending the VGG16 CNN with PIA-GM [23] (contains a VGG16 and 3-layer GCN). The more powerful PCA-GM in [23] is not considered because it is nontrivial to define the cross-graph convolution operation when jointly matching multiple graphs. In this experiment, the VGG16 net of PIA-GM is initialized with ImageNet classification weights and the GCN layers are randomly initialized following [23]. As shown in Table 7 and Table 8 where MGM and MGM^3 problems on Willow dataset are considered, respectively. When involving GCN, both matching and clustering accuracies are inferior compared to the GCN-free counterparts. Results in Table 7 and Table 8 show that initialization is important for unsupervised learning, and random initialization may be inadequate for the GCN layers. Other initialization techniques may be adopted for the GCN layers, but they are beyond the scope of this paper.

5.5.4 Visualization of Matching with a Mixture of Modes

A visualization of the prediction of GANN-MGM^3 is shown in Fig. 11, where images are embedded to 2D space based on their graph-wise similarity scores obtained by Eq. 13. Multidimensional scaling is adopted as the embedding technique. As shown in Fig. 11, most Honored Grebes are swimming, most Baird Sparrows are standing and most Caspian Terns are flying, and these three poses distribute in three directions of the embedded space. Fig. 11 suggests that the pose of birds may be a key ingredient for our algorithm to distinguish different birds, and even the misclassified image is confusing considering its pose. Pose information is mainly encoded by edges, and in this paper, we mainly focus on utilizing graph information (i.e. node information and edge information) that is available for the most general MGM^3 task to distinguish different categories.

6 Conclusions

We have presented a unified unsupervised graph matching learning method for two-graph and multi-graph matching, as well as the realistic setting with a mixture of modes, where an off-the-shelf graph matching solver and a differentiable Sinkhorn net offer two different matching predictions, resulting in a discrepancy minimization pipeline. Besides, a unified embodiment of graduated assignment algorithm is proposed to serve as the traditional solver. Promising results are obtained showing the feasibility and advantage of introducing deep networks, especially unsupervised deep networks, to such challenging combinatorial problems.

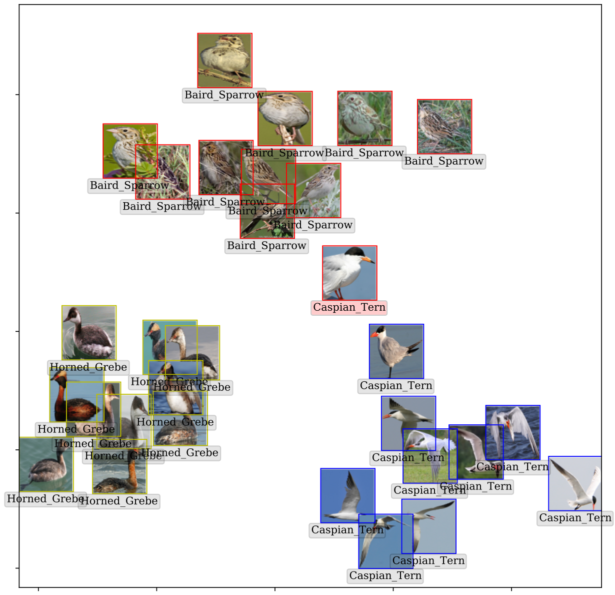


Fig. 11. Visualization of clustering by our method on CUB2011, by embedding the graph-wise distances to 2D via multidimensional scaling. The label under the images denotes their ground truth, and the color of the outer box (red/blue/yellow) shows the predicted modes. We mark the only misclassified "Caspian Tern" with red label background.

Acknowledgments

This work is in part supported by National Key Research and Development Program of China (2020AAA0107600), NSFC (62222607), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). The author Runzhong Wang is also partly sponsored by Wen-Tsun Wu Honorary Doctoral Scholarship, AI Institute, Shanghai Jiao Tong University.

References

[1] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Comput. Vis. Pattern Recog.*, 2020, pp. 4938–4947.

