Domain Adaptive Classification on Heterogeneous Information Networks

Shuwen Yang[1] , **Guojie Song**[1*] , **Yilun Jin**[2] and **Lun Du**[3]
[1]Key Laboratory of Machine Perception (Ministry of Education), Peking University, China
[2]The Hong Kong University of Science and Technology, Hong Kong SAR, China
[3]Microsoft Research, China
{swyang, gjsong}@pku.edu.cn, yilun.jin@connect.ust.hk, lun.du@microsoft.com

Abstract

Heterogeneous Information Networks (HINs) are ubiquitous structures in that they can depict complex relational data. Due to their complexity, it is hard to obtain sufficient labeled data on HINs, hampering classification on HINs. While domain adaptation (DA) techniques have been widely utilized in images and texts, the heterogeneity and complex semantics pose specific challenges towards domain adaptive classification on HINs. On one hand, HINs involve multiple levels of semantics, making it demanding to do domain alignment among them. On the other hand, the trade-off between *domain similarity* and *distinguishability* must be elaborately chosen, in that domain invariant features have been shown to be homogeneous and uninformative for classification. In this paper, we propose Multi-space Domain Adaptive Classification (MuSDAC) to handle the problem of DA on HINs. Specifically, we utilize multi-channel shared weight GCNs, projecting nodes in HINs to multiple spaces where pairwise alignment is carried out. In addition, we propose a heuristic sampling algorithm that efficiently chooses the combination of channels featuring distinguishability, and moving-averaged weighted voting scheme to fuse the selected channels, minimizing both transfer and classification loss. Extensive experiments on pairwise datasets endorse not only our model's performance on domain adaptive classification on HINs and contributions by individual components.

1 Introduction

Heterogeneous Information Networks (HIN) are one of the most prevalent data structures and have been wildly utilized to store complex relational data. Node classification in HINs is an important but equally challenging task in that HINs contain various types of nodes and edges, hence rich semantics. Up to now, many representation learning or collective classification models have been proposed to support semi-supervised classification on HINs [Zhang *et al.*, 2018; Wang *et al.*, 2019].

However, supervised models highly rely on labeled data, which may be costly or even impossible to obtain for complex structures like HINs [Jin *et al.*, 2020]. As a result, we intuitively resort to transfer learning when labeled data on HINs are scarce.

Domain Adaptation (DA), which supports transfer learning from a source domain with sufficient labeled data to an unlabeled target domain by minimizing their domain discrepancy [Mansour *et al.*, 2009; Long *et al.*, 2015], has already caught interests from Computer Vision (CV) and Natural Language Processing (NLP) [Long *et al.*, 2018; Rozantsev *et al.*, 2018]. Domain adaptation methods, such as Maximum Mean Discrepancy (MMD) [Dziugaite *et al.*, 2015] and Generative Adversarial Network (GAN) [Goodfellow *et al.*, 2014], are able to align the embedding distributions of different domains, enabling the transfer of downstream machine learning models. We hence resort to DA techniques as potential solutions to the problem of transfer learning on HINs.

Many embedding models on HINs apply a multi-channel architecture based on meta-paths [Sun *et al.*, 2011], where nodes are projected to several embedding spaces through multiple GNN channels before finally fused to a single collection of representations for downstream tasks [Zhang *et al.*, 2018]. While it seems that transferable classification on HINs can be accomplished by simply adding a regularization to such an architecture to minimize distribution discrepancy, it should be noted that the heterogeneity and rich semantics of HINs pose specific challenges:

- HINs feature multiple levels of semantics where domain alignment should be done, which makes it difficult to simultaneously align them in a single embedding space.
- Domain adaptation underscores domain invariant features, which are likely to be homogeneous and uninformative for classification [Chen *et al.*, 2019]. On the other hand, features indicative for classification are generally domain-variant. We hence conclude that the trade-off between *transfer* and *classification*, *domain similarity* and *distinguishability* is necessary for the task of DA, especially on HINs where classification is hard.

To solve these problems, in this paper we propose Multi-

[*] Corresponding author

Space Domain Adaptive Classification (MuSDAC)[1], which adopts multi-channel shared weight GCNs [Kipf and Welling, 2016] to project the nodes from both source and target domains to multiple embedding spaces, where multi-space alignment is applied, such that the rich levels of semantics of HINs are preserved independently within each space. By doing so, only one pairwise domain alignment, instead of multiple pairs, is needed in each space.

