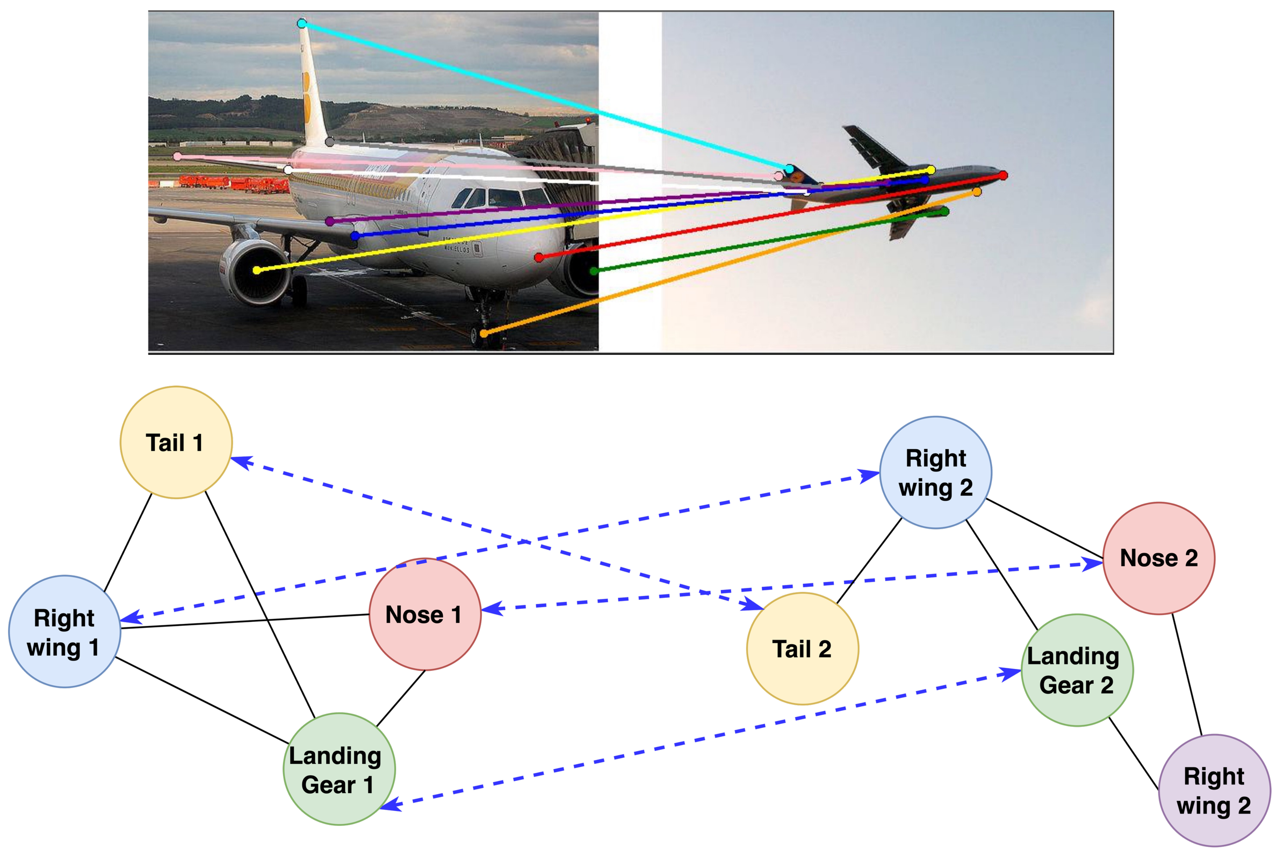
Contrastive General Graph Matching with Adaptive Augmentation Sampling

Jianyuan Bo, Yuan Fang
Singapore Management University, Singapore
{jybo.2020, yfang}@smu.edu.sg

IJCAI JEJU 2024

Introduction

Graph matching is crucial across various domains such as bioinformatics, social network analysis, and computer vision. The effectiveness of traditional methods, however, is curtailed by significant challenges, including those inherent in the burgeoning field of contrastive learning.

