

# HOGDA: Boosting Semi-supervised Graph Domain Adaptation via High-Order Structure-Guided Adaptive Feature Alignment (Supplementary Materials)

Anonymous Authors

**Table 1: The statistics of three real-world graphs. Note that ‘#’ means ‘the number of’. ‘Attr.’ refer to ‘Attributes’. ‘Avg.’ represents ‘Average’.**

Graph	#Nodes	#Edges	#Attr.	Avg. Degree	Label Proportion (%)
ACMv9 (A)	9,360	15,602	5,571	1.667	20.5/29.6/22.5/8.6/18.8
Citationv1 (C)	8,935	15,113	5,379	1.691	25.3/26.0/22.5/7.7/18.5
DBLPv7 (D)	5,484	8,130	4,412	1.482	21.7/33.0/23.8/6.0/15.5

## 1 EXPERIMENTS

### 1.1 Datasets

Our experiments involves three real-world graphs: *ACMv9* (A), *Citationv1* (C), and *DBLPv7* (D). Table 1 displays various statistical information of three graphs, including graph scale, attributes, average degree, and label proportion. We can observe substantial intrinsic discrepancy among these graphs. In this paper, we adopt an alternating approach where we select one of these graphs as the source domain, while considering the remaining two as the target domains.

### 1.2 Implementation Details

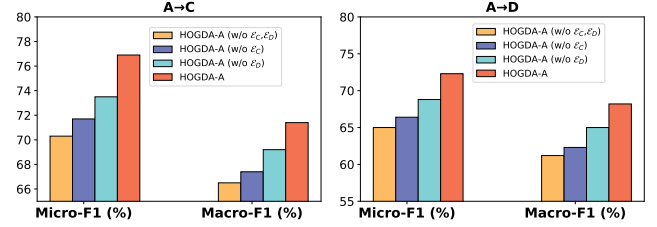
We conduct our experiments using the PyTorch library and follow the standard protocols [2] for SGDA in all experiments. We employ a two-layer GCN as the feature extractor  $\mathcal{F}$  of our HOGDA model following [2]. We perform each random experiment five times record the average accuracy along with standard deviation.

To achieve stable network optimization, we employ Adam optimizer with a weight decay of 0.001 and an initial learning rate of 0.001 during training. In terms of the trade-off coefficients  $\eta$  and  $\beta$ , we choose  $\eta = 1$  and  $\beta = 0.5$  for all transfer tasks. Notably, instead of fixing the parameter  $\eta$ , we adopt a progressive schedule to dynamically adjust  $\eta$  from 0 to 1 by multiplying by  $\frac{1 - \exp(-10\varrho)}{1 + \exp(-10\varrho)}$  to more stably train the domain discriminator  $\mathcal{D}$ , where  $\varrho$  is the training progress. We set the training epoch to 200 across all experiments. Furthermore, the hyper-parameter  $K$  (i.e., order of moment feature) in the HSIM module is set to 3 (see Figure X for further analysis). Note that it is also simple to change the prior  $u$  in TNC strategy from the uniform distribution to any arbitrary distribution in the objective function  $\mathcal{I}$  if there is any extra knowledge about the frequency of clusters. Additionally, the dimension of node features  $e$  is consistently set to 512 for all methods, including the compared methods.

## 2 MORE EXPERIMENTS AND ANALYSIS

### 2.1 More Ablation Study

**7) Effect of Mixed Entropy-aware Weighted Mechanism  $w$ :** To show the effectiveness of the node weighted mechanism  $w$  in



**Figure 1: Transfer performance (%) with node weighted strategies on tasks A→C and A→D. Source Graph Label Rate: 5%.**

our AWDA strategy, we compare HOGDA-A with its three variants on two typical transfer tasks: **A→C** and **A→D**. The variants of HOGDA-A are as follows: **(1) HOGDA-A (w/o  $\mathcal{E}_D$ )**, the variant only utilizes the entropy of classifier output  $\mathcal{E}_C$  to estimate node transferability, i.e.,  $w(n_i) = 1 + e^{-\mathcal{E}_C}$ . **(2) HOGDA-A (w/o  $\mathcal{E}_C$ )**, the variant simply utilizes the entropy of discriminator output to estimate node transferability, i.e.,  $w(n_i) = 1 + e^{\mathcal{E}_D}$ . **(3) HOGDA-A (w/o  $\mathcal{E}_C, \mathcal{E}_D$ )**, the variant assigns equal weight to different nodes, i.e.,  $w(n_i) = 1$ .