In addition, we propose a *Two-level Selection* strategy that efficiently aggregates embedding spaces to ensure both *domain similarity* and *distinguishability*, in response to the trade-off mentioned above. First, we utilize *Heuristic Combination Sampling Algorithm*, a polynomial-time algorithm that selects spaces featuring clear class boundaries, alleviating the need for combinatorial search of spaces. In what follows, we propose *Moving-averaged Weighted Voting*, a weighting scheme to elaborately fuse the selected spaces, hence minimizing both transfer and classification loss. We evaluate MuSDAC quantitatively on three pairs of networks where MuSDAC outperforms various baselines on transferable classification. We also carry out model analysis and visualization to verify contributions of individual model components.

Our contributions are summarized as follows:

- We address the unexplored problem of transferable classification among HINs with MuSDAC, which adopts multi-channel shared weight GCNs [Kipf and Welling, 2016] and applies multi-space alignment to enable domain adaptation on different semantic spaces.
- To achieve the trade-off between transfer and classification, we design *Heuristic Combination Sampling Algorithm* to pick out discriminative combinations efficiently and apply *Moving-averaged Weighted Voting* to assemble the outputs from all channels. We also theoretically analyze the weighted voting in multi-space alignment.
- We carry out both quantitative and qualitative experiments where MuSDAC demonstrates itself by outperforming competitive counterparts.

2 Related Work

2.1 HIN Classification

Many models have been designed to perform classification on HINs [Hosseini *et al.*, 2018]. [Zhang *et al.*, 2018] proposed GraphInception to handle collective classification on HINs by learning the deep relational features. Besides, heterogeneous network embedding models, which project nodes in HINs to a low-dimensional space, also enables classification [Gui *et al.*, 2016; Dong *et al.*, 2017; Shi *et al.*, 2018; Lin *et al.*, 2019]. HAN [Wang *et al.*, 2019] learns representations based on hierarchical attention, while NEP [Yang *et al.*, 2019] leverages distributed embeddings to represent objects and dynamically composed modular networks to model their complex interactions. However, these models may fail on new domains without labeled instances due to domain discrepancies.

2.2 Domain Adaption

Domain adaption is wildly used to enable transfer learning among different but relevant domains without manual tagging in fields like CV and NLP. Recent studies mainly focus on learning the domain invariant features for instances in different domains by minimizing their distribution discrepancy via regularizations such as MMD or GAN. [Long *et al.*, 2018; Rozantsev *et al.*, 2018]

To the best of our knowledge, DANE [Zhang *et al.*, 2019] is the only work considering domain adaption among homogeneous networks in the field of network representation. However, DANE is unable to deal with the rich semantics in HINs as the shared weight GCN architecture is specially designed for homogeneous networks.

3 Problem Statement

3.1 Definitions

Definition 1 (Heterogeneous Network [Shi *et al.*, 2016])**.** *A heterogeneous network $\mathcal{G}$ consists of a node set $\mathcal{V} = \bigcup_{i=1}^{n} \mathcal{V}_i$ with n types of nodes and an edge set $\mathcal{E} = \bigcup_{i=1}^{m} \mathcal{E}_i$ with m types of edges.*

*In heterogeneous networks, a **meta-path** Φ_i is the path in the form of $\mathcal{V}_{i_1} \xrightarrow{\mathcal{E}_{i_1}} \mathcal{V}_{i_2} \xrightarrow{\mathcal{E}_{i_2}} \cdots \xrightarrow{\mathcal{E}_{i_l}} \mathcal{V}_{i_(l+1)}$, which defines a composite relationship between two nodes $\mathcal{V}_{i_1}, \mathcal{V}_{i_(l+1)}$.*

Definition 2 (Multi-channel Network [Zhang *et al.*, 2018])**.** *Given $\mathcal{V}_1$ as the node type pending classification. We decompose HIN to a multi-channel network with meta-path set $\mathbf{\Phi} = \{\Phi_1, \cdots, \Phi_N\}$, where each channel is a homogeneous network containing nodes $\mathcal{V}_1$ connected via a certain type of meta-path. The resulting network is defined as $G = \{(\mathcal{V}_l, A_l)|l = 1, \cdots, N\}$, where **meta-path adjacency matrix** A_l represents the number of meta-path Φ_l connecting each node pair in $\mathcal{V}_1$.*

3.2 Problem: Transferable Classification on HINs

Given two HINs $(\mathcal{G}_S, \mathcal{X}_S)$ and $(\mathcal{G}_T, \mathcal{X}_T)$, where $\mathcal{G}_S, \mathcal{G}_T$ share the same node and edge types and $\mathcal{X}$ represents the features of $\mathcal{V}_1$, transferable classification on HINs aims to utilize the structural information on both networks as well as labels on $\mathcal{V}_{S,1}$ to predict the labels on $\mathcal{V}_{T,1}$.

