Learning with Partial and Noisy Correspondence in Graph Matching

Yijie Lin, Mouxing Yang, Peng Hu, Jiancheng Lv, Hao Chen, Xi Peng

***Abstract*—The success of existing graph matching methods heavily relies on high-quality training data with complete and precise correspondences between keypoints across different graphs. However, this assumption is often violated in real-world scenarios, leading to partial correspondence and noisy correspondence challenges. In brief, partial correspondence arises from viewpoint occlusions, where certain keypoints (*i.e.*, outliers) lack valid counterparts in the target graph, while noisy correspondence refers to both incorrectly established (*i.e.*, false positives) and neglected (*i.e.*, false negatives) correspondences due to annotation error. In this paper, we propose the first unified framework to address both partial and noisy correspondence challenges in graph matching. Specifically, we introduce a dual-expert cooperative framework that integrates Koopmans-Beckmann and Lawler's quadratic assignment programming formulations (KB-QAP and L-QAP) through an align-fuse-refine pipeline. In the alignment stage, the KB-QAP expert aligns keypoints and distinguishes inliers from outliers using a novel quadratic contrastive loss. In the fusion stage, the L-QAP expert employs a graph transformer on the association graph to merge the aligned graphs and incorporates a learnable outlier-rejection mechanism to handle partial correspondences. Finally, by exploiting the different noise resistances of the two experts, we identify and refine the false positive and false negative correspondences, thereby enhancing robustness against noisy correspondence. Extensive experiments on four widely-used graph matching datasets demonstrate the effectiveness of our method against 17 competitive baselines in both partial and noisy correspondence scenarios. The code is available at https://github.com/XLearning-SCU/2026-TPAMI-COMMON.**

***Index Terms*—Graph Matching, Noisy Correspondence, Partial Correspondence**

I. Introduction

GRAPH matching [1] seeks to establish correspondences between keypoints of different graphs, serving as a cornerstone for various applications such as object tracking [2], [3], cross-domain alignment [4], structure-from-motion [5], and SLAM [6]. The core of graph matching lies in exploring and exploiting bi-level affinities between graphs, *i.e.* node-to-node (*linear*) and edge-to-edge (*quadratic*) affinities. By encoding the high-order geometric information through advanced graph neural networks [7]–[10] and adopting differentiable quadratic loss functions [11], [12], graph matching methods have achieved promising performance in correspondence estimation.

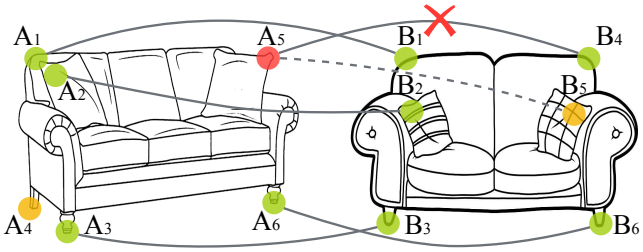


Fig. 1. Illustration of partial and noisy correspondence. In this manually annotated example, green dots denote correctly annotated keypoints, red dots denote mislabeled keypoints, and yellow dots denote outliers. Solid lines indicate the provided (possibly corrupted) correspondences, while dashed lines indicate overlooked correspondences. **Partial correspondence** occurs when certain keypoints have no valid counterpart, *e.g.*, outlier A_4 (left rear leg) is unaligned due to occlusion. **Noisy correspondence** refers to the supervision noise. A mislabeled keypoint can induce a *false positive* correspondence, *e.g.*, A_5 (right throw pillow) is mislabeled as "right backrest" and thus incorrectly matched to B_4. It can also lead to *false negative* by missing true matches, *e.g.*, the true counterpart B_5 of A_5 is mistakenly labeled as an outlier, causing the correspondence $\mathrm{A}_5\mathrm{B}_5$ to be overlooked.

Despite these advancements, most graph matching methods often rely on an implicit yet unrealistic assumption, *i.e.*, all keypoints are fully and accurately matched across graphs in training data. However, in practice, this assumption is rarely satisfied due to occlusions [9], large viewpoint variations [13], and poor recognizability [14], leading to two distinct challenges: *partial correspondence* and *noisy correspondence*. As shown in Fig. 1, partial correspondence [15] occurs when certain keypoints (*i.e.*, outliers) lack a valid counterpart in the target image, typically due to the object being occluded or partially visible. This challenge is common in graph matching for both manually annotated [13] and automatically detected keypoints [16]. In contrast, noisy correspondence [17] refers to corrupted training supervision: some annotated keypoint pairs are incorrect, which induces false positives and false negatives as illustrated in Fig. 1. To be specific, false positives occur when irrelevant keypoints are mistakenly regarded as matches, while false negatives refer to valid correspondences that are overlooked.

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant CJ202303, CJ202403; in part by NSFC under Grant 624B2099, U25A201523, 62472295; in part by Sichuan Science and Technology Planning Project under Grant 24NSFTD0130; in part by Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM610; in part by System of Systems and Artificial Intelligence Laboratory pioneer fund grant under Grant HLJGGG20240327517-15.

Y. Lin, M. Yang, P. Hu, and J. Lv are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: {linyijie.gm, yangmouxing, penghu.ml}@gmail.com; lvjiancheng@scu.edu.cn). H. Chen is with the Department of Computer Science and Engineering, the Department of Chemical and Biological Engineering, and the Division of Life Science, Hong Kong University of Science and Technology, Hong Kong, China (e-mail: jhc@cse.ust.hk). X. Peng is with the College of Computer Science, Sichuan University, Chengdu 610065, China, and also with the National Key Laboratory of Fundamental Algorithms and Models for Engineering Simulation, Sichuan University, Chengdu 610065, China (e-mail: pengx.gm@gmail.com).

Corresponding author: Xi Peng.

Both challenges can severely degrade graph matching performance. On the one hand, outliers in partial correspondence disrupt the alignment between valid keypoints, leading to misassignments. On the other hand, noisy correspondence hampers both node- and edge-level representation learning and causes cumulative errors, as graph-based models rely on effective information propagation and aggregation. Therefore, addressing both challenges is critical to improving the robustness of graph matching methods.

To tackle the partial correspondence challenge, partial graph matching methods [9], [18], [19] typically leverage simple cues to identify outliers and use them as auxiliary supervision during training, enabling the model to better distinguish between outliers and inliers during inference. Such cues include annotated keypoint types (*e.g.*, the absence of a "left rear leg" in the right image of Fig. 1) or multiple-view geometry principles [20]. This preprocessing enables models to effectively learn outlier patterns, improving their ability to filter outliers during matching. However, these methods mainly focus on outlier detection and assume that the annotated correspondences are reliable, without accounting for corrupted supervision that induces false positives and false negatives.

As a result, learning under noisy correspondence remains challenging and comparatively less explored in graph matching due to the following reasons: First, accurately identifying noisy correspondences before training is nearly impossible without prior engineering, such as manually curated verification labels [21]. Second, graph matching is a constrained combinatorial optimization problem where each keypoint is strictly aligned to only one counterpart. In contrast, existing noisy correspondence methods [22] are designed for unconstrained one-to-many matching scenarios, such as cross-modal retrieval where a single query can correspond to multiple candidates. Due to strict one-to-one alignment constraints, directly adopting existing noisy correspondence learning methods is nontrivial for graph matching.

Despite recent progress in tackling partial and noisy correspondence separately, no unified framework has been developed to address both challenges simultaneously. In fact, an effective graph matching method must not only handle noisy correspondence during training but also filter out outliers during inference. To this end, we propose COntrastive Matching with Momentum cOoperatioN (COMMON$^+$), the first unified framework designed to tackle both challenges in graph matching.

COMMON$^+$ is inspired by the complementary nature of two fundamental quadratic assignment problem (QAP) formulations: Koopmans-Beckmann's QAP (KB-QAP) prioritizes graph alignment and Lawler's QAP (L-QAP) focuses on graph fusion. Unlike existing methods that typically adopt either formulation in isolation, we propose a dual-expert cooperative framework that integrates both paradigms through an align-fuse-refine process, specifically designed to address partial and noisy correspondence challenges.

To be specific, COMMON$^+$ consists of the following three key components: i) Graph alignment (KB-QAP expert). The goal of this stage is to project different graphs into a shared space for preliminary keypoint alignment. Specifically, we encode each graph independently using a Siamese graph network, equipped with a novel quadratic contrastive loss. This loss integrates both linear and quadratic geometric relationships and dynamically estimates a partition threshold to distinguish inliers from outliers. ii) Graph fusion (L-QAP expert). Once the graphs are aligned, we resolve outliers and establish more accurate correspondences through graph fusion. Specifically, we apply a graph transformer on the association graph, which is constructed by explicitly fusing the two graphs via L-QAP. To address partial correspondence, we incorporate a learnable outlier-rejection mechanism within the Sinkhorn algorithm to effectively filter out outliers. iii) Momentum cooperation. To mitigate error propagation from noisy correspondence, we introduce a momentum-based cooperation mechanism, where the two complementary experts cooperatively refine the supervision signals. Specifically, we apply the Hungarian algorithm [23] separately to KB-QAP and L-QAP experts. By cross-referencing predicted assignments with the annotated permutation matrix, we dynamically adjust supervision and enhance robustness through bootstrapping.

In summary, the contributions are given as follows:

- We reveal a crucial yet less-explored challenge in graph matching, termed noisy correspondence, which includes mislabeled keypoint pairs (false positives) and overlooked matches (false negatives).
- We propose a unified framework that simultaneously addresses partial and noisy correspondence by integrating KB-QAP and L-QAP formulations within two complementary graph networks. Extensive experiments on real-world datasets demonstrate that these networks collaborate effectively, improving the robustness of graph matching and achieving state-of-the-art performance.
- To solve the KB-QAP objective, we extend the standard linear contrastive loss with quadratic geometric information. By introducing two novel graph consistency regularizers, our loss function effectively captures high-order structural relationships.

II. Related work

In this section, we briefly review three topics relevant to this study: deep graph matching, partial graph matching, and contrastive learning.

A. Deep Graph Matching

Deep graph matching [24]–[26] aims to align keypoints between graphs by leveraging node-to-node and edge-to-edge correlations. Methods in this area can be broadly classified based on how they incorporate high-order information from graph structures: i) network-based approaches [10], [27]–[29] integrate high-order structural information through customized network architectures. For instance, SuperGlue [9] employs self- and cross-attention mechanisms to capture structural relationships within and between graphs. GMTR [30], on the other hand, modifies the vision transformer to encode both patches and keypoints as sequential data. ii) optimization-driven approaches [31] employ differentiable optimization strategies to capture high-order information. For example, QCDGM [11]

incorporates a differentiable Frank-Wolfe algorithm to optimize quadratic constraints. Some other works [32], [33] utilize the differentiable black-box solver [12] to address quadratic assignment problems.

Despite their effectiveness, most existing methods assume that the node-to-node and edge-to-edge correspondences are faultless. In practice, this assumption is often unrealistic due to poor annotations and errors in multi-view geometry, resulting in noisy correspondence. Notably, addressing noisy correspondence caused by imprecise annotations is fundamentally different from the perturbed scenarios explored in adversarial attack [34] or certified robustness studies [35].

B. Partial Graph Matching

Partial graph matching [36], [37] addresses scenarios where only a subset of nodes in one graph has valid counterparts in other graphs. Existing approaches for partial graph matching can be broadly categorized into prior-based methods and learning-based methods. Prior-based methods rely on explicit spatial and geometric assumptions, often incorporating domain-specific knowledge. For instance, structure priors like graph isomorphism [38] and transformation priors such as motion, homography, and pose [39], [40] are utilized to guide the matching. Additionally, cycle consistency prior has also been adopted as the regularizer for multiple graph matching [15], [41]. While effective, these methods are constrained by their dependence on predefined assumptions, limiting flexibility in more complex scenarios. In contrast, learning-based methods aim to reduce dependence on rigid priors, offering flexibility and adaptability to diverse matching scenarios. These methods address outliers in a data-driven manner through mechanisms such as learnable thresholds [9] and dummy rows [18], which help to identify valid counterparts from outliers. Unlike partition-based approaches, AFAT [19] explicitly estimates the number of inliers during the matching process.

In this paper, we propose a unified data-driven framework to tackle both partial and noisy correspondence. Leveraging the connection between contrastive learning and graph matching, we treat valid correspondences as positive pairs and outliers as negative samples. By dynamically estimating the partition threshold based on the distance between inliers and outliers, our method achieves state-of-the-art results using a simple threshold-based Hungarian algorithm. This unified approach provides a robust and flexible solution to the challenges of partial and noisy correspondence.

C. Contrastive Learning

Contrastive learning [42]–[49] has emerged as a powerful paradigm for representation learning. The core idea is to pull similar (positive) pairs closer while pushing dissimilar (negative) pairs apart. Classic works like SimCLR [50] and MoCo [51] introduce image-level contrastive losses, treating augmented views of the same image as positive pairs and other images as negative samples. Building on these foundations, subsequent advancements have extended to pixel-level [52]–[54] and graph-level [55], [56] contrastive learning.

This study addresses two critical limitations in existing contrastive learning approaches. First, as highlighted by Moskalev et al. [57], most contrastive learning methods focus exclusively on the instance discrimination problem, which involves aligning pairs of objects (*e.g.*, two pixels or two graphs). This narrow focus overlooks high-order correlations, such as the structural relationships among objects (*e.g.,* the graph structure of keypoints). Second, these methods often rely on the assumption that positive pairs are strongly associated. However, this assumption becomes unreliable in real-world scenarios where noisy correspondence is common. To address these challenges, we propose a quadratic contrastive learning loss and a momentum cooperation strategy. These innovations effectively capture high-order correlations and enhance robustness to noisy correspondences.