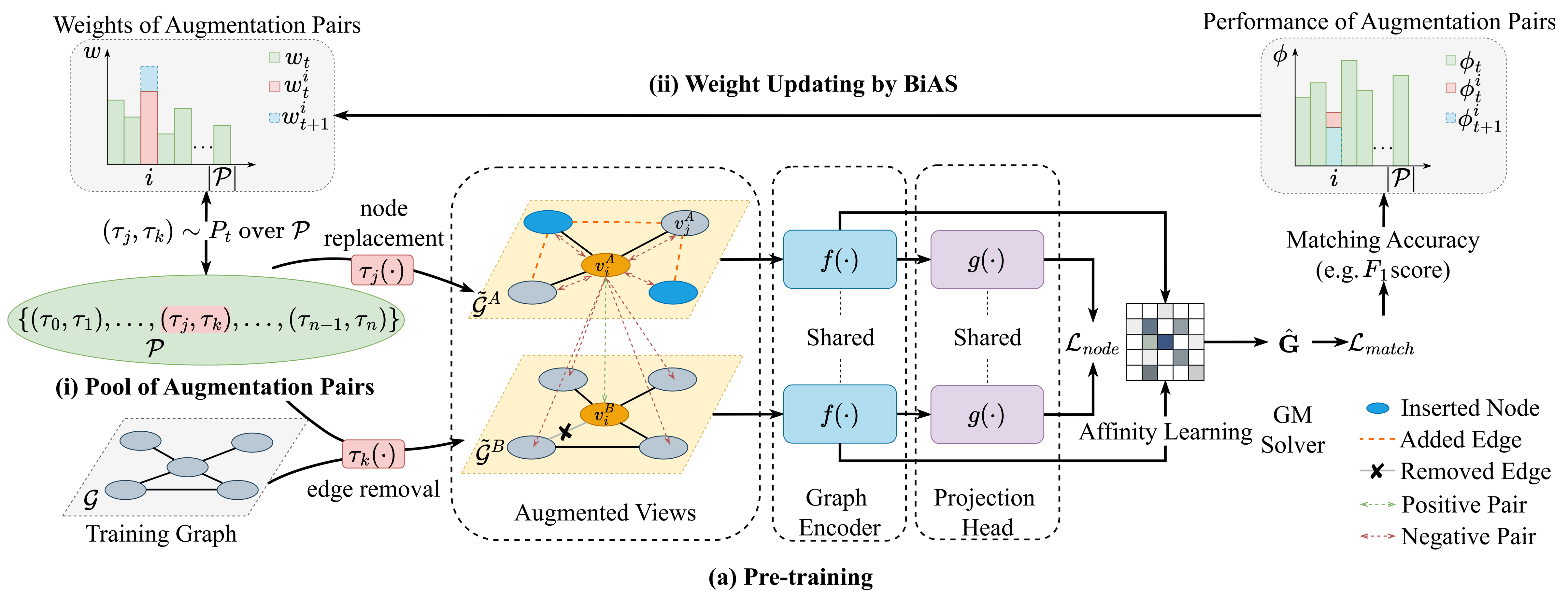


Example of visual graph matching.

Motivation

- **Dependency on Labeled Data**: Most graph matching techniques heavily rely on extensive labeled data for training. This dependence is resource-intensive and limits applicability in areas where such data is scarce or expensive to procure.
- **Lack of Generalizability**: Existing approaches often require additional side information or are tailored to specific graph types, which hinders their broader application. This specificity reduces the utility of graph matching methods in new or diverse areas that demand adaptable solutions.
- **Contrastive Learning Limitations**: While contrastive learning offers a promising direction for self-supervised learning in graph matching, it typically necessitates careful selection of augmentations to generate effective positive and negative samples. This requirement presents a challenge in tuning and selecting these augmentations without exacerbating the computational burden or risking model overfitting.

Proposed Method: GCGM



Graph-centric **C**ontrastive framework for **G**raph **M**atching (GCGM), leverages a comprehensive pool of graph augmentations to enhance robustness and effectiveness in graph matching. **B**oosting-inspired **A**daptive **A**ugmentation **S**ampler (BiAS) adaptively selects challenging augmentations tailored for graph matching, optimizing the learning process without manual tuning.

Pool of Augmentations Pairs

A distinctive feature of GCGM is its use of a large and diverse pool of graph augmentations, initially randomly initialized. This pool includes various structural and feature-based modifications.