4 Proposed Method

4.1 Overview of MuSDAC

In this section we introduce MuSDAC, an unsupervised domain adaptive classification model on HINs (see Fig. 1 for an overview). We first introduce general pipelines before elaborating on details of individual components.

Multi-channel Shared Weight GCN

To process the heterogeneous information, we decompose source and target HINs $\mathcal{G}_S, \mathcal{G}_T$ to multi-channel networks with meta-path set $\mathbf{\Phi}$, and feed both of them through N independent GCNs [Kipf and Welling, 2016] to produce **original channel** embedding sets $\mathcal{C} = \{C_l|l = 1, \cdots, N\}$:

$$C_l = \hat{A}_l \sigma(\hat{A}_l \mathcal{X} W_l^{(0)}) W_l^{(1)} \tag{1}$$

[1] Code available on https://github.com/PKUterran/MuSDAC

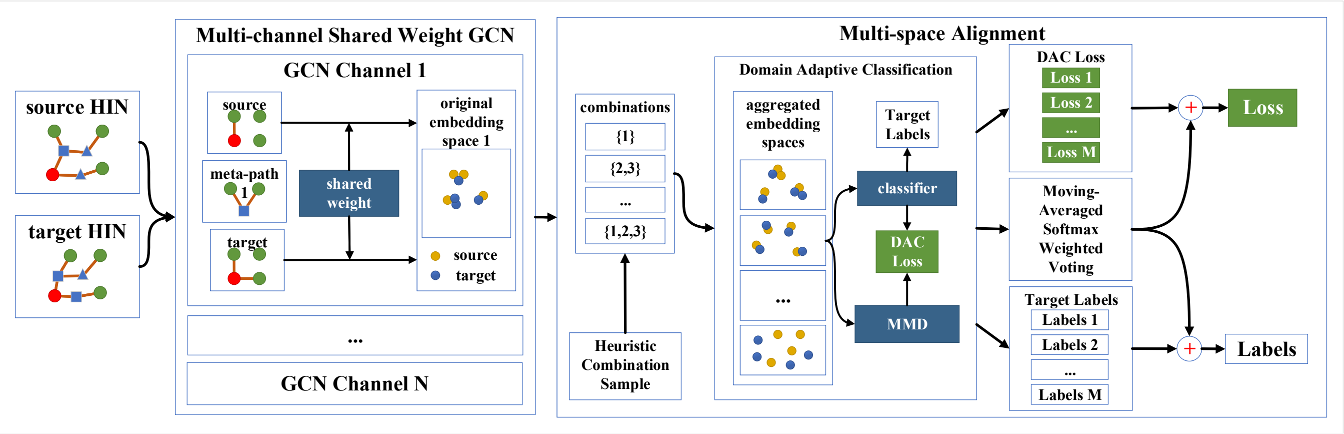

Figure 1: An overview of MuSDAC, which uses multi-channel shared weight GCNs to process HIN based on meta-path and applies multi-space alignment to recognize transferable semantic information for DA classification task.

where $\hat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$ with $\widetilde{A} = A + I_N$ and $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$. Inside channel l, we apply a shared parameter set $\{W_l^{(0)}, W_l^{(1)}\}$ to project the nodes in both networks to the same embedding space.

Multi-space Alignment

To extract complex relational features in HINs, we combine $\mathcal{C}_Z = \{C_l | l \in Z\}$, a subset of $\mathcal{C}$ with combination $Z \subseteq \{1, \cdots, N\}$ and $Z \neq \emptyset$, through 1-dimensional convolution [Zhang *et al.*, 2018]. We denote $\mathcal{M}_Z$ to be the embeddings of **aggregated channel** with combination Z.

Here we utilize Algorithm 1 to generate a set of combinations $\mathcal{Z} = \{Z_j | j = 1, \cdots, M\}$, containing $M = \mathcal{O}(N)$ combinations featuring distinguishability, which will be elaborated in Section 4.2. We then re-project the nodes to several new embedding spaces and get **aggregated channel** embedding set $\mathcal{M} = \{M_{Z_j} | j = 1, \cdots, M\}$ where $\mathcal{M}_{Z_j}$ is a single embedding matrix.

In the j-th aggregated channel Z_j, we denote $\mathcal{M}_{Z_j,S}, \mathcal{M}_{Z_j,T}$ as the embeddings of source and target instances, upon which a classifier is employed for prediction

$$\hat{\mathcal{Y}}_j = \mathrm{softmax}(\mathcal{M}_{Z_j} W_j^C) \tag{2}$$

where W_j^C are parameters of the classifier in the j-th channel.