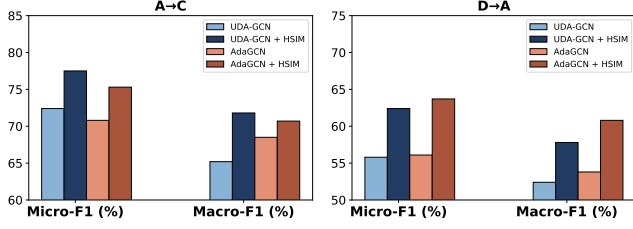
The results shows in Figure 1 reflect the following observations: **(1)** HOGDA-A (w/o  $\mathcal{E}_D$ ) greatly outperforms HOGDA-A (w/o  $\mathcal{E}_C$ ) on both tasks, as the entropy of classifier output  $\mathcal{E}_C$  can provide better guidance for the model to achieve the fine-grained alignment of category distributions. **(2)** HOGDA-A works better than HOGDA-A (w/o  $\mathcal{E}_C$ ) and HOGDA-A (w/o  $\mathcal{E}_D$ ), indicating that dynamically combining  $\mathcal{E}_C$  and  $\mathcal{E}_D$  to re-weight nodes during the adversarial domain alignment can facilitate the model to learn more transferable features. **(3)** Compared to HOGDA-A (w/o  $\mathcal{E}_C, \mathcal{E}_D$ ), HOGDA and the remaining two variants achieve significant performance gains on both tasks, indicating the effectiveness of re-weighting nodes based on their transferability during the adversarial domain adaptation process.

**8) Effect of Trustworthy Weighted Mechanism  $\Omega$ :** To demonstrate the superiority of the trustworthy weighted mechanism  $\Omega$  in our TNC strategy, we compare HOGDA-T with its two variants on tasks **A→C** and **A→D**. The variants of HOGDA-T are as follows: **(1) HOGDA-T (w/o  $\mathbb{W}$ )**, the variant simply utilize the spatial prototype information  $\mathbb{S}$  to estimate cluster assignment for each node, i.e.,  $\Omega(i, c) = \mathbb{S}(i, c)$ . **(2) HOGDA-T (w/o  $\mathbb{S}$ )**, the variant only utilizes the classifier prediction information  $\mathbb{W}$  to assign cluster assignment for each node, i.e.,  $\Omega(i, c) = \mathbb{W}(i, c)$ .

As reported in Table 2, compared to variants HOGDA-T (w/o  $\mathbb{W}$ ) and HOGDA-T (w/o  $\mathbb{S}$ ) that solely employ  $\mathbb{S}$  or  $\mathbb{W}$  to estimate clustering assignments for each node, our HOGDA-T achieves higher and more stable transfer performance, as it adaptively combine  $\mathbb{S}$  and  $\mathbb{W}$  to predict the cluster assignment. The improved results

**Table 2: Transfer performance (%) with different cluster assignment mechanisms on tasks  $A \rightarrow C$  and  $A \rightarrow D$ . Source Graph Label Rate: 5%.**

Methods	$A \rightarrow C$		$A \rightarrow D$	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
HOGDA-T (w/o $\mathbb{W}$ )	71.5 $\pm$ 0.69	68.3 $\pm$ 0.88	64.7 $\pm$ 0.62	61.1 $\pm$ 1.15
HOGDA-T (w/o $\mathbb{S}$ )	71.8 $\pm$ 0.83	68.7 $\pm$ 0.95	64.4 $\pm$ 0.81	60.6 $\pm$ 1.29
HOGDA-T	74.9 $\pm$ 0.42	70.8 $\pm$ 0.71	69.0 $\pm$ 0.49	64.5 $\pm$ 0.92



**Figure 2: Flexibility of HSIM Module. Source Graph Label Rate: 5%.**

**Table 3: Detailed training time analysis on task  $A \rightarrow C$ .**

Transfer Task	Method	Training Time
$A \rightarrow C$	HOGDA-A (Baseline)	62.1167 s
	SGDA	76.6956 s
	HOGDA	79.7512 s

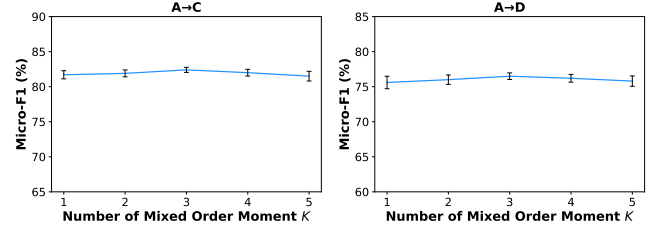
suggest that our trustworthy weighted mechanism can effectively guide the discriminative clustering of unlabeled nodes. Additionally, the results with a smaller fluctuation range also imply the stability of our trustworthy weighted mechanism  $\Omega$ .