D. Differences from the preliminary version

This paper extends our ICCV 2023 work COMMON [17]. Compared with the preliminary version, this manuscript includes: i) *Broader problem scenarios*: we extend from *noisy* correspondence to the more realistic *partial and noisy* correspondence challenge. ii) *New framework*: we propose a multi-expert cooperative *align-fuse-refine* paradigm that unifies KB-QAP and L-QAP with corresponding optimization strategies. Importantly, their complementary predictions are further integrated to refine correspondences via the proposed co-divide and co-refine strategy. iii) *Additional analysis*: we provide new theoretical insight into the proposed graph-geometric regularization. iv) *Expanded experiments*: we broaden the experimental validation in both vision and non-vision scenarios. In vision, we additionally evaluate partial matching with outliers and conduct robustness studies under varying noise and outlier ratios. We further include a more challenging benchmark IMC-PT-GM [19], with up to 50% outliers. Beyond vision, we add social network alignment benchmarks [58]–[60] with graphs containing thousands of nodes (see the supplementary material), demonstrating the scalability and generality of our framework.

III. Method

In this section, we introduce our align-fuse-refine three-step framework for robust graph matching. We first formally define the challenges of partial and noisy correspondence (Section III-A). Then, we present graph alignment via KB-QAP (Section III-B), graph fusion via L-QAP (Section III-C), and momentum cooperation for refining supervision (Section III-D).

A. Problem Definition

Graph matching aims to establish node-to-node correspondences between two graphs $\mathcal{G}_A$ and $\mathcal{G}_B$. Let $\mathcal{G}_A = \{\mathbf{U}_A, \mathbf{E}_A\}$ and $\mathcal{G}_B = \{\mathbf{U}_B, \mathbf{E}_B\}$ represent the two graphs, where $\mathbf{U}_A$ and $\mathbf{U}_B$ are sets of keypoints of sizes n and m ($n \leq m$), and $\mathbf{E}_A, \mathbf{E}_B$ denote the corresponding edge sets constructed by Delaunay Triangulation. The goal is to predict an assignment matrix $\mathbf{Y} \in \mathbb{R}^{n\times m}$ that encodes the matching assignment

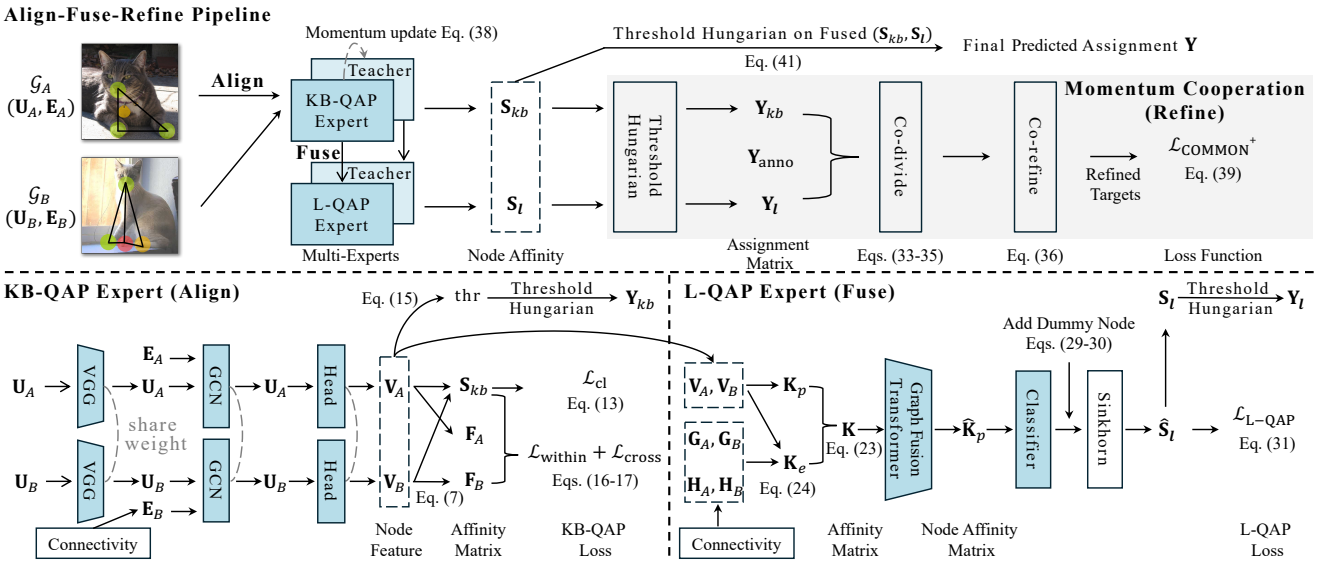


Fig. 2. Overview of our align–fuse–refine framework that integrates KB-QAP and L-QAP formulations in a multi-expert cooperative pipeline. In the figure, blue boxes denote learnable network modules, while transparent boxes denote parameter-free algorithmic operations. (*Bottom Left*) KB-QAP expert: a Siamese graph network embeds each graph into a shared space and predicts alignment affinities. A quadratic contrastive loss further enforces node-to-node and edge-level geometric consistency. (*Bottom Right*) L-QAP expert: an association graph is constructed to fuse information from both graphs. A graph fusion transformer followed by an outlier-aware Sinkhorn algorithm produces fusion-based matching affinities. (*Top Right*) Momentum cooperation: the co-divide and co-refine strategies dynamically refine supervision targets by leveraging complementary predictions from both experts.

between keypoints in $\mathcal{G}_A$ and $\mathcal{G}_B$. In the ideal case, the ground-truth permutation matrix $\mathbf{Y}_{\text{gt}} \in \mathbb{R}^{n\times m}$ specifies the true matching assignment, and we minimize the discrepancy between $\mathbf{Y}$ and $\mathbf{Y}_{\text{gt}}$:

$$\min \mathcal{L}_{\text{Y}} \left(\mathbf{Y}_{\text{gt}}, \mathbf{Y}\right), \tag{1}$$

where $[\mathbf{Y}_{\text{gt}}]_{i,j} = 1$ represents a valid correspondence between the i-th keypoint in $\mathcal{G}_A$ and the j-th keypoint in $\mathcal{G}_B$, while $[\mathbf{Y}_{\text{gt}}]_{i,j} = 0$ indicates that two keypoints are unmatchable. The objective $\mathcal{L}_{\text{Y}}$ measures the discrepancy, commonly using cross-entropy [19] or Hamming distance [12].

However, real-world graph matching is often complicated by *partial* and *noisy* correspondence, both of which significantly degrade matching accuracy. In practice, training supervision is typically provided as an annotated permutation matrix $\mathbf{Y}_{\text{anno}}$, which may be corrupted due to annotation errors. Below, we define partial correspondence with respect to the true matching relation $\mathbf{Y}_{\text{gt}}$, while noisy correspondence characterizes the discrepancy between $\mathbf{Y}_{\text{anno}}$ and $\mathbf{Y}_{\text{gt}}$.

Partial correspondence occurs when certain keypoints lack valid counterparts, often due to occlusions, resulting in outliers that disrupt the matching process.

Definition 1 (**Partial Correspondence**). *If outliers exist in graph $\mathcal{G}_A$ or $\mathcal{G}_B$, the ground-truth assignment satisfies:*

$$\sum_{i,j} [\mathbf{Y}_{\text{gt}}]_{i,j} = k < \max(n, m), \tag{2}$$

where k represents the number of valid correspondences between keypoints. If $\sum_j [\mathbf{Y}_{\text{gt}}]_{i,j} = 0$, the i-th keypoint in $\mathcal{G}_A$ is classified as an outlier. In total, $n + m - 2k$ keypoints are considered outliers.

Noisy correspondence refers to corrupted supervision in $\mathbf{Y}_{\text{anno}}$, which typically arises from annotation errors in challenging images. Note that, the training supervision is $\mathbf{Y}_{\text{anno}}$ rather than $\mathbf{Y}_{\text{gt}}$, and can therefore be imperfect.

Definition 2 (**Noisy Correspondence**). *Noisy correspondence occurs when $\mathbf{Y}_{\text{anno}} \neq \mathbf{Y}_{\text{gt}}$. The positions of false positive and false negative correspondence are given by:*

$$\begin{aligned} \textit{FP-NC} &= \neg\mathbf{Y}_{\text{gt}} \wedge \mathbf{Y}_{\text{anno}}, \\ \textit{FN-NC} &= \mathbf{Y}_{\text{gt}} \wedge \neg\mathbf{Y}_{\text{anno}} \end{aligned} \tag{3}$$

where $\wedge$ denotes the logical AND operation and $\neg$ denotes the logical NOT operation. A value of 1 in the resulting matrices indicates the presence of noisy correspondence at the respective position, where false positives represent irrelevant keypoints incorrectly labeled as matched, and false negatives refer to valid correspondences that are mistakenly omitted.

B. Aligning Graphs through Koopmans-Beckmann's Quadratic Assignment Programming

In this section, we leverage Koopmans-Beckmann's QAP objective in two roles. On the one hand, it serves as an alignment head that produces a KB-QAP matching prediction, which is later fused with the L-QAP prediction in our momentum cooperation strategy (Section III-D). On the other hand, it serves as a training objective that motivates our quadratic contrastive loss for enforcing graph-geometric consistency. In the following, we first present the KB-QAP formulation, and then introduce the Siamese graph network and the proposed quadratic contrastive loss.

Definition 3 (**Koopmans-Beckmann's QAP**). *Let $\mathbf{V}_A \in \mathbb{R}^{n\times d}$ and $\mathbf{V}_B \in \mathbb{R}^{m\times d}$ denote the feature matrices of*

keypoints in graphs $\mathcal{G}_A$ and $\mathcal{G}_B$, respectively. Achieving the matching between two graphs requires optimizing the following objective:

$$\begin{aligned} \underset{\mathbf{Y}}{\operatorname{argmax}} \quad & \underbrace{\operatorname{tr}\left(\mathbf{Y}^\top \mathbf{F}_A \mathbf{Y} \mathbf{F}_B\right)}_{\textit{Quadratic edge affinity}} + \underbrace{\operatorname{tr}\left(\mathbf{S}_{kb}^\top \mathbf{Y}\right)}_{\textit{Linear node affinity}} \\ s.t. \quad & \mathbf{Y} \in \{0,1\}^{n \times m},\ \mathbf{Y}\mathbf{1}_m = \mathbf{1}_n,\ \mathbf{Y}^\top \mathbf{1}_n \leq \mathbf{1}_m \end{aligned} \tag{4}$$

where tr *denotes the trace of the matrix, $\mathbf{S}_{kb} \in \mathbb{R}^{n \times m}$ denotes node-to-node similarity matrix, and $\mathbf{F}_A \in \mathbb{R}^{n \times n}, \mathbf{F}_B \in \mathbb{R}^{m \times m}$ are the adjacency matrices that encode edge information in graphs $\mathcal{G}_A$ and $\mathcal{G}_B$. The constraints on $\mathbf{Y}$ ensure one-to-one matching while accommodating potential outliers.*

1) Graph Alignment Network: As shown in Eq. (4), the KB-QAP formulation jointly maximizes quadratic edge affinity and linear node affinity by independently encoding node and edge features of the two graphs. To implement this, we propose a Siamese graph network that learns structural representations for each graph separately and aligns keypoints in a shared feature space. The network consists of three components: an image encoder, a graph convolutional network, and a projection head.

Image encoder. Following recent graph matching methods [15], [19], [34], [61], we employ VGG16 [62] as the image encoder to extract node features. Specifically, we concatenate the features from relu4_2 and relu5_1 to form the initial node feature matrices $\mathbf{U}_A \in \mathbb{R}^{n \times d}$ and $\mathbf{U}_B \in \mathbb{R}^{m \times d}$, where d is the dimension of the features.

Graph convolutional network. To encode geometric relationships, we initialize the edge structure $\mathbf{E}_A \in \mathbb{R}^{n \times n}$ and $\mathbf{E}_B \in \mathbb{R}^{m \times m}$ with Delaunay triangulation where $[\mathbf{E}]_{i,j} = 1$ if there exists an edge between keypoints i and j, and 0 otherwise. We refine the node features $\mathbf{U}$ using the graph convolutional network SplineConv [63], which updates node representations by aggregating information from neighboring nodes. Formally, the update rule for the i-th keypoint could be expressed as:

$$\operatorname{SplineConv}\left([\mathbf{U}]_i\right) = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \langle [\mathbf{U}]_j, f([\mathbf{E}]_{i,j}) \rangle, \tag{5}$$

where $\mathcal{N}(i)$ indicates the neighbors of keypoint i, $\langle \cdot, \cdot \rangle$ represents the dot product, and f is the B-Spline kernel. The refined node features are obtained by independently feeding graphs $\mathcal{G}_A$ and $\mathcal{G}_B$ into SplineConv.

Projection head. Following classical contrastive learning paradigms [50], [64], we obtain the final node features $\mathbf{V}_A$ and $\mathbf{V}_B$ through a fully connected layer. Formally,

$$\mathbf{V} = \operatorname{norm}\left(g\left(\mathbf{U}\right)\right), \tag{6}$$

where g is a fully connected layer equipped with batch normalization and ReLU activation. The norm operation denotes ℓ_2 normalization, ensuring that the resulting feature vectors have unit norm.