Model Learning

Theorem 1 (Domain Adaptive Classification (DAC) [Ben–David *et al.*, 2010])**.** *The upper-bound of prediction error on target domain can be reduced by minimizing: (a) the error of hypothesis h on source domain; (b) the $\mathcal{H}\Delta\mathcal{H}$ distance between both domains, which measures the domain discrepancy; (c) the error λ of the ideal joint hypothesis h^*:*

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{M}_S, \mathcal{M}_T) + \lambda \tag{3}$$

According to Theorem 1, we apply DAC, where the prediction error on target labels in the j-th channel can be reduced by minimizing the classification loss on $M_{Z_j,S}$ as well as the distance between $M_{Z_j,S}$ and $M_{Z_j,T}$:

$$L_{Z_j,D} = \mathrm{CE}(\hat{\mathcal{Y}}_{j,S}, \mathcal{Y}_S),\ L_{Z_j,T} = \mathrm{MMD}(M_{Z_j,S}, M_{Z_j,T})$$
$$L_{Z_j} = L_{Z_j,D} + \gamma L_{Z_j,T} \tag{4}$$

where CE is the cross entropy function, MMD is the Maximum Mean Discrepancy measuring the distribution distance and γ is a hyper-parameter controlling the gradients.

In $L_{Z_j,D}$ the final prediction is a weighted voting of the outputs from all classifiers with a weight vector $\boldsymbol{\theta}$. The overall loss is also a weighted sum of DAC losses from the aggregated channels, where the same weight $\boldsymbol{\theta}$ is adopted.

$$\hat{\mathcal{Y}} = \sum_j \theta_j \hat{\mathcal{Y}}_j \qquad L = \sum_j \theta_j L_{Z_j} \tag{5}$$

4.2 Heuristic Combination Sampling Algorithm

In this section we introduce our design of *Heuristic Combination Sampling Algorithm*. We first introduce and verify an important assumption related to the design of our algorithm, before introducing the algorithm.

Assumption of Estimating Distinguishability

Inspired by [Rafailidis and Weiss, 2019], we formally define *distinguishability* and *domain similarity* of a combination.

Definition 3. *Given Z, by minimizing $L_Z = L_{Z,D} + \gamma L_{Z,T}$ according to Eq.4, we measure Z's distinguishability D_Z and domain similarity T_Z by:*

$$D_Z = -L_{Z,D}, \qquad T_Z = -L_{Z,T} \tag{6}$$

Definition 4 (Sub-combination)**.** *$\widetilde{Z}$ is a sub-combination of Z iff $\widetilde{Z} \subset Z \wedge \widetilde{Z} \neq \emptyset$. We denote Z's sub-combination set as* $\mathbf{S}(Z)$ [1].

Based on the definitions we present an important assumption on estimating D_Z.

[1] In this paper we use lowercase $z \in \{1, 2 ... n\}$ to denote a channel id, capitalized Z to denote a combination, i.e. set of channels $Z \subseteq \{1, 2 ... n\}$ and calligraphy $\mathcal{Z}$ to denote a set of combinations $\mathcal{Z} = \{Z_i\}$.

Assumption 1 (Estimation of Distinguishability)**.** *For a combination Z and $Z_{\mathcal{T}} = \underset{Z' \in \mathbf{S}(Z)}{\arg\max} T_{Z'}$, It is satisfied that:*

$$|D_Z - D_{Z_{\mathcal{T}}}| \leq \delta \quad with \quad p > 1 - \varepsilon \tag{7}$$

In brief, D_Z can be estimated by the distinguishability of Z's sub-combination with highest domain similarity.

Analysis of Assumption 1

In this section we present analysis of Assumption 1 from both theoretical and empirical point-of-view.

Definition 5 (Emphasis of features in embedding matrix)**.** *Given a matrix $\mathcal{M}$, we apply singular value decomposition (SVD), yielding $\mathcal{M} = \mathbf{U\Sigma V^T}$. We define $\mathbf{U}_i$ (i-th singular vector) to be an extracted feature and its corresponding singular value $\mathbf{\Sigma}_{ii}$ to be its emphasis. The features embodied in $\mathcal{M}$ is defined as $\mathcal{F} = \{\mathbf{U}_i, \mathbf{\Sigma}_{ii} > \varepsilon\}$.*

In domain adaptation, the singular vectors representing the domain invariants are more likely to be emphasized. In addition, some of them might be excessively emphasized and hence acquire an unduly high singular value. We refer to them as **"trap-vectors"**, in that they boost domain similarity at the expense of other singular vectors that embody rich semantics crucial for distinguishability [Chen *et al.*, 2019].