[2] X. Du, J. Yan, and H. Zha, "Joint link prediction and network alignment via cross-graph embedding," *Int. Joint Conf. Artificial Intell.*, pp. 2251–2257, 2019.

[3] S. Yang, J. Tian, H. Zhang, J. Yan, H. He, and Y. Jin, "Transms: Knowledge graph embedding for complex relations by multidirectional semantics," in *Int. Joint Conf. Artificial Intell.*, 2019, pp. 1935–1942.

[4] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido, "A survey for the quadratic assignment problem," *Eur. J. Operational Research*, vol. 176, no. 2, pp. 657–690, 2007.

[5] M. Cho, J. Lee, and K. M. Lee, "Reweighted random walks for graph matching," in *Eur. Conf. Comput. Vis.*, 2010, pp. 492–505.

[6] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Int. Conf. Comput. Vis.*, 2005, pp. 1482–1489.

[7] T. Yu, J. Yan, Y. Wang, W. Liu *et al.*, "Generalizing graph matching beyond quadratic assignment model," in *Neural Info. Process. Systems*, 2018, pp. 853–863.

[8] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu, "Consistency-driven alternating optimization for multigraph matching: A unified approach," *Trans. Image Process.*, vol. 24, no. 3, pp. 994–1009, 2015.

[9] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, 2016.

[10] D. Pachauri, R. Kondor, and V. Singh, "Solving the multi-way matching problem by permutation synchronization," in *Neural Info. Process. Systems*, 2013, pp. 1860–1868.

[11] Q. Wang, X. Zhou, and K. Daniilidis, "Multi-image semantic matching by mining consistent features," in *Comput. Vis. Pattern Recog.*, 2018, pp. 685–694.

[12] T. Wang, Z. Jiang, and J. Yan, "Clustering-aware multiple graph matching via decayed pairwise matching composition," *AAAI Conf. Artificial Intell.*, pp. 7–12, 2020.

[13] J. He, Z. Huang, N. Wang, and Z. Zhang, "Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking," in *Comput. Vis. Pattern Recog.*, 2021, pp. 5299–5309.

[14] K. Fu, S. Liu, X. Luo, and M. Wang, "Rgm: Robust point cloud registration framework based on deep graph matching," in *Comput. Vis. Pattern Recog.*, 2021, pp. 5299–5309.

[15] N. Dym, H. Maron, and Y. Lipman, "DS++: a flexible, scalable and provably tight relaxation for matching problems," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–14, 2017.

[16] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 377–388, 1996.

[17] C. Edwards, "A branch and bound algorithm for the koopmans-beckmann quadratic assignment problem," in *Combinatorial optimization II*. Springer, 1980, pp. 35–52.

[18] P. Hahn, T. Grant, and N. Hall, "A branch-and-bound algorithm for the quadratic assignment problem based on the hungarian method," *EJOR*, vol. 108, no. 3, pp. 629 – 640, 1998.

[19] T. Caetano, J. McAuley, L. Cheng, Q. Le, and A. J. Smola, "Learning graph matching," *Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1048–1058, 2009.

[20] M. Leordeanu, R. Sukthankar, and M. Hebert, "Unsupervised learning for graph matching," *Int. J. Comput. Vis.*, vol. 96, no. 1, pp. 28–45, 2012.

[21] M. Cho, K. Alahari, and J. Ponce, "Learning graphs to match," in *Int. Conf. Comput. Vis.*, 2013, pp. 25–32.

[22] A. Zanfir and C. Sminchisescu, "Deep learning of graph matching," in *Comput. Vis. Pattern Recog.*, 2018, pp. 2684–2693.

[23] R. Wang, J. Yan, and X. Yang, "Learning combinatorial embedding networks for deep graph matching," in *Int. Conf. Comput. Vis.*, 2019, pp. 3056–3065.

[24] Z. Zhang and W. S. Lee, "Deep graphical feature learning for the feature matching problem," in *Int. Conf. Comput. Vis.*, 2019, pp. 5087–5096.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Rep.*, 2014.