Based on the refined features, we compute the node similarity matrix and the edge adjacency matrices in Eq. (4) as

$$\mathbf{S}_{kb} = \mathbf{V}_A \mathbf{V}_B^\top,\ \mathbf{F}_A = \mathbf{V}_A \mathbf{V}_A^\top,\ \mathbf{F}_B = \mathbf{V}_B \mathbf{V}_B^\top, \tag{7}$$

where $\mathbf{F}_A$ and $\mathbf{F}_B$ are learned affinity matrices derived from node embeddings. This choice is i) compatible with a broad range of GNN backbones, without requiring dedicated edge-feature encoders, and ii) more robust under partial and noisy correspondence, where the raw graph structure (*e.g.*, Delaunay or k-NN) can be unstable.

2) Quadratic Contrastive Loss: Building on combinatorial optimization theory [57] that contrastive learning is equivalent to solving linear assignment problems, we introduce a quadratic contrastive loss that endows contrastive learning with quadratic information. Specifically, the linear assignment problem (corresponding to the second term in Eq. (4)) can be formulated using a structured linear assignment loss [65]:

$$\mathcal{L}_{la} = \max_{\mathbf{Y} \in \Pi} \operatorname{tr}\left(\mathbf{S}_{kb} \mathbf{Y}^\top\right) - \operatorname{tr}\left(\mathbf{S}_{kb} \mathbf{Y}_{\text{anno}}^\top\right), \tag{8}$$

where Π denotes the set of all $n \times m$ permutation matrices that satisfy the constraint in Eq. (4). Note that $\mathcal{L}_{la} \geq 0$ and $\mathcal{L}_{la} = 0$ if and only if the node similarities produced by the Siamese graph network lead to the correct assignment. By minimizing $\mathcal{L}_{la}$, the network learns to correctly assign keypoints from one graph to the other.

Theorem 1 (***Equivalence between Linear Assignment Loss and InfoNCE*** [17], [57])**.** *The log-sum-exp smoothed structured linear assignment loss $\mathcal{L}_{la}$ with row-stochastic relaxation is equivalent to the InfoNCE contrastive loss [64].*

Proof 1. *According to [57], the constraints in Eq. (4) can be relaxed to a row-stochastic condition, i.e., $\mathbf{Y} \in \{0,1\}^{n \times m}$ and $\sum_j [\mathbf{Y}]_{i,j} = 1\ \forall i$. Under this relaxation, Eq. (8) can be reformulated as:*

$$\begin{aligned} \mathcal{L}_{la} &= -\operatorname{tr}\left(\mathbf{S}_{kb} \mathbf{Y}_{\text{anno}}^\top\right) + \max_{\mathbf{Y} \in \mathcal{R}} \operatorname{tr}(\mathbf{S}_{kb} \mathbf{Y}^\top) \\ &= -\operatorname{tr}\left(\mathbf{S}_{kb} \mathbf{Y}_{\text{anno}}^\top\right) + \max_{[\mathbf{Y}]_1 \ldots [\mathbf{Y}]_n} \sum_i \Big(\sum_j [\mathbf{S}_{kb}]_{i,j} \, [\mathbf{Y}]_{i,j}\Big) \\ &= -\operatorname{tr}\left(\mathbf{S}_{kb} \mathbf{Y}_{\text{anno}}^\top\right) + \sum_i \max_{[\mathbf{Y}]_i} \Big(\sum_j [\mathbf{S}_{kb}]_{i,j} \, [\mathbf{Y}]_{i,j}\Big) \\ &= -\operatorname{tr}\left(\mathbf{S}_{kb} \mathbf{Y}_{\text{anno}}^\top\right) + \sum_i \max_j [\mathbf{S}_{kb}]_{ij}\,, \end{aligned} \tag{9}$$

where the third equality follows from the independence of the rows $[\mathbf{Y}]_1, \ldots, [\mathbf{Y}]_n$ and the final equality holds because $[\mathbf{Y}]_i$ is a one-hot vector containing the maximum index. Since Eq. (9) is non-smooth and challenging to optimize, we approximate the max *function through log-sum-exp smoothing technique [66], yielding the InfoNCE contrastive loss:*

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{(i,j) \in \mathbf{Y}_{\text{anno}}} [\mathbf{S}_{kb}]_{i,j} + \tau \sum_i \log\Big(\sum_j \exp\big(\frac{1}{\tau} [\mathbf{S}_{kb}]_{i,j}\big)\Big), \tag{10}$$

where the smoothing parameter τ controls the degree of approximation. □

Notably, Theorem 1 highlights that the contrastive learning formulation provides a fast and differentiable approximation to the linear assignment problem, offering an efficient solution to aligning keypoints.

Robust contrastive loss for partial correspondence. To alleviate the negative impact of outliers in both graphs, we

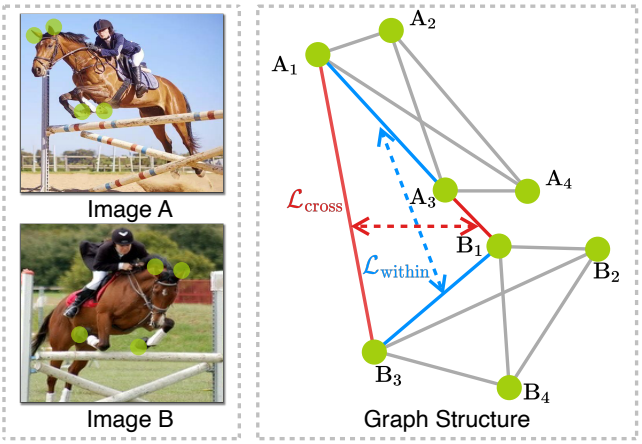


Fig. 3. Illustration of within-graph and cross-graph consistency. (*Left*) Two input images with green keypoints. (*Right*) The red and blue solid lines represent within-graph and cross-graph edges, respectively. The red and blue dashed arrows highlight the within-graph and cross-graph consistency enforced between edges.

simplify the computation of contrastive learning loss by first aligning keypoints $\mathbf{V}_A$ and $\mathbf{V}_B$ based on the annotated permutation matrix $\mathbf{Y}_{\text{anno}}$:

$$\hat{\mathbf{V}}_A, \hat{\mathbf{V}}_B = \{[\mathbf{V}_A]_i\}, \{[\mathbf{Y}_{\text{anno}}\mathbf{V}_B]_i\}, \ \forall \sum_{j=1}^{m} [\mathbf{Y}_{\text{anno}}]_{i,j} > 0, \tag{11}$$

where $\hat{\mathbf{V}}_A$ and $\hat{\mathbf{V}}_B$ consist only of keypoints with valid correspondences. The remaining unmatched keypoints are regarded as outliers.

$$\mathbf{O} = \{[\mathbf{V}_A]_i \mid \sum_{j=1}^{m} [\mathbf{Y}_{\text{anno}}]_{i,j} = 0\} \cup \{[\mathbf{V}_B]_i \mid \sum_{j=1}^{n} [\mathbf{Y}_{\text{anno}}]_{j,i} = 0\}. \tag{12}$$

We then apply contrastive learning to the pairwise similarity matrix by computing the loss across rows and columns:

$$\mathcal{L}_{\text{cl}} = \mathcal{H}\left(\mathbf{I}_n, \rho\left(\hat{\mathbf{V}}_A \hat{\mathbf{V}}_B^\top\right)\right) + \mathcal{H}\left(\mathbf{I}_n, \rho\left(\hat{\mathbf{V}}_B \hat{\mathbf{V}}_A^\top\right)\right), \tag{13}$$

where $\mathbf{I}_n$ is the identity matrix, $\mathcal{H}$ denotes the row-wise cross-entropy function, and ρ represents the softmax function:

$$[\rho\left(\hat{\mathbf{V}}_A \hat{\mathbf{V}}_B^\top\right)]_{i,j} = \frac{\exp([\hat{\mathbf{V}}_A]_i [\hat{\mathbf{V}}_B]_j^\top / \tau)}{\sum_k \exp\left([\hat{\mathbf{V}}_A]_i [\hat{\mathbf{V}}_B]_k^\top / \tau\right) + \sum_l \exp\left([\hat{\mathbf{V}}_A]_i [\mathbf{O}]_l^\top / \tau\right)}. \tag{14}$$

This formulation incorporates all outliers as negative samples in contrastive learning, improving robustness against partial correspondence. During inference, outliers are removed using the Hungarian algorithm with a threshold following [67]. The threshold is estimated based on the average similarity between inliers and outliers:

$$\text{thr} = \frac{1}{2}\left(\frac{\sum_{i,j}[\hat{\mathbf{V}}_A \mathbf{O}^\top]_{i,j}}{|\hat{\mathbf{V}}_A||\mathbf{O}|} + \frac{\sum_{i,j}[\hat{\mathbf{V}}_B \mathbf{O}^\top]_{i,j}}{|\hat{\mathbf{V}}_B||\mathbf{O}|}\right), \tag{15}$$

where $|\cdot|$ denotes the number of elements in the set.

Although the widely adopted InfoNCE loss effectively addresses the linear assignment problem, it overlooks a critical aspect of graph matching: edge alignment. In fact, incorporating edge information into the matching process is desirable for improving robustness in graph matching [7]. Consequently, while contrastive learning provides a solid foundation, it fails to fully exploit graph structures and may result in suboptimal performance. To address this limitation, we introduce a quadratic contrastive loss augmented with two novel graph-geometric consistency regularizers, namely, *within-graph consistency* and *cross-graph consistency*. Both terms operate on inlier correspondences only, since outliers do not have valid counterparts in the target graph.

Within-graph consistency encourages the alignment between edge structures within each graph,

$$\mathcal{L}_{\text{within}} = \|\hat{\mathbf{V}}_A \hat{\mathbf{V}}_A^\top - \hat{\mathbf{V}}_B \hat{\mathbf{V}}_B^\top\|_F^2, \tag{16}$$

where $\|\cdot\|_F$ is the Frobenius norm.

Cross-graph consistency encourages the alignment of the edges across two graphs:

$$\mathcal{L}_{\text{cross}} = \|\hat{\mathbf{V}}_A \hat{\mathbf{V}}_B^\top - \hat{\mathbf{V}}_B \hat{\mathbf{V}}_A^\top\|_F^2. \tag{17}$$

Here, "edges" refer to induced pairwise affinities computed from node embeddings, rather than explicitly learned edge embeddings. Under this induced-affinity view, Proposition 1 shows that $\mathcal{L}_{\text{within}}$ is directly connected to the quadratic affinity term in the KB-QAP objective.

Proposition 1 (***Connection between Within-graph Consistency and Quadratic Affinity***)**.** *The within-graph consistency loss admits the following decomposition:*

$$\mathcal{L}_{\text{within}} = \|\mathbf{F}_A\|_F^2 + \|\mathbf{F}_B\|_F^2 - 2\ \text{tr}\left(\mathbf{Y}_{\text{anno}}^\top \mathbf{F}_A \mathbf{Y}_{\text{anno}} \mathbf{F}_B\right), \tag{18}$$

which reveals that minimizing $\mathcal{L}_{\text{within}}$ implicitly maximizes the quadratic edge affinity in Eq. (4), while the Frobenius-norm terms act as scale regularizers for the adjacency matrices. Specifically, the regularizers penalize redundant correlations between nodes, encouraging the model to selectively preserve the most reliable structural connections.

Derivation 1. *Let $\hat{\mathbf{F}}_A = \hat{\mathbf{V}}_A \hat{\mathbf{V}}_A^\top$ and $\hat{\mathbf{F}}_B = \hat{\mathbf{V}}_B \hat{\mathbf{V}}_B^\top$ denote the adjacency matrices constructed from the aligned keypoints. Expanding the Frobenius norm in $\mathcal{L}_{\text{within}}$ yields:*

$$\begin{aligned} \mathcal{L}_{\text{within}} &= \|\hat{\mathbf{F}}_A - \hat{\mathbf{F}}_B\|_F^2 \\ &= \text{tr}(\hat{\mathbf{F}}_A^\top \hat{\mathbf{F}}_A) + \text{tr}(\hat{\mathbf{F}}_B^\top \hat{\mathbf{F}}_B) - 2\ \text{tr}(\hat{\mathbf{F}}_A^\top \hat{\mathbf{F}}_B). \end{aligned} \tag{19}$$

Given the outlier-filtered node features $\hat{\mathbf{V}}_A = \mathbf{V}_A$ and $\hat{\mathbf{V}}_B = \mathbf{Y}_{\text{anno}}\mathbf{V}_B$, we have $\hat{\mathbf{F}}_A = \mathbf{F}_A$ and $\hat{\mathbf{F}}_B = \mathbf{Y}_{\text{anno}}\mathbf{F}_B\mathbf{Y}_{\text{anno}}^\top$. Based on the orthogonality $\mathbf{Y}_{\text{anno}}^\top \mathbf{Y}_{\text{anno}} = \mathbf{I}$ on the filtered inlier set, the first two terms in Eq. (19) correspond to the regularization terms $\|\mathbf{F}_A\|_F^2$ and $\|\mathbf{F}_B\|_F^2$. For the cross term, using the cyclic property of the trace and the symmetry of $\mathbf{F}_A$, we obtain:

$$\begin{aligned} \text{tr}(\hat{\mathbf{F}}_A^\top \hat{\mathbf{F}}_B) &= \text{tr}\left(\mathbf{F}_A^\top (\mathbf{Y}_{\text{anno}} \mathbf{F}_B \mathbf{Y}_{\text{anno}}^\top)\right) \\ &= \text{tr}\left(\mathbf{Y}_{\text{anno}}^\top \mathbf{F}_A^\top \mathbf{Y}_{\text{anno}} \mathbf{F}_B\right) \\ &= \text{tr}\left(\mathbf{Y}_{\text{anno}}^\top \mathbf{F}_A \mathbf{Y}_{\text{anno}} \mathbf{F}_B\right). \end{aligned} \tag{20}$$

Substituting these terms back completes the derivation. □

Graph consistency offers both theoretical insight and an intuitive geometric interpretation. Proposition 1 connects within-graph consistency to the quadratic affinity term in KB-QAP,

while Fig. 3 visualizes the within- and cross-graph edge relations enforced by our regularizers.