Based on this phenomenon, we formulate how features are extracted during aggregation. Denote $\mathcal{F}_Z$ to be the features embodied in the aggregated channels with combination Z:

$$\mathcal{F}_Z \subseteq \bigcup_{Z' \in \mathbf{S}(Z)} \phi_{Z'}(\mathcal{F}_{Z'}) + \widetilde{\phi}(\widetilde{\mathcal{F}}_Z) \tag{8}$$

where $\widetilde{\mathcal{F}}_Z$ refers to the features after aggregating all channels in Z and ϕ is a function mapping the features to the corresponding embedding space $\mathcal{M}_Z$. When filtering the features, the combination $Z_{\mathcal{T}} = \arg\max_{Z' \in \mathbf{S}(Z)} T'_Z$ should generally be favorable in domain adaptation. However, such combination is vulnerable to "trap-vectors" as it is likely to impose excessive weight on domain invariants [Chen *et al.*, 2019] and stifle semantics from other channels. On such occasions, low $D_{Z_{\mathcal{T}}}$ indicates low D_Z and vice versa. Consequently, we claim that D_Z can be approximated by $D_{Z_{\mathcal{T}}}$ since $Z_{\mathcal{T}}$ is most likely to be extracted in domain adaptation, where "trap vectors" are likely to be produced.

Alternatively from the empirical point of view, we visualize the relationship between D_Z and $D_{Z_{\mathcal{T}}}$ in Fig.2. The result shows that $D_{Z_{\mathcal{T}}}$ approximately follows Z_T in most of the times, hence verifying Assumption 1.

Algorithm: Heuristic Combination Sampling

To select linear number of combinations with high distinguishability, a naive method is to minimize Eq.4 (referred to as **pre-test**) for every combination Z and compare D_Z. However, such enumeration would bring prohibitive complexity ($\mathcal{O}(2^N)$). Alternatively, we design a heuristic algorithm to select the combinations Z, which is presented in Algo.1. In the first iteration, we pre-test the combinations in $\mathcal{Z}_{test}(|Z| = 1, \forall Z \in \mathcal{Z}_{test})$, then we try to append a new channel for each in $\mathcal{Z}_{test}$ and predict D_Z (Line 7-13) according to Assumption 1. Finally, a linear number of combinations that have high D_Z will form new $\mathcal{Z}_{test}$, and so on.

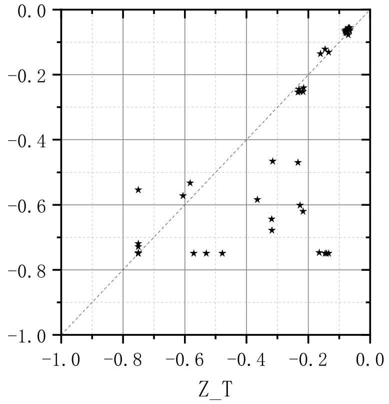

Figure 2: The relationship between Z and $Z_{\mathcal{T}}$'s distinguishability. 31 out of 45 points (68.9%) lie in $|y - x| < 0.05$.

Algorithm 1 Heuristic Combination Sampling Algorithm

```
 1: Pre-tested set $\widetilde{\mathcal{Z}} := \emptyset$.
 2: Under-test set $\mathcal{Z}_{test} := \{\{i\}|i = 1, \cdots, N\}$
 3: for $w = 1, 2, \cdots, N$ do
 4:     Pre-test the combinations in $\mathcal{Z}_{test}$.
 5:     $\widetilde{\mathcal{Z}} := \widetilde{\mathcal{Z}} \cup \mathcal{Z}_{test}, \quad \widetilde{\mathcal{Z}}_{test} := \emptyset$
 6:     if $w \neq N$ then
 7:         for $Z \in \mathcal{Z}_{test}$ do
 8:             for $i \in 1, 2, \cdots, N \wedge i \notin Z$ do
 9:                 $\widetilde{Z} = Z \cup \{i\}$
10:                 Predict $\widetilde{L}_{\widetilde{Z},D}$ according to Assumption 1
11:                 $\widetilde{\mathcal{Z}}_{test} := \widetilde{\mathcal{Z}}_{test} \cup \{\widetilde{Z}\}$
12:             end for
13:         end for
14:         $\mathcal{Z}_{test}$ := up to $N - w$ elements in $\widetilde{\mathcal{Z}}_{test}$ with lowest $\widetilde{L}_{\widetilde{Z},D}$
15:     end if
16: end for
17: return $M$ elements in $\widetilde{\mathcal{Z}}$ with lowest $L_{Z,D}$
```

4.3 Moving-averaged Weighted Voting

In this section we introduce our scheme of weighted voting. We first extend DAC theory, which links voting weights θ to classification error, before introducing methods to decide θ.

Theoretical Foundation of Weighted Voting

Theorem 1 proves that DAC works for a single embedding space in unsupervised domain adaption, while it remains unexplored for domain adaption in multiple embedding spaces. We hence extend it towards the scenario where multiple embedding spaces exist.