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *Int. Conf. Learn. Rep.*, 2017.

[27] G. D. Lowe, "Object recognition from local scale-invariant features," *Int. Conf. Comput. Vis.*, pp. 1150–1150, 1999.

[28] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *Int. Conf. Comput. Vis.*, 2009, pp. 1365–1372.

[29] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[30] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[31] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, koray kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," in *Neural Info. Process. Systems*, 2020.

[32] J. Bromley, I. Guyon, Y. LeCun, E. Sackinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Neural Info. Process. Systems*, 1994.

[33] R. Wang, J. Yan, and X. Yang, "Graduated assignment for joint multi-graph matching and clustering with application to unsupervised graph matching network learning," in *Neural Info. Process. Systems*, 2020.

[34] T. Yu, R. Wang, J. Yan, and B. Li, "Learning deep graph matching with channel-independent embedding and hungarian attention," in *Int. Conf. Learn. Rep.*, 2020.

[35] M. Rolínek, P. Swoboda, D. Zietlow, A. Paulus, V. Musil, and G. Martius, "Deep graph matching via blackbox differentiation of combinatorial solvers," in *Eur. Conf. Comput. Vis.*, 2020, pp. 407–424.

[36] R. Wang, J. Yan, and X. Yang, "Neural graph matching network: Learning lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching," *Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5261–5279, 2022.

[37] Z. Jiang, T. Wang, and J. Yan, "Unifying offline and online multi-graph matching via finding shortest paths on supergraph," *Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3648–3663, 2020.

[38] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[39] X. Chen and K. He, "Exploring simple siamese representation learning," in *Comput. Vis. Pattern Recog.*, 2021, pp. 15 750–15 758.

[40] Y. You, T. Chen, Z. Wang, and Y. Shen, "When does self-supervision help graph convolutional networks?" *Int. Conf. Mach. Learn.*, pp. 10 871–10 880, 2020.

[41] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, "Gpt-gnn: Generative pre-training of graph neural networks," in *ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, 2020, pp. 1857–1867.

[42] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," *Int. Conf. Learn. Rep.*, 2020.

[43] A. K. Sankararaman, S. De, Z. Xu, R. W. Huang, and T. Goldstein, "Contrastive multi-view representation learning on graphs," *Int. Conf. Mach. Learn.*, pp. 4116–4126, 2020.

[44] P. Zhen, H. Wenbing, L. Minnan, Z. Qinghua, R. Yu, X. Tingyang, and H. Junzhou, "Graph representation learning via graphical mutual information maximization," *WWW '20: The Web Conference 2020*, pp. 259–270, 2020.

[45] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "GCC: graph contrastive coding for graph neural network pre-training," in *ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, 2020, pp. 1150–1160.

[46] J. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[47] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. Efros, "Learning dense correspondence via 3d-guided cycle consistency," in *Comput. Vis. Pattern Recog.*, 2016, pp. 117–126.

[48] X. Wang, A. Jabri, and A. Efros, "Learning correspondence from the cycle-consistency of time," in *Comput. Vis. Pattern Recog.*, 2019, pp. 2566–2576.

[49] P. Truong, M. Danelljan, F. Yu, and L. V. Gool, "Warp consistency for unsupervised learning of dense correspondences," in *Int. Conf. Comput. Vis.*, 2021, pp. 10 346–10 356.

[50] J. Kim, K. Ryoo, J. Seo, G. Lee, D. Kim, H. Cho, and S. Kim, "Semi-supervised learning of semantic correspondence with pseudo-labels," in *Comput. Vis. Pattern Recog.*, 2022, pp. 19 699–19 709.

[51] Y. Tian, J. Yan, H. Zhang, Y. Zhang, X. Yang, and H. Zha, "On the convergence of graph matching: Graduated assignment revisited," in *Eur. Conf. Comput. Vis.*, 2012, pp. 821–835.

[52] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in *Neural Info. Process. Systems*, 2006.