Intuitively, Eq. (16) reduces discrepancies between corresponding edges in $\mathcal{G}_A$ and $\mathcal{G}_B$, such as $\mathrm{A}_1\mathrm{A}_3$ and $\mathrm{B}_1\mathrm{B}_3$. Meanwhile, ensuring cross-graph edge alignment is equally crucial for accurate keypoint matching, particularly when objects exhibit semantic variations (*e.g.*, two kinds of horses). Given associated within-graph edges ($\mathrm{A}_1\mathrm{A}_3$ and $\mathrm{B}_1\mathrm{B}_3$), it is desirable that their corresponding keypoints (A_3 and B_3) are equivalent and interchangeable. To ensure this, we establish cross-graph edges by exchanging the counterparts of each keypoint and then minimize differences between them (*e.g.*, $\mathrm{A}_1\mathrm{B}_3$ and $\mathrm{B}_1\mathrm{A}_3$).

Notably, while prior works [7], [9] utilize cross-graph neural networks to encode information from both graphs, they mainly propagate information implicitly. In contrast, our cross-graph consistency offers explicit quadratic supervision to ensure semantic consistency across different objects. These two geometric consistency terms are seamlessly integrated into the contrastive learning framework as regularizers. The overall quadratic contrastive loss for KB-QAP is formulated as:

$$\mathcal{L}_{\mathrm{KB-QAP}} = \mathcal{L}_{\mathrm{cl}} + \mathcal{L}_{\mathrm{within}} + \mathcal{L}_{\mathrm{cross}}, \tag{21}$$

where all loss terms are weighted equally, avoiding the need for laborious hyper-parameter searching.

C. Fusing Graphs through Lawler's Quadratic Assignment Programming

Building upon the graph alignment achieved through Koopmans-Beckmann's QAP in Section III-B, this section presents a graph fusion strategy by solving Lawler's QAP objective.

Definition 4 (Lawler's QAP). *Lawler's QAP aims to maximize the overall alignment of nodes and edges between two graphs. Formally,*

$$\underset{\mathbf{Y}\in\Pi}{\operatorname{argmax}}\, \operatorname{vec}(\mathbf{Y})^\top \mathbf{K} \operatorname{vec}(\mathbf{Y}), \tag{22}$$

where $\operatorname{vec}(\mathbf{Y})$ *reshapes the matching matrix* $\mathbf{Y} \in \mathbb{R}^{n\times m}$ *into a column vector of size* $nm \times 1$ *by stacking its columns. The affinity matrix* $\mathbf{K} \in \mathbb{R}^{nm\times nm}$ *encodes both node-to-node affinities (diagonal elements) and edge-to-edge affinities (off-diagonal elements). Specifically, for edge* ij *in* $\mathcal{G}_A$ *and edge* ab *in* $\mathcal{G}_B$*, the edge-to-edge affinity is encoded by* $[\mathbf{K}]_{ia,jb} = \phi([\mathbf{E}_A]_{i,j}, [\mathbf{E}_B]_{a,b})$ *where* ϕ *represents a distance metric such as a Gaussian kernel. The node similarities are also encoded when the indices satisfy* $ia = jb$*.*

Following NGM [29], the L-QAP affinity matrix $\mathbf{K}$ is built following the factorized formulation [68]:

$$\mathbf{K} = \operatorname{diag}\left(\operatorname{vec}\left(\mathbf{K}_p\right)\right) + \left(\mathbf{G}_B \otimes \mathbf{G}_A\right) \operatorname{diag}\left(\operatorname{vec}\left(\mathbf{K}_e\right)\right) \left(\mathbf{H}_B \otimes \mathbf{H}_A\right)^\top, \tag{23}$$

where $\otimes$ denotes the Kronecker product and $\operatorname{diag}(\cdot)$ constructs a diagonal matrix from the input matrix. Node-edge incidence matrices $\mathbf{G}_A, \mathbf{H}_A \in \{0,1\}^{n\times e_A}$ and $\mathbf{G}_B, \mathbf{H}_B \in \{0,1\}^{m\times e_B}$ define the connectivity structures of the graphs [68], where e_A and e_B indicate the number of edges in $\mathcal{G}_A$ and $\mathcal{G}_B$, respectively. Specifically, $\mathbf{G}$ corresponds to the incoming edges, while $\mathbf{H}$ represents the outgoing edges. The node-to-node similarity matrix $\mathbf{K}_p \in \mathbb{R}^{n\times m}$ and edge-to-edge similarity matrix $\mathbf{K}_e \in \mathbb{R}^{e_A\times e_B}$ are computed by:

$$\begin{gathered} \mathbf{K}_p = \mathbf{V}_A \mathbf{V}_B^\top,\ \mathbf{K}_e = \hat{\mathbf{E}}_A \hat{\mathbf{E}}_B^\top, \\ \hat{\mathbf{E}}_A = \mathbf{G}_A^\top \mathbf{V}_A - \mathbf{H}_A^\top \mathbf{V}_A,\ \hat{\mathbf{E}}_B = \mathbf{G}_B^\top \mathbf{V}_B - \mathbf{H}_B^\top \mathbf{V}_B. \end{gathered} \tag{24}$$

where edge features $\hat{\mathbf{E}}_A$ and $\hat{\mathbf{E}}_B$ are constructed as the difference of node embedding obtained from the Siamese graph network through Eq. (6).

1) Graph Fusion Transformer: As defined in L-QAP, the affinity matrix $\mathbf{K}$ explicitly integrates the two graphs by encoding both node-to-node and edge-to-edge relationships. By capturing the structural correlations between the graphs, $\mathbf{K}$ ensures that both the keypoint alignments (nodes) and their interconnections (edges) are effectively preserved and fused.

To leverage this affinity matrix, we propose a Graph Fusion Transformer. As illustrated in Fig. 2, the affinity matrix $\mathbf{K}$ can be interpreted as an association graph that encodes the correspondences between $\mathcal{G}_A$ and $\mathcal{G}_B$. In this association graph, vertices represent potential node-to-node correspondences, while edges represent potential edge-to-edge correspondences. Specifically, each vertex $\mathbf{K}_{ia,ia}$ corresponds to the correlation between a pair of nodes $[\mathbf{V}_A]_i$ and $[\mathbf{V}_B]_a$, associated with the matching matrix $[\mathbf{Y}]_{i,a}$. Therefore, the graph matching problem can be reformulated as a *vertex classification task* on the association graph where a value of 1 indicates a valid correspondence, and 0 indicates no correspondence. This reformulation leverages the structure of the association graph to fuse the information from two original graphs, simplifying the process of aligning nodes and edges for efficient graph matching.

To effectively perform the vertex classification task on the association graph, we employ TransformerConv [30], [69] to capture both node-level and edge-level relationships. The vertex feature $[\mathbf{K}_p]_{i,a}$ is updated through an attention mechanism that aggregates information between vertex ia and its neighbors:

$$\mathrm{Attention}_{ia,jb} = \mathrm{Softmax}_{jb}\left(\frac{\mathbf{q}_{ia}\mathbf{k}_{jb}^\top}{\sqrt{d}}\right) \cdot \mathbf{v}_{jb}^\top \tag{25}$$

where jb represents all neighboring vertices of ia in the association graph and d denotes the dimension of $\mathbf{q}_{ia}$. The query and key components are defined as:

$$\mathbf{q}_{ia} = [\mathbf{K}]_{ia,ia}\mathbf{W}_q,\ \mathbf{k}_{jb} = [\mathbf{K}]_{jb,jb}\mathbf{W}_k + [\mathbf{K}]_{ia,jb}\mathbf{W}_e. \tag{26}$$

The value component is computed similarly to the key component:

$$\mathbf{v}_{jb} = [\mathbf{K}]_{jb,jb}\mathbf{W}_v + [\mathbf{K}]_{ia,jb}\mathbf{W}_e, \tag{27}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$, and $\mathbf{W}_e$ are learnable projection matrices. By directly incorporating edge information into the key and value components, this formulation effectively captures both node-to-node and edge-to-edge relationships.

2) Partial Matching with Outlier-rejection Sinkhorn: As graph matching is equivalent to vertex classification on the association graph, we employ a vertex classifier combined with the Sinkhorn algorithm to predict the matching result. Specifically, we build a linear classification head on the updated vertex similarity matrix $\hat{\mathbf{K}}_p \in \mathbb{R}^{n\times m\times d}$ to project it back to an $n \times m$ matrix:

$$\mathbf{C} = \text{Classifier}\left(\hat{\mathbf{K}}_p\right). \tag{28}$$

To address the partial correspondence challenge, we introduce a learnable dummy node [9] by augmenting $\mathbf{C}$ with an additional row and column:

$$[\mathbf{C}]_{i,m+1} = [\mathbf{C}]_{n+1,j} = [\mathbf{C}]_{n+1,m+1} = p,\ \forall i \in [1,n], j \in [1,m], \tag{29}$$

where the learnable value p represents the probability of assigning keypoints to the dummy node, acting as a similarity threshold that distinguishes alignable keypoints from outliers. To ensure each keypoint is either matched to the corresponding keypoint or the dummy node, we augment matching constraints as follows:

$$\mathbf{C}\mathbf{1}_{m+1} = [\mathbf{1}_n^\top, m]^\top,\ \mathbf{C}^\top\mathbf{1}_{n+1} = [\mathbf{1}_m^\top, n]^\top. \tag{30}$$

Next, we employ the Sinkhorn variant IPOT [70] on $\mathbf{C}$ with the newly defined constraints, resulting in the matching matrix $\hat{\mathbf{S}}_l$. The matching loss is then formulated as:

$$\begin{aligned}\mathcal{L}_{\text{L-QAP}} = -\sum_{i,j}^{n+1,m+1} \Big(&[\hat{\mathbf{Y}}_{\text{anno}}]_{i,j} \log[\hat{\mathbf{S}}_l]_{i,j} \\ &+ \left(1 - [\hat{\mathbf{Y}}_{\text{anno}}]_{i,j}\right) \log\left(1 - [\hat{\mathbf{S}}_l]_{i,j}\right)\Big),\end{aligned} \tag{31}$$

where $\hat{\mathbf{Y}}_{\text{anno}} \in \mathbf{R}^{(n+1)\times(m+1)}$ is the permutation matrix augmented with an additional row and column to represent keypoints assigned to the dummy node. Here the position $[\cdot]_{n+1,m+1}$ is excluded from calculating the loss. Specifically, the additional row and column are computed as follows:

$$[\hat{\mathbf{Y}}_{\text{anno}}]_{i,m+1} = \left[1 - \sum_j^m [\mathbf{Y}_{\text{anno}}]_{i,j}\right]_+, \tag{32}$$

which equals 1 if keypoint i in $\mathcal{G}_A$ is an outlier. The computation for $[\hat{\mathbf{Y}}_{\text{anno}}]_{n+1,j}$ is analogous. During inference, the soft assignment matrix $\mathbf{S}_l$ is obtained by discarding the dummy node, *i.e.,* $\mathbf{S}_l = [\hat{\mathbf{S}}_l]_{1:n,1:m}$. The Hungarian algorithm is then applied to $\mathbf{S}_l$ to derive the final binary assignment.

D. Momentum Cooperation

As discussed earlier, preprocessing noisy correspondence before training (*i.e.*, obtaining $\mathbf{Y}_{\text{gt}}$ in advance) is infeasible. Inspired by the co-teaching paradigm [71], [72], where two networks with different learning abilities could collaborate to handle different types of noise, we design a robust training strategy that exploits the complementary strengths of KB-QAP and L-QAP. These two networks cooperatively divide and refine correspondences, effectively mitigating noisy correspondence and enhancing matching robustness.

1) Co-divide Noisy Correspondence: We first compute the binary assignment results $\mathbf{Y}_{kb}$ and $\mathbf{Y}_l$ for KB-QAP and L-QAP, respectively, by applying the Hungarian algorithm to the similarity matrices $\mathbf{S}_{kb}$ and $\mathbf{S}_l$. By comparing these predictions with the annotated permutation matrix $\mathbf{Y}_{\text{anno}}$, we categorize the matching results into three types based on logical operations.

Consistent matches refer to assignments that are consistent among KB-QAP, L-QAP, and the annotated permutation matrix, namely,

$$\mathbf{R}_{\text{consistent}} = \left(\mathbf{Y}_{kb} \wedge \mathbf{Y}_l \wedge \mathbf{Y}_{\text{anno}}\right). \tag{33}$$

Partially consistent matches refer to assignments where only one network predicts the annotated correspondence correctly:

$$\mathbf{R}_{\text{partially}} = \left(\neg\mathbf{Y}_{kb} \wedge \mathbf{Y}_l \wedge \mathbf{Y}_{\text{anno}}\right) \vee \left(\mathbf{Y}_{kb} \wedge \neg\mathbf{Y}_l \wedge \mathbf{Y}_{\text{anno}}\right), \tag{34}$$

where $\vee$ represents the logical OR operation.