Theorem 2 (Multi-space Domain Adaption)**.** *The upper-bound of prediction error in target labels can be reduced by minimizing the weighted sum of DAC loss in all embedding* *spaces: (λ is treated as a constant [Chen* et al.*, 2019])*

$$\epsilon_T \leq \sum_j \theta_j(\epsilon_{j,S}(h_j) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{M}_{j,S}, \mathcal{M}_{j,T})) + \lambda \quad (9)$$

Proof. *While the optimization objective $L = \sum_j \theta_j L_{Z_j}$ is a weighted sum of DAC loss in individual combinations, instead of loss on the assembled final label, we find that due to a consistent weight $\boldsymbol{\theta}$, the prediction error is upper bounded by the objective L through Jensen Inequality and Theorem 1.*

$$\begin{aligned}\epsilon_T &= -\sum_{i \in \mathcal{V}_1} y_i \log(\sum_j \theta_j \hat{y}_{i,j}) \\ &\leq \sum_j \theta_j(-\sum_{i \in \mathcal{V}_1} y_i \log(\hat{y}_{i,j})) \\ &= \sum_j \theta_j \epsilon_{j,T}(h_j) \\ &\leq \sum_j \theta_j(\epsilon_{j,S}(h_j) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{M}_{j,S}, \mathcal{M}_{j,T})) + \lambda\end{aligned}$$

Moving-averaged Strategy

As Theorem 2 draws the connection between the final prediction error and errors on individual embedding spaces, we hence focus on the appropriate choice of θ. To obtain voting vector $\boldsymbol{\theta}$ corresponding to the performance of each channel, we first calculate $\widetilde{\boldsymbol{\theta}}$ by their loss values: [Zhu *et al.*, 2017]

$$\beta_j = -\eta L_{Z_j} \qquad \widetilde{\theta}_i = \frac{\exp \beta_j}{\sum_i \exp \beta_i} \quad (10)$$

where η is a hyper-parameter. The higher η is, the larger disparity among $\widetilde{\theta}_j$ will be. However, directly using $\boldsymbol{\theta} = \widetilde{\boldsymbol{\theta}}$ may cause **Weight Dominance**, where a converged combination illustrates a much lower loss and obtains overwhelming voting power, stifling the other combinations that may be potentially helpful. To address this problem, the voting power θ is moving-averaged here to avoid the abrupt change of $\boldsymbol{\theta}$ and hence assure that every combination will have adequate gradients to converge in early training periods. At the end of each iteration, we update $\boldsymbol{\theta}$ with $\boldsymbol{\theta} \leftarrow \alpha\boldsymbol{\theta} + (1-\alpha)\widetilde{\boldsymbol{\theta}}$. Note that $0 < \alpha < 1$ and θ_j is originally set to $1/M$.

5 Experiments

5.1 Experiment Setup

Datasets

We sample pairs of structurally different graphs respectively from **ACM** [Kong *et al.*, 2012], **AMiner** and **DBLP** [Wang *et al.*, 2019]. The statistics of these datasets are presented in Tab.1, hence showing their structural difference. For each pair of graphs, the density of meta-path edges is quite different between each other, which shows that these graph pairs are domain compatible (i.e. structurally dissimilar).[1]

Dataset	Graph A		Graph B	
	Nodes	Edges	Nodes	Edges
ACM	1500	4,960	1500	759
		6,691		3,996
		26,748		75,180
AMiner	1500	4,360	1500	462
		554		3,740
		89,274		67,116
DBLP	1496	2,602	1496	3,460
		673,730		744,994
		977,348		1,068,250

Table 1: Datasets used in experiments

Baselines

We select some state-of-the-art baselines to verify the ability of MuSDAC to handle transferable classification on HINs. For baselines working on homogeneous networks, we merge the adjacency matrices of all meta-paths into one adjacency matrix, hence obtaining a unified network.

- **GCN** [Kipf and Welling, 2016]: a typical graph neural network designed for homogeneous networks.
- **GraphInception** [Zhang *et al.*, 2018]: a deep GCN for collective classification on HINs.
- **HAN** [Wang *et al.*, 2019]: a heterogeneous graph embedding method based on hierarchical attention.
- **DANE** [Zhang *et al.*, 2019]: It adopts domain adaption to learning transferable embeddings on different homogeneous networks.
- **MuSC**: a variant of MuSDAC, which removes MMD loss in Eq.4.
- **SingleDAC**: a variant of MuSDAC, which uses only one combination with the best distinguishability.
- **MuSDAC-GAN**: a MuSDAC variant applying GAN and average voting instead of MMD and weighted voting.