[53] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and map inference," in *Neural Info. Process. Systems*, 2009, pp. 1114–1122.

[54] F. Bernard, J. Thunberg, P. Swoboda, and C. Theobalt, "HiPPI: Higher-order projected power iterations for scalable multi-matching," in *Int. Conf. Comput. Vis.*, 2019, pp. 10 284–10 293.

[55] P. Swoboda, A. Mokarian, C. Theobalt, F. Bernard *et al.*, "A convex relaxation for multi-graph matching," in *Comput. Vis. Pattern Recog.*, 2019, pp. 11 156–11 165.

[56] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, "A short survey of recent advances in graph matching," in *Int. Conf. Multimedia Retrieval*, 2016, pp. 167–174.

[57] J. Yan, S. Yang, and E. R. Hancock, "Learning for graph matching and related combinatorial optimization problems," in *Int. Joint Conf. Artificial Intell.*, 2020, pp. 4988–4996.

[58] R. Wang, J. Yan, and X. Yang, “Combinatorial learning of robust deep graph matching: an embedding based approach,” *Trans. Pattern Anal. Mach. Intell.*, 2020.

[59] T. Wang, H. Liu, Y. Li, Y. Jin, X. Hou, and H. Ling, “Learning combinatorial solver for graph matching,” in *Comput. Vis. Pattern Recog.*, 2020, pp. 7568–7577.

[60] T. Yu, R. Wang, J. Yan, and B. Li., “Deep latent graph matching,” in *Int. Conf. Mach. Learn.*, 2021, pp. 12 187–12 197.

[61] H. Liu, T. Wang, Y. Li, C. Lang, Y. Jin, and H. Ling, “Joint graph learning and matching for semantic feature correspondence,” *Pattern Recognition*, vol. 134, p. 109059, 2023.

[62] R. Tron, X. Zhou, C. Esteves, and K. Daniilidis, “Fast multi-image matching via density-based clustering,” in *Comput. Vis. Pattern Recog.*, 2017, pp. 4057–4066.

[63] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[64] L. Torresani, V. Kolmogorov, and C. Rother, “A dual decomposition approach to feature correspondence,” *Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 259–271, 2012.

[65] X. Yang, H. Qiao, and Z.-Y. Liu, “Outlier robust point correspondence based on gnccp,” *Pattern Recognition Letters*, vol. 55, pp. 8–14, 2015.

[66] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, “Image matching from handcrafted to deep features: A survey,” *Int. J. Comput. Vis.*, no. 1, 2021.

[67] C. G. Harris and M. J. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*, 1988, pp. 147–151.

[68] J. Shi and C. Tomasi, “Good features to track,” in *Comput. Vis. Pattern Recog.*, 2002, pp. 593–600.

[69] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *Comput. Vis. Pattern Recog.*, 2018, pp. 224–236.

[70] A. B. Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, “Key.net: Keypoint detection by handcrafted and learned cnn filters,” in *Int. Conf. Comput. Vis.*, 2019, pp. 5836–5844.

[71] R. Sinkhorn and A. Rangarajan, “A relationship between arbitrary positive matrices and doubly stochastic matrices,” *Ann. Math. Statistics*, vol. 35, no. 2, pp. 876–879, 1964.

[72] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, “Loftr: Detector-free local feature matching with transformers,” in *Comput. Vis. Pattern Recog.*, 2021.

[73] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[74] K. He, G. Gkioxari, P. Dollár, and B. R. Girshick, “Mask r-cnn,” *Int. Conf. Comput. Vis.*, pp. 386–397, 2017.

[75] H. W. Kuhn, “The hungarian method for the assignment problem,” in *Export. Naval Research Logistics Quarterly*, 1955, pp. 83–97.

[76] J. J. Kosowsky and A. L. Yuille, “The invisible hand algorithm: solving the assignment problem with statistical physics,” *Neural Networks*, vol. 7, no. 3, pp. 477–490, 1994.