Incorrect matches are assignments where both $\mathbf{Y}_{kb}$ and $\mathbf{Y}_l$ consistently agree on a prediction that contradicts the annotated permutation matrix:

$$\mathbf{R}_{\text{incorrect}} = \left(\mathbf{Y}_{kb} \wedge \mathbf{Y}_l \wedge \neg\mathbf{Y}_{\text{anno}}\right) \vee \left(\neg\mathbf{Y}_{kb} \wedge \neg\mathbf{Y}_l \wedge \mathbf{Y}_{\text{anno}}\right). \tag{35}$$

The co-divide strategy works under the hypothesis that inconsistencies often signal the presence of noise [71]. Specifically, false positive correspondences can appear in both $\mathbf{R}_{\text{partial}}$ and $\mathbf{R}_{\text{incorrect}}$, while false negative correspondences are typically found in the first term of $\mathbf{R}_{\text{incorrect}}$. Instead of explicitly distinguishing false positive and false negative cases, the noisy correspondences can be addressed in the subsequent refinement step.

2) Co-refine Noisy correspondence: Based on the categorization of matches, we dynamically refine the supervision signals by integrating information from both networks. The refinement process is defined as:

$$\begin{cases} [\mathbf{Y}_{\text{anno}}]_{ij} = 1, & \forall (i,j) \in \mathbf{R}_{\text{consistent}} \\ [\mathbf{Y}_{\text{anno}}]_{ij} = (1-\alpha) + \alpha\mathbf{Y}_{\text{partially}}, & \forall (i,j) \in \mathbf{R}_{\text{partially}} \\ [\mathbf{Y}_{\text{anno}}]_{ij} = \left([\mathbf{S}_{kb}]_{ij} + [\mathbf{S}_l]_{ij}\right)/2 & \forall (i,j) \in \mathbf{R}_{\text{incorrect}}, \end{cases} \tag{36}$$

where α is a hyper-parameter balancing the annotations and the network's predictions.

For consistent matches, the original supervision of "1" is retained, as both networks agree with the annotation. For partially consistent matches, the supervision signal is updated using a weighted combination of the annotated permutation matrix and the prediction from the network whose assignment matches the annotation, *i.e.,*

$$\mathbf{Y}_{\text{partially}} = \left([\mathbf{S}_{kb}]_{ij}[\mathbf{Y}_{kb}]_{ij} + [\mathbf{S}_l]_{ij}[\mathbf{Y}_l]_{ij}\right). \tag{37}$$

For incorrect matches, the supervision signal is updated as the average of the predictions from KB-QAP and L-QAP, leveraging the combined confidence of both networks.

The refined supervision signals serve as updated ground truth in Eq. (32), ensuring that both false positive and false negative noisy correspondences are gradually corrected. Additionally, there are three cases beyond Eqs. (33-35), where at

TABLE I
KEYPOINT MATCHING ACCURACY (%) ON PASCAL VOC WITH STANDARD INTERSECTION FILTERING. OUR METHODS ARE MARKED IN GRAY.

Method	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	Tv	Mean
GMN [24]	41.6	59.6	60.3	48.0	79.2	70.2	67.4	64.9	39.2	61.3	66.9	59.8	61.1	59.8	37.2	78.2	68.0	49.9	84.2	91.4	62.4
PCA [7]	49.8	61.9	65.3	57.2	78.8	75.6	64.7	69.7	41.6	63.4	50.7	67.1	66.7	61.6	44.5	81.2	67.8	59.2	78.5	90.4	64.8
NGM [29]	50.1	63.5	57.9	53.4	79.8	77.1	73.6	68.2	41.1	66.4	40.8	60.3	61.9	63.5	45.6	77.1	69.3	65.5	79.2	88.2	64.1
IPCA [8]	53.8	66.2	67.1	61.2	80.4	75.3	72.6	72.5	44.6	65.2	54.3	67.2	67.9	64.2	47.9	84.4	70.8	64.0	83.8	90.8	67.7
LCS [76]	46.9	58.0	63.6	69.9	87.8	79.8	71.8	60.3	44.8	64.3	79.4	57.5	64.4	57.6	52.4	96.1	62.9	65.8	94.4	92.0	68.5
CIE [10]	52.5	68.6	70.2	57.1	82.1	77.0	70.7	73.1	43.8	69.9	62.4	70.2	70.3	66.4	47.6	85.3	71.7	64.0	83.9	91.7	68.9
QC-DGM [11]	49.6	64.6	67.1	62.4	82.1	79.9	74.8	73.5	43.0	68.4	66.5	67.2	71.4	70.1	48.6	92.4	69.2	70.9	90.9	92.0	70.3
DGMC [25]	50.4	67.6	70.7	70.5	87.2	85.2	82.5	74.3	46.2	69.4	69.9	73.9	73.8	65.4	51.6	98.0	73.2	69.6	94.3	89.6	73.2
BBGM [12]	61.9	71.1	79.7	79.0	87.4	94.0	89.5	80.2	56.8	79.1	64.6	78.9	76.2	75.1	65.2	98.2	77.3	77.0	94.9	93.9	79.0
NGM-v2 [29]	61.8	71.2	77.6	78.8	87.3	93.6	87.7	79.8	55.4	77.8	89.5	78.8	80.1	79.2	62.6	97.7	77.7	75.7	96.7	93.2	80.1
SCGM [61]	62.9	72.9	79.6	79.5	89.3	94.1	89.1	79.2	58.4	79.3	80.5	79.9	79.5	76.8	64.8	98.1	78.0	75.9	98.0	93.2	80.5
ASAR [34]	62.9	74.3	79.5	80.1	89.2	94.0	88.9	78.9	58.8	79.8	88.2	78.9	79.5	77.9	64.9	98.2	77.5	77.1	98.6	93.7	81.1
CREAM [22]	67.0	75.6	82.2	78.1	89.4	91.6	89.3	81.6	62.1	82.3	74.3	81.7	80.9	79.0	67.7	99.3	78.9	73.7	98.3	94.7	81.4
COMMON [17]	65.6	75.2	80.8	79.5	89.3	92.3	90.1	81.8	61.6	80.7	95.0	82.0	81.6	79.5	66.6	98.9	78.9	80.9	99.3	93.8	82.7
COMMON$^+$	68.8	75.5	82.6	77.4	90.0	92.2	89.5	80.7	61.8	82.4	95.3	80.5	82.1	81.6	67.7	98.8	79.9	81.0	98.5	95.4	**83.1**

least one of $\mathbf{Y}_{kb}$ or $\mathbf{Y}_l$ aligns with $\mathbf{Y}_{anno} = 0$. These cases correspond to unalignable assignments and are retained as "0".

3) Momentum-based Enhancement: Deep neural networks often exhibit the memorization effect [73], where they initially learn simple patterns before adapting to more complex ones. In graph matching, precise annotated correspondence can be viewed as simple patterns, while noisy correspondence represents complex ones. Inspired by this phenomenon, we propose a momentum enhancement strategy that learns from high-quality pseudo-targets generated by the momentum model [74], [75].

The momentum model acts as a continuously evolving teacher, retaining simple patterns through an exponential moving average (EMA) of the base model's parameters. Let θ_{q} denote the parameters of the base model and θ_{k} the parameters of the momentum model, the momentum model is updated as:

$$\theta_{\mathrm{k}} \leftarrow t \cdot \theta_{\mathrm{k}} + (1-t) \cdot \theta_{\mathrm{q}}, \tag{38}$$

where t is the momentum coefficient fixed as 0.995.

During training, predictions from the momentum model guide both the co-divide and co-refine processes, providing robust pseudo-supervision. The overall loss function is formulated as,

$$\mathcal{L}_{\mathrm{COMMON^+}} = \mathcal{L}_{\mathrm{KB-QAP}} + \beta \mathcal{L}_{\mathrm{L-QAP}}, \tag{39}$$

where the balance hyper-parameter β is fixed as 0.1 across all experiments.

To further enhance robustness, we maintain a momentum-based outlier threshold from Eq. (15), updated as:

$$\mathrm{thr}_k \leftarrow t \cdot \mathrm{thr}_k + (1-t) \cdot \mathrm{thr}. \tag{40}$$

At inference time, this adaptive threshold is used with the threshold-based Hungarian algorithm [67] to filter outliers. The final matching result $\mathbf{Y}$ is derived by applying the Hungarian algorithm to the averaged soft predictions from KB-QAP and L-QAP, using the computed threshold:

$$\mathbf{Y} = \mathrm{Hungarian}\left(\frac{\mathbf{S}_{kb} + \mathbf{S}_l}{2},\ \mathrm{thr}_k\right). \tag{41}$$

IV. EXPERIMENTS

In this section, we carry out extensive experiments on four widely used visual graph matching datasets with comparisons of 17 state-of-the-art approaches. We also provide social network matching experiments in the supplementary material.

A. Experimental Settings

1) Datasets: We conduct experiments on the following widely used datasets.

- Pascal VOC with Berkeley annotations [14] includes 20 object classes. This dataset contains numerous challenging instances with high variability due to occlusions and pose variations.
- SPair-71K [13] contains 70,958 pairs of object images across various classes, providing a rich set of paired images for evaluating object matching and alignment tasks. SPair-71K offers diverse instances with complex transformations, including scale variations, rotation, and occlusions, making it a comprehensive benchmark for object matching.
- IMC-PT-GM [19] consists of images depicting 16 tourists worldwide. Unlike benchmarks that focus on keypoint matching of single objects, IMC-PT-GM involves matching larger scenes with the largest number of keypoints (averaging 44.48), and the highest partial correspondence rate (55.5%). This dataset bridges closer to real-world tasks such as structure-from-motion [5].
- Willow Object [77] comprises five object categories, each annotated with 10 distinctive landmarks by human experts. As the annotations in the dataset are relatively precise, we primarily use this dataset to evaluate the impact of synthetic partial and noisy correspondences, providing a robust test of the models.

2) Compared Methods: We compare our method against 14 widely recognized deep graph matching methods: GMN [24], PCA [7], IPCA [8], NGM [29], DGMC [25], QC-DGM [11], CIE [10], LCS [76], BBGM [12], NGM-v2 [29], SCGM [61], ASAR [34], CREAM [22], GMTR [30], and COMMON [17].

TABLE II
KEYPOINT MATCHING ACCURACY (%) ON SPAIR-71K.

Method	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dog	Horse	Mbike	Person	Plant	Sheep	Train	Tv	Mean
GMN [24]	59.9	51.0	74.3	46.7	63.3	75.5	69.5	64.6	57.5	73.0	58.7	59.1	63.2	51.2	86.9	57.9	70.0	92.4	65.3
PCA [7]	64.7	45.7	78.1	51.3	63.8	72.7	61.2	62.8	62.6	68.2	59.1	61.2	64.9	57.7	87.4	60.4	72.5	92.8	66.0
NGM [29]	66.4	52.6	77.0	49.6	67.7	78.8	67.6	68.3	59.2	73.6	63.9	60.7	70.7	60.9	87.5	63.9	79.8	91.5	68.9
IPCA [8]	69.0	52.9	80.4	54.3	66.5	80.0	68.5	71.4	61.4	74.8	66.3	65.1	69.6	63.9	91.1	65.4	82.9	97.5	71.2
CIE [10]	71.5	57.1	81.7	56.7	67.9	82.5	73.4	74.5	62.6	78.0	68.7	66.3	73.7	66.0	92.5	67.2	82.3	97.5	73.3
NGM-v2 [29]	68.8	63.3	86.8	70.1	69.7	94.7	87.4	77.4	72.1	80.7	74.3	72.5	79.5	73.4	98.9	81.2	94.3	98.7	80.2
BBGM [12]	75.3	65.0	87.6	78.0	69.8	94.0	87.8	78.3	72.8	82.7	76.6	76.3	80.1	75.0	98.7	85.2	96.3	98.0	82.1
ASAR [34]	72.4	61.8	91.8	79.1	71.2	97.4	90.4	78.3	74.2	83.1	77.3	77.0	83.1	76.4	99.5	85.2	97.8	99.5	83.1
CREAM [22]	78.4	70.3	90.5	78.6	72.1	98.5	91.7	82.0	71.4	87.1	82.4	75.4	83.5	84.4	99.4	86.0	99.5	99.9	85.1
COMMON [17]	77.3	68.2	92.0	79.5	70.4	97.5	91.6	82.5	72.2	88.0	80.0	74.1	83.4	82.8	99.9	84.4	98.2	99.8	84.5
COMMON$^+$	79.8	72.3	91.7	78.7	70.8	98.0	91.8	81.9	72.8	88.2	83.3	76.4	83.4	83.9	99.9	86.1	99.2	99.9	**85.5**

TABLE III
KEYPOINT MATCHING ACCURACY (%) ON WILLOW OBJECT.