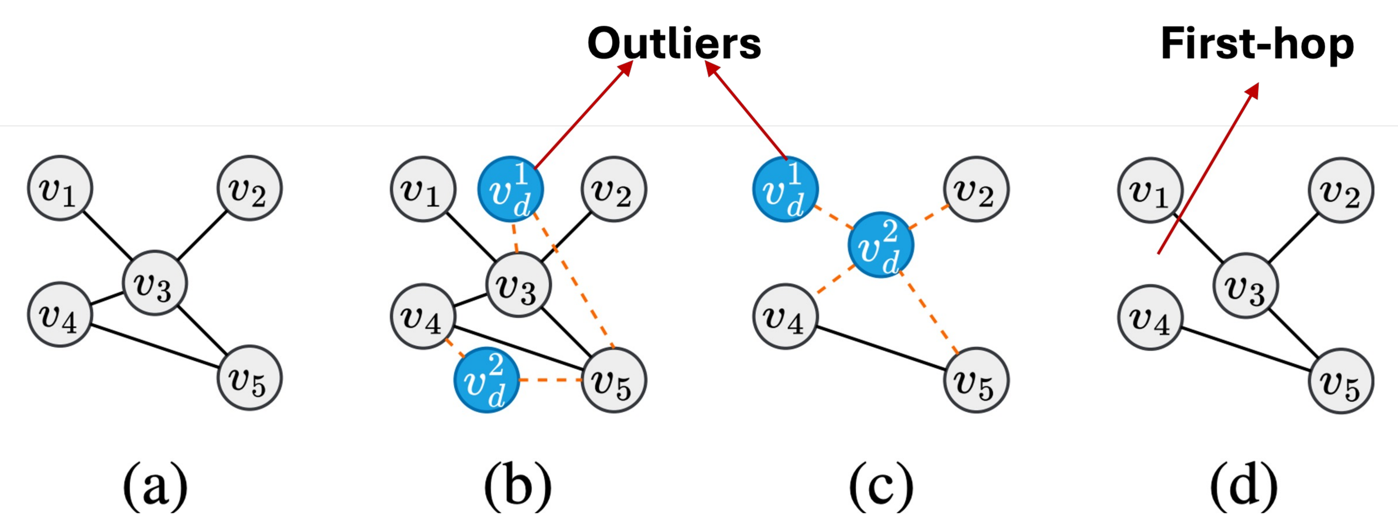


Figure 2: Graph augmentation: **(a)** input graph; **(b)** node insertion; **(c)** node replacement; **(d)** edge removal. The blue node represents the inserted node, and the dotted edge indicates the added edge.

Type	Matching scenario	Hyperparameters
NI	node outlier (unequal node count in two views)	$p_{\text{ni}} \in [0.1, 0.9]$: fraction of nodes inserted; $k_{\text{ni}} \geq 2$: size of subset; $aggr_{\text{ni}} \in \{\text{mean}, \text{max}\}$: aggregation function; $e_{\text{ni}} \geq 1$: number of edges inserted
NR	node outlier (equal node count in two views)	$p_{\text{nr}} \in [0.1, 0.9]$: fraction of nodes replaced; $k_{\text{nr}} \geq 2$: size of subset; $aggr_{\text{nr}} \in \{\text{mean}, \text{max}\}$: aggregation function; $e_{\text{nr}} \geq 1$: number of edges inserted
ER	sparse/noisy first-order connections	$p_{\text{er}} \in [0.1, 0.9]$: probability of each edge being removed
FS	feature variations & noises	$\alpha \in [0.2, 0.8]$: lower bound of uniform distribution; $\beta \in [1.2, 1.8]$: upper bound of uniform distribution

Table 1: Details of the four major types of graph augmentation.

BiAS

$$w_{t+1}^i = \lambda w_t^i + (1-\lambda)e^{\alpha(1-\phi_t^i)}$$

- **Adaptive Sampling Strategy**: BiAS introduces a novel weight update scheme that adaptively adjusts the probabilities of selecting specific augmentations based on their impact on the model's performance.
- **Weight Update Scheme**: The weights of augmentation pairs are dynamically updated based on their performance in improving the matching accuracy. Pairs that result in lower performance scores are sampled more frequently in subsequent training iterations, encouraging the model to focus on more challenging and informative examples.

$\lambda = 0.6, \alpha = 2$

w_t	ϕ_t	w_{t+1}
1	0.9	1.09
1	0.4	**1.93**

Example of weight update.

Experiments

Real-world Datasets

Methods	Pascal VOC Intsec	Pascal VOC Unfilt	Willow Intsec	SPair-71k Intsec	SPair-71k Unfilt
CIE (SUP)	66.8±0.4	-	82.6±0.2	69.3±0.3	-
BBGM (SUP)	77.3±0.1	55.0±0.1	96.2±0.1	77.7±0.2	48.4±0.2
NGMv2 (SUP)	76.8±0.1	56.7±0.1	94.5±0.3	76.6±0.2	49.8±0.08
IPFP	45.8±0.02	31.5±0.002	80.1±0.06	57.0±0.04	31.7±0.01
RRWM	47.2±0.02	31.7±0.001	83.4±0.09	58.6±0.05	32.3±0.01
SM	46.2±0.03	30.4±0.002	81.3±0.08	57.7±0.04	30.3±0.01
GANN-GM^	34.5±0.3	23.4±0.2	89.3±0.1	34.7±0.4	19.4±0.3
SCGM+BBGM	54.8±0.05	36.6±0.04	93.1±0.08	60.2±0.05	34.1±0.01
SCGM+NGMv2	50.8±0.1	32.9±0.03	84.2±0.1	59.8±0.1	30.5±0.3
GCGM+BBGM	56.8±0.02	36.2±0.01	94.4±0.3	60.6±0.1	**35.9**±0.07
GCGM+NGMv2	**57.3**±0.11	**37.4**±0.07	**95.0**±0.1	**62.6**±0.02	35.4±0.07

Synthetic Dataset

Methods	Synthetic Intsec	Synthetic Unfilt
GANN-GM^	11.2±0.04	10.2±0.03
SCGM + BBGM	33.5±2.0	24.3±1.2
SCGM + NGMv2	35.2±0.6	25.0±0.4
GCGM	**58.1**±0.5	**39.9**±0.4

Intersection
No outliers.