## 2.2 More Analysis

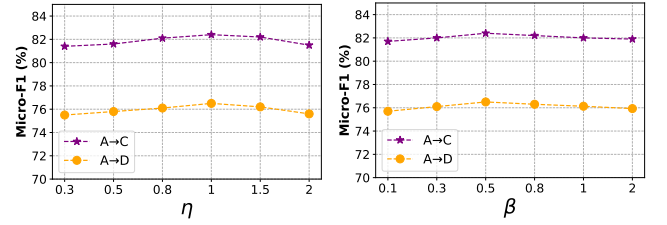
**9) Flexibility of HSIM Module:** Other GTL methods can also achieve performance improvements by leveraging the HSIM module  $\mathcal{H}$ . Two typical GTL methods, UDA-GCN [4] and AdaGCN [1], are used as baselines. We compare the Baseline and its variant Baseline + HSIM in terms of transfer performance on two transfer tasks  $A \rightarrow C$  and  $D \rightarrow A$ .

The results illustrated in Figure 2 demonstrate the addition of HSIM module can help the module capture more domain-invariant node features and greatly boost generalization performance, particularly on difficult transfer tasks (e.g.,  $D \rightarrow A$ ). There improved results also imply the importance of graph structure information in GTL task.

**10) Training Time:** Table 3 shows the training time of HOGDA-A (baseline), SGDA [2] and our proposed HOGDA on  $A \rightarrow C$  task. HOGDA-A simply use the source domain classification loss  $\mathcal{L}_{cls}$  and AWDA strategy to optimize the model, resulting in the shortest training time. Because SGDA utilizes posterior scores-based pseudo-labeling strategy to guide the clustering of unlabeled nodes, its training time will be somewhat greater than that of HOGDA-A.



**Figure 3: Parameter sensitivity analysis of hyper-parameter  $K$  in the HSIM module on tasks  $A \rightarrow C$  and  $A \rightarrow D$ .**



**Figure 4: Parameter sensitivity analysis of trade-off coefficients  $\eta$  and  $\beta$  on two transfer tasks  $A \rightarrow C$  and  $A \rightarrow D$ .**

Due to the introduction of HSIM module and TNC strategy, our HOGDA models require longer training time (**1.28x**) than baseline HOGDA-A, which does not significantly increase the training time but brings a large performance gain.

## 2.3 Parameter Sensitivity

**11) Effect of Number of Mixed Order Moment Features  $K$ :** As depicted in Figure 3, we investigate the effect of hyper-parameter  $K$  (i.e., the number of mixed order moment features) in the HSIM module on transfer tasks  $A \rightarrow C$  and  $A \rightarrow D$ . It can be observed that as  $K$  increase, the model's transfer performance first improves and then declines. It suggests that incorporating a broader range of high-order moments information can boost the model's generalization ability. However, including an excessive number of high-order moment features can lead to a decline in the model's performance. This is because excessively high-order moments may cause more instability during the training process [3], resulting in performance degradation.

**12) Effect of Trade-off Coefficient:** Figure 4 illustrates our evaluation of the sensitivity of several hyper-parameters on transfer tasks  $A \rightarrow C$  and  $A \rightarrow D$ . The evaluated hyper-parameters include two trade-off parameters, namely  $\eta$  and  $\beta$ . As  $\eta$  grows, the model's transfer ability first rises and then falls, implying that selecting an appropriate value to adjust the AWDA loss  $\mathcal{L}_{awda}$  can effectively promote the learning of transferable features and mitigate feature distributions discrepancy. In terms of the coefficient  $\beta$ , our model exhibits strong robustness to variations in  $\beta$ , and we find that 0.5 is the optimal value for both tasks.

## 3 CODE

We assure a release of code following publication. We sincerely appreciate the reviewers' patience.

REFERENCES

[1] Quanyu Dai, Xiao-Ming Wu, Jiaren Xiao, Xiao Shen, and Dan Wang. 2023. Graph Transfer Learning via Adversarial Domain Adaptation With Graph Convolution. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2023), 4908–4922. <https://doi.org/10.1109/TKDE.2022.3144250>

[2] Ziyue Qiao, Xiao Luo, Meng Xiao, Hao Dong, Yuanchun Zhou, and Hui Xiong. 2023. Semi-supervised domain adaptation in graph transfer learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. 2279–2287.

[3] Sarunas J Raudys, Anil K Jain, et al. 1991. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis and machine intelligence* 13, 3 (1991), 252–264.

[4] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. 2020. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of The Web Conference 2020*. 1457–1467.