Method	Car	Duck	Face	Mbike	Wbottle	Mean
GMN [24]	67.9	76.7	99.8	69.2	83.1	79.3
NGM [29]	84.2	77.6	99.4	76.8	88.3	85.3
PCA [7]	87.6	83.6	100	77.6	88.4	87.4
CIE [10]	85.8	82.1	99.9	88.4	88.7	89.0
IPCA [8]	90.4	88.6	100	83.0	88.3	90.1
SCGM [61]	91.3	73.0	100	95.6	96.6	91.3
ASAR [34]	92.5	84.0	100	95.4	99.0	94.2
LCS [76]	91.2	86.2	100	99.4	97.9	94.9
DGMC [25]	98.3	90.2	100	98.5	98.1	97.0
BBGM [12]	96.8	89.9	100	99.8	99.4	97.2
NGM-v2 [29]	97.4	93.4	100	98.6	98.3	97.5
QC-DGM [11]	98.0	92.8	100	98.8	99.0	97.7
CREAM [22]	97.7	96.3	100	100	99.8	98.8
COMMON [17]	97.6	98.2	100	100	99.6	99.1
COMMON$^+$	98.3	98.2	100	100	100	**99.3**

TABLE IV
KEYPOINT MATCHING ACCURACY (%) WITH VISION TRANSFORMER

Method	Pascal	SPair-71k	Willow	Mean
BBGM [12]	83.6	83.0	98.2	88.3
GMTR [30]	84.5	83.9	98.2	88.9
CREAM [22]	85.6	85.7	99.1	90.1
COMMON [17]	85.6	85.5	99.1	90.1
COMMON$^+$	85.9	86.0	99.5	**90.5**

Among them, CREAM and COMMON (our conference version) are the only methods designed for graph matching with noisy correspondence. Moreover, we include three methods specifically designed for partial matching: ZACR [37], GCAN [18], and AFAT [19].

3) Metrics: Following existing methods [18], [19], we adopt matching accuracy as the metric for the full matching task and F1-score for the partial matching task. For the full matching task (without outliers), matching accuracy is defined as: $\frac{\mathrm{tr}(\mathbf{Y}^\top \mathbf{Y}_{\mathrm{gt}})}{\mathrm{sum}(\mathbf{Y}_{\mathrm{gt}})}$. This metric measures the fraction of correctly predicted matches relative to the total number of ground-truth matches. For the partial matching task (with outliers), F1-score balances the correctness and completeness of the predicted correspondences, calculated as $\frac{2\cdot(\mathrm{precision}\cdot\mathrm{recall})}{\mathrm{precision}+\mathrm{recall}}$ where $\mathrm{precision} = \frac{\mathrm{tr}\left(\mathbf{Y}^\top \mathbf{Y}_{\mathrm{anno}}\right)}{\mathrm{sum}(\mathbf{Y})}$ and $\mathrm{recall} = \frac{\mathrm{tr}\left(\mathbf{Y}^\top \mathbf{Y}_{\mathrm{gt}}\right)}{\mathrm{sum}(\mathbf{Y}_{\mathrm{gt}})}$. For comprehensive evaluations, we report both average and per-category performance.

4) Implementation Details: All datasets and methods are preprocessed and evaluated using the ThinkMatch [78] framework[1], ensuring consistency, reproducibility, and fair comparisons. Our method is implemented in PyTorch 1.10.0 and all experiments are conducted on Ubuntu 20.04 with an NVIDIA 3090 GPU. To optimize the networks, we use the Adam optimizer [79] with default parameters. The initial learning rate is set to $3e^{-4}$ for the graph networks and $2e^{-5}$ for fine-tuning the VGG network. The batch size is set to 8 image pairs, and the softmax temperature τ for contrastive learning is fixed at 0.07. The network is warmed up for 1 epoch before applying the momentum cooperation strategy, with the temperature parameter α fixed at 0.4.

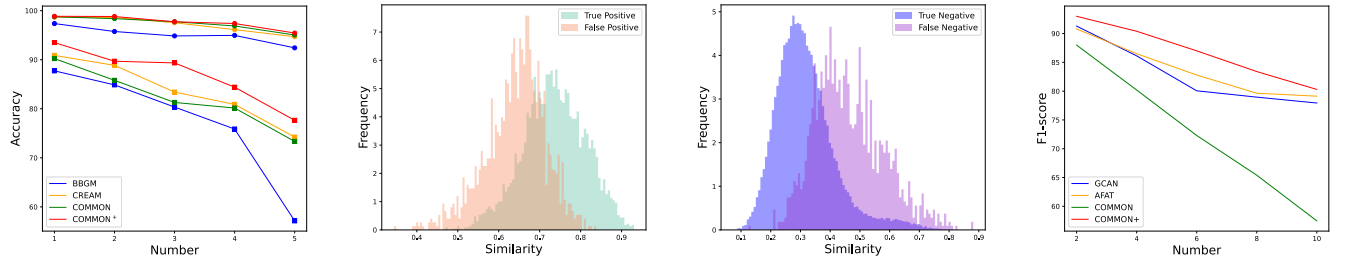
[1]https://github.com/Thinklab-SJTU/ThinkMatch

B. Evaluation on Noisy Correspondence

In this section, we evaluate the effectiveness of our method in addressing noisy correspondence through experiments on datasets with real-world noise, synthetic noise, and varying viewpoints.

1) Matching Results with Real-world Noise: To evaluate the ability of methods to specifically address noisy correspondence, we exclude outliers before performing graph matching following the protocol from BBGM [12]. As shown in Table I, our method achieves state-of-the-art performance on Pascal VOC, surpassing the most competitive baseline method CREAM by +1.7% and outperforming COMMON by 0.4% in terms of the mean matching accuracy. Notably, significant improvements are observed in object categories heavily affected by noisy correspondence, such as table (+7.1%) and sofa (+3.9%), highlighting the robustness of our approach. Similarly, on SPair-71k (Table II), our method consistently outperforms all the baselines. Furthermore, as shown in Table III, our method demonstrates its effectiveness on the Willow Object dataset with relatively small-scale training data.

Given the growing prominence of Vision Transformers (ViT) in visual tasks, we further evaluate the compatibility of existing methods by replacing the VGG16 backbone with ViT-B/16 pretrained on ImageNet. We also compare our method



(a) Performance under noisy correspondence (b) Similarity distributions of TP and FP (c) Similarity distributions of TN and FN (d) Performance under partial correspondence

Fig. 4. Effectiveness of addressing noisy and partial correspondence. In (a), the line with dot markers represents the performance under varying levels of false positive correspondence during training, while the line with square markers includes additional false negative correspondences.

TABLE V
PERCENTAGE (%) OF KB-QAP AND L-QAP EXPERTS IN IDENTIFYING NOISY CORRESPONDENCE. "CONSISTENT" REFERS TO CASES WHERE BOTH EXPERTS IDENTIFY NOISY CORRESPONDENCE, WHILE "PARTIALLY CONSISTENT" INDICATES CASES WHERE ONLY ONE EXPERT IDENTIFIES IT. "ALL" REPRESENTS THE TOTAL COMBINATION OF BOTH TYPES.

Noise	KB-QAP Expert	L-QAP Expert	Consistent	Partially Consistent	All
FP	42.2	42.6	40.4	4.1	44.4
FN	55.4	58.1	54.6	5.3	59.9

TABLE VI
KEYPOINT MATCHING ACCURACY (%) ON SPAIR-71K GROUPED BY LEVELS OF DIFFICULTY IN THE VIEWPOINT OF THE IMAGE PAIRS.

Method	Viewpoint Difficulty			All
	Easy	Medium	Hard	
BBGM [12]	84.7	78.9	73.6	82.1
ASAR [34]	86.5	79.1	72.5	83.1
CREAM [22]	87.8	82.2	76.2	85.1
COMMON [17]	86.6	81.4	76.4	84.5
COMMON$^+$	88.1	82.6	76.4	**85.5**

against GMTR [30], a state-of-the-art ViT-based graph matching method. As shown in Table IV, our method achieves the best performance across all datasets. Specifically, COMMON$^+$ surpasses GMTR by a notable margin of $+1.6\%$ in average accuracy. Moreover, both CREAM and COMMON outperform GMTR, underscoring the importance of addressing noisy correspondence in graph matching. By evaluating our method across diverse benchmarks and feature extractors, these results consistently demonstrate the robustness and adaptability of our approach.

2) Synthetic Noise Experiment: To explicitly evaluate the robustness of our method, we conduct experiments on the Willow Object dataset with synthetic noisy correspondence. In this experiment, both false positive and false negative correspondences are introduced to simulate noisy conditions. Specifically, we simulate false positive correspondences by randomly selecting keypoints in the training set and adding displacement to their locations. The displacement (s, θ) is sampled from a uniform distribution: $s \sim \mathcal{U}(0.1, 0.2)$ and $\theta \sim \mathcal{U}(0, 360)$, where s is the magnitude of displacement, and θ is the angle. The displacement value is scaled relative to the bounding box size. Correspondences derived from these displaced keypoints are treated as false positives. To simulate false negatives, we randomly select two keypoints in one image and flip their labels.

Varying noise ratios. As shown in the lines with dot markers in Fig. 4(a), we vary the number of displaced keypoints from 1 to 5 to evaluate false positive correspondences. Across all noise levels, our method consistently outperforms baseline methods. The lines with square markers in Fig. 4(a) present results with two additional false negative correspondences (one label flip). Similarly, our method achieves the best performance, validating its effectiveness in addressing both types of noisy correspondence, even under challenging conditions with high noise levels.

Distribution of similarity scores. We analyze the similarity distributions of the keypoints derived from our method when training with both false positive and false negative correspondence. In Fig. 4(b), we present the histograms of similarity scores for true positive correspondences (keypoint pairs without synthetic noise) and false positive correspondences. As shown, false positive correspondences exhibit an average similarity score of 0.65, whereas true positive correspondences achieve a higher average similarity of 0.75, reflecting a clear separation between the two categories. Similarly, in Fig. 4(c), we display histograms of similarity scores for true negative and false negative correspondences. The peaks of the distributions for the two types of correspondence are distinct, further confirming our model's capacity to differentiate between them. Overall, these findings demonstrate that our method could alleviate the negative impact of noisy correspondence during optimization.

Complementarity of KB-QAP and L-QAP experts. We analyze the complementary characteristics of the KB-QAP and L-QAP experts, specifically focusing on their ability to identify false positive (FP) and false negative (FN) correspondences. To assess their performance, we add 2 FP and 2 FN correspondences to the dataset. As shown in Table V, the L-QAP expert demonstrates better performance in identifying noisy correspondence, as it further fuses the graph after the alignment of the KB-QAP expert. Notably, around 4% and 5% of the predictions from both experts are inconsistent for FP or FN, respectively. By considering both "consistent" and "partially consistent" matches, we achieve improved discrimination of noisy correspondence, as shown in the last column. This complementary behavior allows the

TABLE VII
KEYPOINT MATCHING F1-SCORE (%) ON WILLOW OBJECT (+RANDOM OUTLIERS) AND IMC-PT-GM (50/100 ANCHORS). PCH MEANS PARTIAL CORRESPONDENCE HANDLING STRATEGY.

Dataset Name		Willow Object						IMC-PT-GM (50 anchors)				IMC-PT-GM (100 anchors)			
GM Network	PCH	Car	Duck	Face	Mbike	Bottle	Mean	Reichstag	Sacre	St_peters	Mean	Reichstag	Sacre	St_peters	Mean
ZACR [37]	ZACR [37]	47.3	44.7	77.7	39.9	53.6	52.6	72.1	33.7	29.5	45.1	39.4	33.1	30.4	34.3
PCA [7]	None	55.8	56.5	81.2	46.4	58.1	59.6	83.4	47.5	58.5	63.1	70.7	43.1	58.8	57.5
BBGM [12]	LPMP [80]	65.1	60.7	85.5	71.6	65.5	69.7	85.4	55.1	59.3	66.6	88.1	55.0	56.4	66.5
NGM-v2 [29]	None	78.9	66.6	84.3	63.1	76.0	73.8	90.8	55.9	64.3	70.3	78.4	54.9	69.3	67.6
	Threshold	86.8	74.5	91.2	71.0	83.8	81.4	91.4	56.8	65.8	71.3	80.3	56.9	71.6	69.6
	Dummy node	83.3	69.7	95.7	68.8	86.7	80.8	88.5	56.1	63.0	69.2	80.0	57.0	71.3	69.5
	AFAT-U [19]	82.6	74.5	90.6	73.9	87.0	81.7	90.5	58.7	66.9	72.0	81.7	57.0	72.2	70.3
	AFAT-I [19]	84.6	75.7	92.0	74.5	88.6	83.1	92.3	58.7	66.7	72.8	82.0	57.0	71.4	70.1
GCAN [18]	Dummy node	74.8	75.7	92.8	77.1	83.5	80.8	87.2	55.1	63.0	68.4	80.4	55.7	72.8	69.6
	AFAT-U [19]	80.1	78.0	90.6	76.0	87.0	82.3	86.9	59.4	67.1	71.1	82.6	58.2	73.8	71.5
	AFAT-I [19]	82.2	77.7	92.7	77.2	88.6	83.7	91.0	60.3	67.3	72.9	82.7	57.8	72.4	70.9
COMMON [17]	None	73.1	68.4	82.4	61.0	79.4	72.8	89.7	55.8	65.2	70.2	78.8	54.1	68.3	67.1
COMMON [17]	Threshold	80.7	74.3	90.3	71.2	84.1	80.1	89.9	57.2	65.0	70.7	79.1	56.3	69.5	68.3
COMMON$^+$	Threshold	85.7	78.7	95.2	75.7	90.4	**85.1**	91.4	59.4	70.0	**73.6**	83.1	60.0	75.5	**72.9**

TABLE VIII
KEYPOINT MATCHING F1-SCORE (%) ON PASCAL VOC IN PARTIAL MATCHING SETTING.