Hyper-parameter Settings

We use default parameter settings for GCN, GraphInception, HAN and DANE. In MuSDAC and its variants, the dimensionality of the first and second hidden layers of the multi-channel GCN is 64 and 32 respectively, before aggregated to 16 in the aggregated channel. The number of sampled combinations $|\mathcal{Z}| = M = 2N - 1$. In DAC, we use 5 Gaussian kernels for MMD and $\gamma = 10$. In weighted voting, we take $\eta = -25$ and $\alpha = 0.95$.

5.2 Classification Results

The results of classification with different transfer settings are shown in Tab. 2, where we are able to draw four observations.

1. Compared to GraphInception, HAN, MuSC which neglect transfer learning, MuSDAC achieves improvement especially in ACM B→A and AMiner A→B where a significant difference in meta-path density between the graph pair is observed (as shown in Table 1).

[1] The explicit description of datasets is on https://github.com/PKUterran/MuSDAC/blob/master/data/DATA.md.

Accuracy	ACM		AMiner		DBLP		AVG
	A→B	B→A	A→B	B→A	A→B	B→A	
GCN	0.701	0.574	0.476	0.750	0.428	0.363	0.549
GraphInception	0.593	0.534	0.539	0.603	0.709	0.700	0.613
HAN	0.683	0.682	0.695	0.663	0.808	0.761	0.715
DANE	0.724	0.602	0.701	0.795	0.672	0.636	0.688
MuSC	0.688	0.757	0.651	0.776	0.804	0.769	0.741
SingleDAC	0.717	0.769	0.751	0.810	0.808	0.787	0.775
MuSDAC-GAN	0.688	0.771	0.754	**0.817**	0.808	0.783	0.770
MuSDAC	**0.726**	**0.790**	**0.755**	0.799	**0.817**	**0.798**	**0.781**

Table 2: Experiment results of transferable node classification

2. Compared to DANE which deals with homogeneous networks, a significant improvement by MuSDAC is observed, especially in DBLP where heterogeneity is most prominent (as shown by denser meta-paths).
3. MuSDAC is able to recover more complex semantics than SingleDAC via multiple channel combinations, hence performing better.
4. MuSDAC-GAN fails to outperform MuSDAC as MMD is able to more precisely reflect domain distances while GAN losses cannot, hence facilitating the use of weighted voting.

5.3 Analysis of Two-level Selection

Accuracy	ACM		AMiner	
	A→B	B→A	A→B	B→A
Random+Average	0.702	0.762	0.726	0.681
Random+Moving	0.717	0.786	0.732	0.793
Heuristic+Average	0.717	**0.794**	0.746	0.775
Heuristic+Weighted	0.707	0.782	0.730	**0.806**
Heuristic+Moving	**0.726**	0.790	**0.755**	0.799

Table 3: Two-level Selection Analysis

In this section, we analyze two-level selection strategy from the following perspectives.

- Selection: whether to randomly select M combinations (Random) or use heuristic algorithm (Heuristic).
- Voting: whether to use average voting (Average), weighted voting (Weighted) or moving-averaged weighted voting (Moving).

The results are shown in Table 3 which shows that two-level selection strategy is indeed able to significantly improve the performance compared to simpler counterparts, especially when they are used simultaneously.

5.4 Hyper-parameter Sensitivity

We test the sensitivity of γ in Eq.4 on AMiner A→B and plot the results in Fig.3(a). It can be shown that as long as regularization is used ($\gamma > 0$), no significant change in performance is observed.

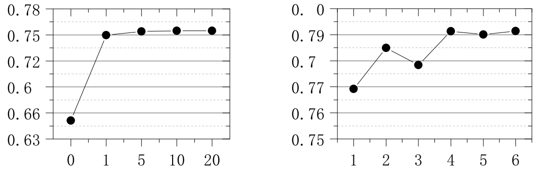

Figure 3: Hyper-parameter Sensitivity of γ (left) and combination number when $N = 3$ (right)

We also vary the number of combinations to sample $M = 1$ to 6 (with $N = 3$) on ACM B→A and plot the results in Fig. 3(b). We conclude that better performance is attained as we sample more combinations and levels of semantics.[1]

5.5 Visualization

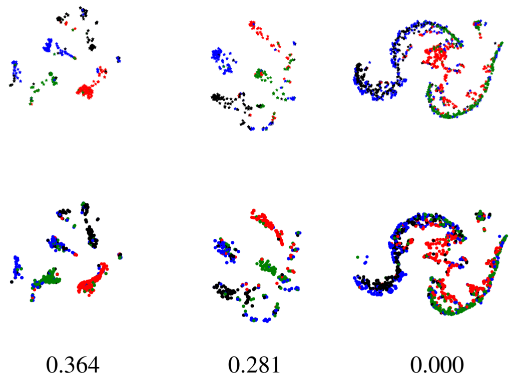

Figure 4: Visualization of AMiner A→B generated by MuSDAC. Two graphs in the same column refer to the source and target embedding distribution in an aggregated channel, whose final voting power is listed beneath the graphs.