Unfiltered
Includes all nodes, assessing robustness to outliers and node count variations.

Graph Augmentation

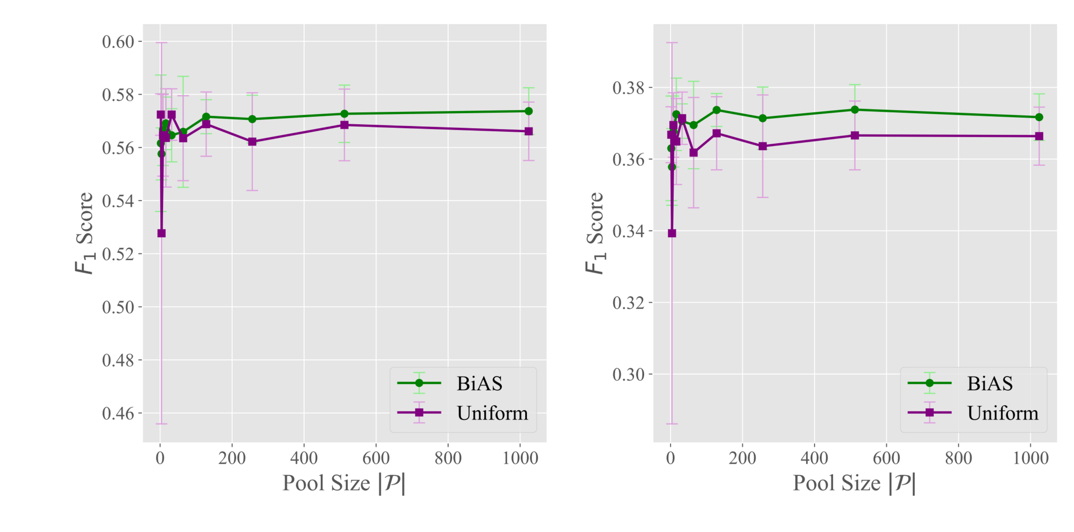
Augmentation Set	Pascal VOC Intsec	Pascal VOC Unfilt	Willow Intsec	SPair-71k Intsec	SPair-71k Unfilt	Synthetic Intsec	Synthetic Unfilt
$\mathcal{T}\backslash$NI	56.9	36.6	94.8	61.8	34.9	57.9	**40.5**
$\mathcal{T}\backslash$NR	56.5	36.5	**95.1**	61.8	34.4	57.8	40.0
$\mathcal{T}\backslash$ER	57.3	37.2	95.0	59.8	32.5	57.9	40.0
$\mathcal{T}\backslash$FS	**57.5**	37.2	95.0	62.1	35.1	57.8	40.3
$\mathcal{T}$	57.3	**37.4**	95.0	**62.6**	**35.4**	**58.1**	39.9

Table 4: Ablation study on graph augmentations.

BiAS and Augmentation Pool

Settings	$\mathcal{P}$	Pascal VOC Intsec	Pascal VOC Unfilt	Pascal VOC Time/h	Willow Intsec	Willow Time/h	SPair-71k Intsec	SPair-71k Unfilt	SPair-71k Time/h
Random	✗	55.0	35.9	0.26	93.8	0.04	61.4	35.2	0.38
Tuning	✗	55.9	36.8	23.76	94.8	4.54	61.5	35.6	31.96
Tuning + BiAS	✗	55.8	37.0	23.95	**95.4**	4.54	61.9	**36.0**	31.88
Uniform	✓	56.9	36.7	0.32	94.7	0.05	62.0	34.8	0.35
BiAS	✓	**57.3**	**37.4**	0.39	95.0	0.05	**62.6**	35.4	0.34

Table 5: Performance of different initialization of augmentations and the use of augmentation pool. Time (hour) represents the total wall clock time spent on tuning the augmentations (for 'Tuning' methods) and training the model. The '$\mathcal{P}$' column indicates if a pool of augmentation pairs is used.



(a) Intersection (b) Unfiltered

Figure A: Comparison of BiAS and 'Uniform' samplers across different augmentation pool sizes on Pascal VOC dataset.