GM Network	PCH	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	Tv	Mean
ZACR [37]	ZACR [37]	12.2	31.8	31.7	23.0	35.0	28.3	21.8	32.6	19.6	23.8	33.8	29.9	28.8	21.4	10.8	39.0	26.9	15.5	55.8	82.5	30.2
PCA [7]	None	35.6	60.3	43.7	34.5	81.5	54.9	30.1	47.8	30.4	46.4	43.9	44.5	46.1	52.4	29.4	78.7	40.7	30.4	58.6	81.2	48.6
BBGM [12]	LPMP [80]	42.2	66.7	54.9	46.1	85.7	66.5	39.8	60.3	38.9	65.1	60.1	58.4	58.1	62.4	41.3	96.1	53.5	26.3	75.9	82.6	59.0
NGM-v2 [29]	None	45.5	65.3	55.3	45.8	88.4	64.3	45.9	58.6	43.3	59.1	39.2	55.7	58.0	65.3	44.4	95.4	50.3	41.2	72.4	81.8	58.8
	Threshold	48.3	65.4	55.3	48.6	87.6	63.0	51.1	61.1	39.6	63.3	33.6	59.2	59.3	63.4	46.9	95.2	53.5	45.5	73.4	81.4	59.4
	Dummy node	44.7	61.9	57.1	41.9	83.9	63.9	54.1	60.8	40.5	64.2	36.2	60.6	60.8	61.9	48.7	91.2	56.2	37.4	63.2	82.2	58.6
	AFAT-U [19]	45.7	67.7	57.3	44.9	90.1	65.5	49.9	59.3	44.0	62.0	54.9	58.4	58.6	63.8	45.9	94.8	50.9	37.3	74.2	82.8	60.2
	AFAT-I [19]	45.0	67.3	55.9	45.6	90.3	64.6	48.7	58.0	44.7	60.2	54.8	57.2	57.5	63.4	45.2	95.3	49.3	41.6	73.6	82.4	59.9
GCAN [18]	Dummy node	46.3	67.7	57.4	45.0	87.1	64.8	57.5	61.2	40.8	61.6	37.3	59.9	59.2	64.6	49.7	95.1	54.5	28.5	77.9	83.1	59.7
	AFAT-U [19]	47.1	70.8	58.1	45.8	90.8	66.5	49.6	58.8	50.6	64.6	47.2	60.5	62.3	65.7	46.3	95.4	52.7	47.4	74.2	83.8	62.0
	AFAT-I [19]	46.1	69.9	56.1	46.6	90.7	66.1	48.1	57.9	49.9	63.9	50.4	59.0	61.6	65.0	44.7	95.5	50.9	49.2	74.0	83.8	61.6
COMMON [17]	None	48.8	70.6	58.9	49.1	89.1	63.3	47.8	61.5	45.5	62.5	34.2	60.1	59.1	69.9	47.8	96.4	50.6	40.3	77.6	84.7	60.9
COMMON [17]	Threshold	47.6	72.3	58.5	48.9	89.2	63.6	53.0	60.7	49.0	63.9	43.7	60.0	56.1	70.5	50.2	96.6	49.4	37.5	79.0	83.4	61.7
COMMON$^+$	Threshold	47.5	71.4	57.9	49.9	86.9	61.6	58.3	64.7	50.7	66.3	50.8	61.1	56.8	69.9	52.2	97.0	57.0	39.6	78.5	83.2	**63.1**

two experts to collaborate and refine noisy correspondences, thereby enhancing the robustness of graph matching.

3) Evaluation under Different Viewpoint Difficulty: SPair-71k [13] categorizes image pairs into easy, medium, and hard levels based on the extent of viewpoint variation. Notably, image pairs with higher viewpoint difficulty often contain more noisy correspondences such as occlusion [17], posing significant challenges to graph matching. As presented in Table VI, our method consistently outperforms baseline approaches across all difficulty levels. For hard pairs, which are most probably affected by noisy correspondences, our method achieves a substantial improvement of +2.8% compared to the traditional method BBGM [12]. Furthermore, our approach outperforms the noisy correspondence learning method COMMON [17], with gains of +1.5% on easy pairs and +1.2% on medium pairs. These results demonstrate that our method not only excels in handling noisy correspondence in challenging conditions but also maintains strong performance across varying levels of viewpoint difficulty.

C. Evaluation on Partial Correspondence

In this section, we evaluate the effectiveness of our method in handling partial correspondence through experiments on real-world datasets with both sparse and dense keypoints, as well as scenarios with varying numbers of outliers.

1) Matching Results on Datasets with Real-world Outliers: We evaluate the performance of various methods on the Pascal VOC, IMC-PT-GM, and Willow Object datasets. Among these, the IMC-PT-GM dataset poses significant challenges due to its high density of keypoints and an outlier rate of 55%, which reflects real-world architectural matching scenarios such as landmarks like the "Basilique du Sacré Coeur". As shown in the middle and right parts of Table VII, our method achieves superior results using a simple threshold-based Hungarian approach, surpassing the most comparable baseline GCAN [18] combined with the outlier-handling method AFAT [19]. Additionally, applying the threshold-based Hungarian algorithm to COMMON results in only marginal improvements, further emphasizing the advantages of our method.

The Pascal VOC dataset offers additional validation of

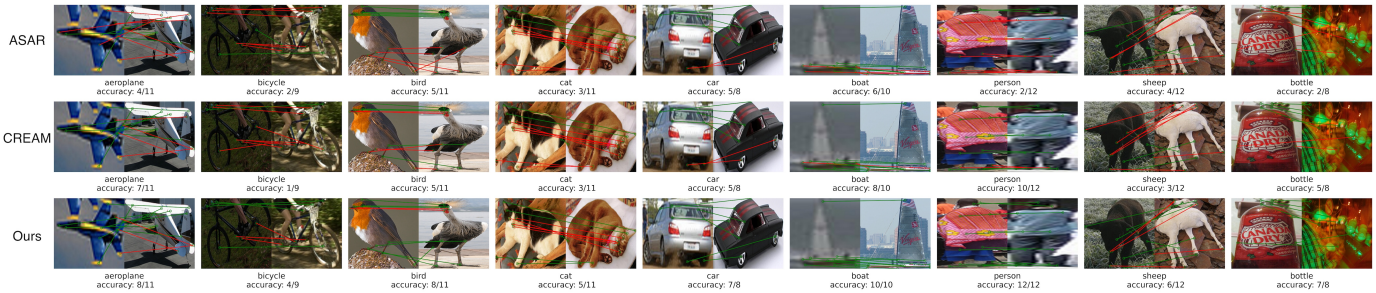


Fig. 5. Visualization of matching results on Pascal VOC and SPair-71k without outliers. Circles indicate annotated keypoints, while green and red lines represent correct and false matching results, respectively.

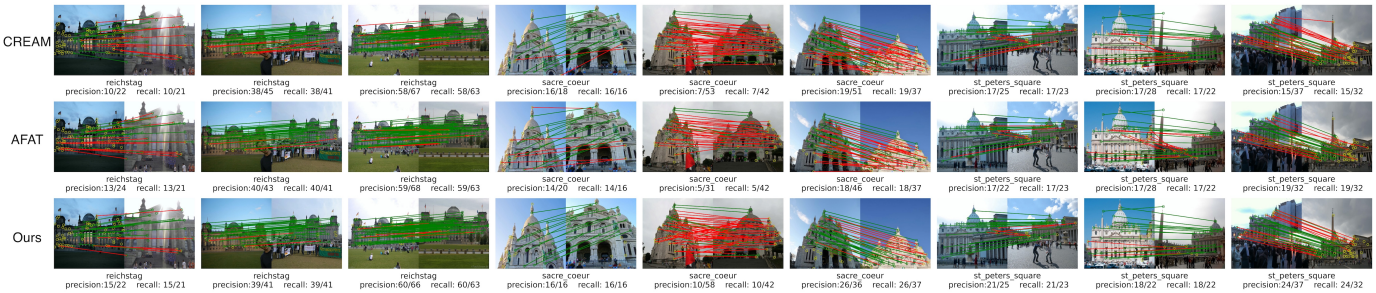


Fig. 6. Visualization of matching results on IMC-PT-GM. Green and yellow circles denote annotated inliers and outliers, respectively.

TABLE IX
ABLATION STUDY IN TERMS OF ACCURACY (FULL MATCHING) AND F1-SCORE (PARTIAL MATCHING).

	Method	Full Matching SPair-71k	Full Matching Pascal	Partial Matching IMC-50	Partial Matching IMC-100	Partial Matching Pascal
1	**COMMON$^+$**	**85.45**	**83.12**	**73.59**	72.88	**63.06**
2	– Result from KB-QAP head	85.14	82.84	71.80	69.03	62.84
3	– Result from L-QAP head	84.59	82.35	72.97	**72.92**	63.00
4	– w/o Momentum Cooperation	85.08	82.30	72.75	72.44	61.99
5	– w/o Quadratic Contrast	80.44	75.87	68.92	68.94	56.68
6	KB-QAP form	84.54	82.67	70.70	68.28	61.66
7	– w/o Graph Consistency	83.83	81.95	69.63	66.16	60.73
8	– w Regularized InfoNCE [57]	83.66	81.67	69.25	64.73	60.44
9	– w InfoNCE [81]	83.38	81.33	69.21	65.03	60.51
10	L-QAP form	79.75	76.21	69.28	69.52	57.08
11	– w/o Dummy Node	–	–	68.90	68.47	56.73

our approach. As demonstrated in Table VIII, our method consistently outperforms other approaches. Moreover, on the Willow Object dataset, we simulate challenging conditions by randomly adding 1 to 10 background outliers to each image following the protocol in [19]. The results in the left part of Table VII confirm that our method consistently outperforms existing partial matching methods. These results highlight the adaptability and effectiveness of our method in addressing both sparse and dense keypoint scenarios.

2) Varying Outlier Numbers: To explicitly evaluate the robustness of our method, we further conduct experiments on the Willow Object dataset with synthetic outliers. For each image, 2 to 10 background outliers are randomly added with an interval of 2. As illustrated in Fig. 4(d), our method consistently outperforms the partial matching baselines GCAN [18] and AFAT [19] across all noise levels. Notably, AFAT, which is post-trained on the GCAN model, shows improved performance but still lags behind our approach. These results clearly demonstrate the robustness of our method.

D. Ablation Studies

We perform ablation studies to analyze the contributions of each component in our framework under both full matching and partial matching scenarios. For full matching, outliers are manually excluded before matching and the dummy node mechanism is therefore not applied. As shown in Table IX, the full COMMON$^+$ framework achieves the best overall performance by combining the KB-QAP head, the L-QAP head, and the proposed training strategies (Line 1). From the ablation results, we make the following observations:

Complementary nature of KB-QAP and L-QAP. When using a single prediction head, KB-QAP is more favorable in full-matching scenarios, whereas L-QAP is more robust in partial matching scenarios with many outliers (Lines 2 vs. 3), indicating their complementary strengths. Importantly, integrating the predictions of both heads yields the best overall results (Lines 1–3).

Mutual enhancement of the two objectives. Beyond prediction fusion, the two QAP objectives also provide complementary supervision. Adding the L-QAP objective improves the KB-QAP head prediction (Lines 2 vs. 6), suggesting that fusion-oriented supervision provides additional cues beneficial to alignment. Symmetrically, adding the KB-QAP objective improves the L-QAP head prediction (Lines 3 vs. 10), indicating that alignment-oriented supervision helps refine fusion-based matching. Moreover, momentum cooperation further

improves performance (Lines 1 vs. 4), supporting the effectiveness of our align-fuse-refine design.

Effect of quadratic contrastive learning and graph consistency. Removing the quadratic contrastive loss leads to a clear performance drop (Lines 5 vs. 1), and the model behavior becomes much closer to training on the L-QAP form alone, highlighting the importance of enforcing graph alignment before fusion. In addition, graph consistency regularization improves performance (Lines 6 vs. 7), underscoring the role of structure-aware geometric constraints in graph matching. Finally, our quadratic contrastive objective consistently outperforms alternative contrastive losses (Lines 6, 8, 9).

E. Visualization of Matching Results

In this section, we present visualizations of the matching results in both full and partial matching scenarios. Fig. 5 showcases the results of full matching on the Pascal VOC and SPair-71k datasets. As shown, our method consistently outperforms comparable baselines ASAR [34] and CREAM [22], which are designed for handling adversarial noise and noisy correspondence problems, respectively. In particular, our approach excels in challenging scenarios such as images with blurring or low recognizability (*e.g.*, boat and bottle cases) and those with large viewpoint variations (*e.g.*, car and sheep cases). These findings validate the robustness of our method in complex visual matching tasks.

Fig. 6 compares our method against the partial matching baseline AFAT [19] and the SOTA graph matching baseline CREAM [22] on the IMC-PT-GM dataset. For a fair comparison, we apply threshold-based Hungarian filtering to CREAM and manually adjust the threshold. As shown, AFAT surpasses CREAM in estimating matchable keypoints due to its explicit modeling of the number of matchable keypoints. In contrast, our method introduces a simple learnable threshold for outlier filtering, achieving superior results particularly in scenarios with large-scale variations (*e.g.*, the 7th and 9th columns). The results of this challenging dataset further emphasize the effectiveness and adaptability of our approach.

V. Conclusion

This paper addresses the intertwined challenges of noisy and partial correspondence in graph matching. To address this issue, we propose a unified method that seamlessly integrates Koopmans-Beckmann's Quadratic Assignment Programming and Lawler's Quadratic Assignment Programming for graph alignment and fusion. By leveraging the complementary strengths of these formulations, our method effectively mitigates the impact of noisy and partial correspondence through momentum cooperation. Our approach has been extensively validated across both full and partial matching settings on diverse real-world datasets with varying keypoint densities, demonstrating its robustness and effectiveness. This work may inspire some further exploration of the partial and noisy correspondence problem.