[77] S. Gold and A. Rangarajan, “Softmax to softassign: neural network algorithms for combinatorial optimization,” *J. Artif. Neural Netw*, vol. 2, no. 4, pp. 381–399, 1996.

[78] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” *Neural Info. Process. Systems*, pp. 2292–2300, 2013.

[79] T. C. Koopmans and M. Beckmann, “Assignment problems and the location of economic activities,” *Econometrica*, pp. 53–76, 1957.

[80] A. Solé-Ribalta and F. Serratosa, “Graduated assignment algorithm for finding the common labelling of a set of graphs,” in *Structural, Syntactic, and Statistical Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 180–190.

[81] X. Zhou, M. Zhu, and K. Daniilidis, “Multi-image matching via fast alternating minimization,” in *Int. Conf. Comput. Vis.*, 2015, pp. 4032–4040.

[82] Y. Chen, L. Guibas, and Q. Huang, “Near-optimal joint object matching via convex relaxation,” in *Int. Conf. Mach. Learn.*, 2014, pp. 100–108.

[83] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” in *ACM-SIAM Symposium on Discrete Algorithms*, 2007, p. 1027–1035.

[84] A. Rangarajan, A. Yuille, and E. Mjolsness, “Convergence properties of the softassign quadratic assignment algorithm,” *Neural Computation*, vol. 11, no. 6, pp. 1455–1474, 1999.

[85] M. Fey, J. E. Lenssen, C. Morris, J. Masci, and N. M. Kriege, “Deep graph matching consensus,” in *Int. Conf. Learn. Rep.*, 2020.

[86] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2005.

[87] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[88] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” California Institute of Technology, Tech. Rep., 2007.

[89] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[90] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

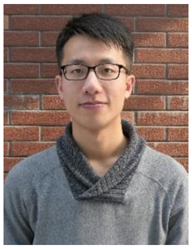
[91] M. Fey, J. Eric Lenssen, F. Weichert, and H. Müller, “SplineCNN: Fast geometric deep learning with continuous b-spline kernels,” in *Comput. Vis. Pattern Recog.*, 2018, pp. 869–877.

[92] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *stat*, vol. 1050, p. 20, 2017.

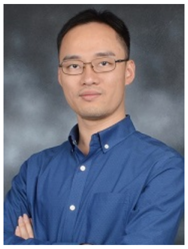
[93] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls, “Image matching across wide baselines: From paper to practice,” *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 517–547, 2020.

[94] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Comput. Vis. Pattern Recog.*, 2016, pp. 4104–4113.

[95] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, “Pixelwise view selection for unstructured multi-view stereo,” in *Eur. Conf. Comput. Vis.*, 2016, pp. 501–518.



Runzhong Wang (S’21) is a PhD Candidate with Department of Computer Science and Engineering, and AI Institute, Shanghai Jiao Tong University. He obtained B.E. in Electrical Engineering from Shanghai Jiao Tong University. He has published first-authored papers in ICCV, CVPR, NeurIPS, and IEEE TPAMI on learning for combinatorial optimization. He is maintaining a graph matching repository: github.com/Thinklab-SJTU/ThinkMatch with 700 stars.



Junchi Yan (S’10-M’11-SM’21) is currently an Associate Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University. Before that, he was a Principal Scientist with IBM Research – China, where he started his career in April 2011. He obtained the Ph.D. in Electrical Engineering from Shanghai Jiao Tong University. He regularly serves as Area Chair for CVPR, AAAI, ICML, NeurIPS etc. and Associate Editor for Pattern Recognition.

Xiaokang Yang (M’00-SM’04-F’19) received the B. S. degree from Xiamen University, in 1994, the M. S. from Chinese Academy of Sciences in 1997, and the Ph.D. from Shanghai Jiao Tong University in 2000. He is currently a Distinguished Professor of AI Institute, Shanghai Jiao Tong University, Shanghai, China. His research interests are image processing and computer vision. He serves as an Associate Editor of IEEE Transactions on Multimedia.