We visualize 3 out of 5 aggregated channels on AMiner A→B in Fig.4 to provide an intuitive understanding about our voting scheme. It can be shown that the first two channels with high voting power perform well in that they illus-

[1] More experimental materials can be found on https://github.com/PKUterran/MuSDAC/tree/master/result.

trate both clear boundaries among categories, and similar embedding distributions. Contrarily, the last channel is hardly effective in voting due to its blurred boundaries. The result qualitatively shows that MuSDAC is able to pick indicative combinations for transferable classification task.

6 Conclusion

We propose MuSDAC featuring multi-space architecture and two-level selection, which succeeds in solving the problems underlying in domain adaptive classification on HINs. Compared to various baselines, MuSDAC shows promising performance from both prediction accuracy and visualization.

Acknowledgments

This work was supported by the National Natural Science Foundation of China. Besides, we are thankful to Yizhou Zhang for his helpful suggestions.

References

[Ben-David *et al.*, 2010] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

[Chen *et al.*, 2019] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, pages 1081–1090, 2019.

[Dong *et al.*, 2017] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, pages 135–144, 2017.

[Dziugaite *et al.*, 2015] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, pages 258–267, 2015.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[Gui *et al.*, 2016] Huan Gui, Jialu Liu, Fangbo Tao, Meng Jiang, Brandon Norick, and Jiawei Han. Large-scale embedding learning in heterogeneous event data. In *ICDM*, pages 907–912, 2016.

[Hosseini *et al.*, 2018] Anahita Hosseini, Ting Chen, Wenjun Wu, Yizhou Sun, and Majid Sarrafzadeh. Heteromed: Heterogeneous information network for medical diagnosis. In *CIKM*, pages 763–772, 2018.

[Jin *et al.*, 2020] Lichen Jin, Yizhou Zhang, Guojie Song, and Yilun Jin. Active domain transfer on network embedding. In *WWW*, pages 2683–2689, 2020.

[Kipf and Welling, 2016] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907, 2016.

[Kong *et al.*, 2012] Xiangnan Kong, Philip S. Yu, Ying Ding, and David J. Wild. Meta path-based collective classification in heterogeneous information networks. In *CIKM*, pages 1567–1571, 2012.

[Lin *et al.*, 2019] Yucheng Lin, Xiaoqing Yang, Zang Li, and Jieping Ye. Ahine: Adaptive heterogeneous information network embedding. *ArXiv*, abs/1909.01087, 2019.

[Long *et al.*, 2015] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[Long *et al.*, 2018] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1640–1650, 2018.

[Mansour *et al.*, 2009] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009.

[Rafailidis and Weiss, 2019] Dimitrios Rafailidis and Gerhard Weiss. Adaptive deep learning of cross-domain loss in collaborative filtering. *ArXiv*, abs/1907.01645, 2019.

[Rozantsev *et al.*, 2018] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. In *PAMI*, pages 801–814, 2018.

[Shi *et al.*, 2016] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. A survey of heterogeneous information network analysis. In *TKDE*, pages 17–37, 2016.

[Shi *et al.*, 2018] Yu Shi, Qi Zhu, Fang Guo, Chao Zhang, and Jiawei Han. Easing embedding learning by comprehensive transcription of heterogeneous information networks. In *KDD*, pages 2190–2199, 2018.

[Sun *et al.*, 2011] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, pages 992–1003, 2011.

[Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *WWW*, pages 2022–2032, 2019.

[Yang *et al.*, 2019] Carl Yang, Jieyu Zhang, and Jiawei Han. Neural embedding propagation on heterogeneous networks. In *ICDM*, pages 698–707, 2019.

[Zhang *et al.*, 2018] Yizhou Zhang, Yun Xiong, Xiangnan Kong, Shanshan Li, Jinhong Mi, and Yangyong Zhu. Deep collective classification in heterogeneous information networks. In *WWW*, pages 399–408, 2018.

[Zhang *et al.*, 2019] Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. Dane: Domain adaptive network embedding. In *IJCAI*, pages 4362–4368, 2019.

[Zhu *et al.*, 2017] Yi Zhu, Shawn D. Newsam, and Zaikun Xu. UC merced submission to the activitynet challenge 2016. *ArXiv*, abs/1704.03503, 2017.