Author Contributions

All authors contributed significantly to this work. Xi Peng conceived the study, designed the COMMON algorithm, and supervised the project. Yijie Lin co-designed and implemented the algorithm, conducted the experiments, and drafted the manuscript. Mouxing Yang and Peng Hu analyzed the experimental results and contributed to the formulation of the manuscript. Jiancheng Lv and Hao Chen provided technical discussions and constructive feedback on the manuscript. All authors reviewed and approved the final version.

References

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

[2] Y. Yang, Z. Ren, H. Li, C. Zhou, X. Wang, and G. Hua, "Learning dynamics via graph neural networks for human pose estimation and tracking," in *Proc. Comput. Vis. Pattern Recognit.*, 2021, pp. 8074–8084.

[3] N. Ufer and B. Ommer, "Deep semantic feature matching," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 6914–6923.

[4] L. Chen, Z. Gan, Y. Cheng, L. Li, L. Carin, and J. Liu, "Graph optimal transport for cross-domain alignment," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1542–1553.

[5] P. Lindenberger, P.-E. Sarlin, V. Larsson, and M. Pollefeys, "Pixel-perfect structure-from-motion with featuremetric refinement," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 5987–5997.

[6] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.

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

[8] ——, "Combinatorial learning of robust deep graph matching: an embedding based approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

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

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

[11] Q. Gao, F. Wang, N. Xue, J.-G. Yu, and G.-S. Xia, "Deep graph matching under quadratic constraint," in *Proc. Comput. Vis. Pattern Recognit.*, 2021, pp. 5069–5078.

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

[13] J. Min, J. Lee, J. Ponce, and M. Cho, "Spair-71k: A large-scale benchmark for semantic correspondence," *arXiv:1908.10543*, 2019.

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

[15] R. Wang, J. Yan, and X. Yang, "Unsupervised learning of graph matching with mixture of modes via discrepancy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10 500–10 518, 2023.

[16] 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," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 517–547, 2021.

[17] Y. Lin, M. Yang, J. Yu, P. Hu, C. Zhang, and X. Peng, "Graph matching with bi-level noisy correspondence," in *Proc. Int. Conf. Comput. Vis.*, 2023, pp. 23 362–23 371.

[18] Z. Jiang, H. Rahmani, P. Angelov, S. Black, and B. M. Williams, "Graph-context attention networks for size-varied deep graph matching," in *Proc. Comput. Vis. Pattern Recognit.*, 2022, pp. 2343–2352.

[19] R. Wang, Z. Guo, S. Jiang, X. Yang, and J. Yan, "Deep learning of partial graph matching via differentiable top-k," in *Proc. Comput. Vis. Pattern Recognit.*, 2023, pp. 6272–6281.

[20] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision," 2003.

[21] K.-H. Lee, X. He, L. Zhang, and L. Yang, "Cleannet: Transfer learning for scalable image classifier training with label noise," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 5447–5456.

[22] X. Ma, M. Yang, Y. Li, P. Hu, J. Lv, and X. Peng, "Cross-modal retrieval with noisy correspondence via consistency refining and mining," *IEEE Trans. Image Process.*, vol. 33, pp. 2587–2598, 2024.

[23] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

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

[25] M. Fey, J. E. Lenssen, C. Morris, J. Masci, and N. M. Kriege, "Deep graph matching consensus," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[26] H. Liu, T. Wang, C. Lang, Y. Li, and H. Ling, "Deep probabilistic graph matching," *IEEE Trans. Knowl. Data Eng.*, 2025.

[27] 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.

[28] B. Jiang, P. Sun, Z. Zhang, J. Tang, and B. Luo, "Gamnet: Robust feature matching via graph adversarial-matching network," in *Proc. ACM Int. Conf. Multimedia*, 2021, pp. 5419–5426.

[29] 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5261–5279, 2022.

[30] J. Guo, S. Zhang, R. Wang, C. Liu, and J. Yan, "Gmtr: Graph matching transformers," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2024, pp. 6535–6539.

[31] L. Liu, M. C. Hughes, S. Hassoun, and L. Liu, "Stochastic iterative graph matching," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6815–6825.

[32] S. Tourani, C. Rother, M. H. Khan, and B. Savchynskyy, "Unsupervised deep graph matching based on cycle consistency," *arXiv preprint arXiv:2307.08930*, 2023.

[33] M. Nguyen, D. Nguyen, N. Diep, T. N. Pham, T. Cao, B. Nguyen, P. Swoboda, N. Ho, S. Albarqouni, P. Xie *et al.*, "Lvm-med: Learning large-scale self-supervised vision models for medical imaging via second-order graph matching," *Proc. Neural Inf. Process. Syst.*, vol. 36, 2024.

[34] Q. Ren, Q. Bao, R. Wang, and J. Yan, "Appearance and structure aware robust deep visual graph matching: attack, defense and beyond," in *Proc. Comput. Vis. Pattern Recognit.*, 2022, pp. 15 263–15 272.

[35] H. Shao, L. Wang, Y. Wang, Q. Ren, and J. Yan, "Certified robustness on visual graph matching via searching optimal smoothing range," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2024, pp. 2596–2607.

[36] J. Qu, H. Ling, C. Zhang, X. Lyu, and Z. Tang, "Adaptive edge attention for graph matching with outliers," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 966–972.

[37] F. Wang, N. Xue, J.-G. Yu, and G.-S. Xia, "Zero-assignment constraint for graph matching with outliers," in *Proc. Comput. Vis. Pattern Recognit.*, 2020, pp. 3033–3042.

[38] Y. Bai, D. Xu, Y. Sun, and W. Wang, "Glsearch: Maximum common subgraph detection via learning to search," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 588–598.

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

[40] X. Yang, H. Qiao, and Z.-Y. Liu, "Outlier robust point correspondence based on gnccp," *Pattern Recognit. Lett.*, vol. 55, pp. 8–14, 2015.

[41] J. Yan, M. Cho, H. Zha, X. Yang, and S. M. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, 2015.

[42] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning," in *Proc. Comput. Vis. Pattern Recognit.*, 2021, pp. 16 684–16 693.

[43] Y. Li, M. Yang, D. Peng, T. Li, J. Huang, and X. Peng, "Twin contrastive learning for online clustering," *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2205–2221, 2022.

[44] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," in *Proc. Empir. Methods Nat. Lang. Process.*, 2021.

[45] P. O. O Pinheiro, A. Almahairi, R. Benmalek, F. Golemo, and A. C. Courville, "Unsupervised learning of dense visual representations," in *Proc. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4489–4500.

[46] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "Completer: Incomplete multi-view clustering via contrastive prediction," in *Proc. Comput. Vis. Pattern Recognit.*, 2021, pp. 11 174–11 183.

[47] S. Goel, H. Bansal, S. Bhatia, R. Rossi, V. Vinay, and A. Grover, "Cyclip: Cyclic contrastive language-image pretraining," *Proc. Neural Inf. Process. Syst.*, vol. 35, pp. 6704–6719, 2022.

[48] Y. Lin, Y. Gou, X. Liu, J. Bai, J. Lv, and X. Peng, "Dual contrastive prediction for incomplete multi-view representation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4513–4524, 2022.

[49] Y. Lin, J. Zhang, Z. Huang, J. Liu, and X. Peng, "Multi-granularity correspondence learning from long-term noisy videos," in *Proc. Int. Conf. Learn. Represent.*, 2024.

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

[51] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[52] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 7303–7313.

[53] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," in *Proc. Comput. Vis. Pattern Recognit.*, 2021, pp. 3024–3033.

[54] A. Jabri, A. Owens, and A. Efros, "Space-time correspondence as a contrastive random walk," *Proc. Neural Inf. Process. Syst.*, vol. 33, pp. 19 545–19 560, 2020.

[55] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Proc. Neural Inf. Process. Syst.*, vol. 33, pp. 5812–5823, 2020.

[56] Y. Zhang, H. Zhu, Z. Song, P. Koniusz, and I. King, "Costa: Covariance-preserving feature augmentation for graph contrastive learning," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2022, pp. 2524–2534.

[57] A. Moskalev, I. Sosnovik, V. Fischer, and A. Smeulders, "Contrasting quadratic assignments for set-based representation learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 88–104.

[58] J. Zhang and P. S. Yu, "Integrated anchor and social link predictions across social networks," in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 2125–2131.

[59] S. Zhang, H. Tong, J. Tang, J. Xu, and W. Fan, "Incomplete network alignment: Problem definitions and fast solutions," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 14, no. 4, pp. 1–26, 2020.

[60] E. Zhong, W. Fan, J. Wang, L. Xiao, and Y. Li, "Comsoc: adaptive transfer of user behaviors over composite social network," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 696–704.

[61] C. Liu, S. Zhang, X. Yang, and J. Yan, "Self-supervised learning of visual graph matching," in *Proc. Eur. Conf. Comput. Vis.*, 2022.

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

[63] M. Fey, J. E. Lenssen, F. Weichert, and H. M"uller, "Splinecnn: Fast geometric deep learning with continuous b-spline kernels," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 869–877.

[64] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv:2003.04297*, 2020.

[65] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, no. 9, 2005.

[66] A. Beck and M. Teboulle, "Smoothing and first order methods: A unified framework," *SIAM J. Optim.*, vol. 22, no. 2, pp. 557–580, 2012.

[67] H. C. Indelman and T. Hazan, "Learning latent partial matchings with gumbel-ipf networks," in *Proc. Assoc. Advancement Artif. Intell. Stat.*, 2024, pp. 1513–1521.

[68] F. Zhou and F. De la Torre, "Factorized graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1774–1789, 2015.

[69] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1548–1554.

[70] Y. Xie, X. Wang, R. Wang, and H. Zha, "A fast proximal point method for computing exact wasserstein distance," in *Proc. Uncertainty Artif. Intell.*, 2020, pp. 433–453.

[71] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[72] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Proc. Neural Inf. Process. Syst.*, vol. 31, 2018.

[73] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio *et al.*, "A closer look at memorization in deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 233–242.
[74] T. Han, W. Xie, and A. Zisserman, "Temporal alignment networks for long-term video," in *Proc. Comput. Vis. Pattern Recognit.*, 2022, pp. 2906–2916.
[75] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," *Proc. Neural Inf. Process. Syst.*, vol. 34, pp. 9694–9705, 2021.
[76] T. Wang, H. Liu, Y. Li, Y. Jin, X. Hou, and H. Ling, "Learning combinatorial solver for graph matching," in *Proc. Comput. Vis. Pattern Recognit.*, 2020, pp. 7568–7577.
[77] M. Cho, K. Alahari, and J. Ponce, "Learning graphs to match," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 25–32.
[78] R. Wang, Z. Guo, W. Pan, J. Ma, Y. Zhang, N. Yang, Q. Liu, L. Wei, H. Zhang, C. Liu *et al.*, "Pygmtools: A python graph matching toolkit," *J. Mach. Learn. Res.*, vol. 25, pp. 1–7, 2024.
[79] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[80] P. Swoboda, C. Rother, H. Abu Alhaija, D. Kainmuller, and B. Savchynskyy, "A study of lagrangean decompositions and dual ascent solvers for graph matching," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 1607–1616.
[81] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

Yijie Lin received both the B.S. and Ph.D. degrees in computer science from Sichuan University, Chengdu, China. His research interests include multi-modal and AI4Science. In these areas, he has authored more than 10 articles in top-tier conferences and journals.

Mouxing Yang received the Ph.D. degree in computer science at the College of Computer Science, Sichuan University. His research interests include multi-modal learning and noisy correspondence learning. On these areas, he has authored more than 10 articles in the top-tier conferences and journals.

Peng Hu received the Ph.D. degree in computer science and technology from Sichuan University, China, in 2019. He is currently a professor at the College of Computer Science, Sichuan University. His research interests mainly focus on multimodal learning, cross-modal retrieval, and network compression. On these areas, he has authored more than 50 articles in the top-tier conferences and journals.

Jiancheng Lv (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is a professor and the Dean of the College of Computer Science, Sichuan University, Chengdu. His research interests include subspace learning in neural networks, natural language generation, and computer vision.

Hao Chen (Senior Member, IEEE) is an Assistant Professor at the Department of CSE&CBE, and Division of Life Science, The Hong Kong University of Science and Technology. He leads the Smart Lab, focusing on large and trustworthy AI for healthcare. He received the Ph.D. degree from The Chinese University of Hong Kong (CUHK) in 2017. He has 200+ publications in Nature Biomedical Engineering, Nature Communications, MICCAI, IEEE-TMI, TNNLS, MIA, CVPR, ICCV, ICLR, ICML, AAAI, Lancet Digital Health, Nature Machine Intelligence, etc. He received several premium awards such as Asian Young Scientist Fellowship, MICCAI Young Scientist Impact Award, and several best paper awards. He serves as the Associate Editor of multiple journals, including IEEE RBME, TMI, TNNLS, JBHI, CMIG, etc. He serves as the Program Committee of multiple international conferences, including Area Chair of ICLR 2025-2026, MICCAI 2021-2023, CVPR 2024-2026, ACM MM 2024, etc. He also led the team winning 15+ medical grand challenges.

Xi Peng (Senior Member, IEEE) is currently the Cheung Kong distinguished professor at the College of Computer Science, Sichuan University. His research interests mainly focus on machine learning, multi-media analysis, and AI4Science. In these areas, he has co-authored around 100 articles in Nature Communications, JMLR, TPAMI, ICLR, ICML, NeurIPS, and related venues. Dr. Peng has served as an Associate Editor for five journals including IEEE TPAMI and IEEE TIP.