Graph-Guided Unsupervised Clustering for Source-Free Domain Adaptation

Jing Wang
jing@ece.ubc.ca

Mingyuan Du
dukedu98@gmail.com

Shurui Feng
surifeng@student.ubc.ca

Weiye Liang
liangwy9@student.ubc.ca

Abstract

Domain adaptation aims at transferring knowledge learned from one labeled dataset (*source domain*) to another unlabeled dataset (*target domain*) by learning a shared representation. While under its source-free settings (SFDA), the source data are not accessible when adapting the learned representation, i.e., the source pre-trained model, to the target domain. Because of the absence of source data, an analytical estimation of the domain shift is not likely to be derived directly. The existing methods either treat SFDA either as an unsupervised clustering problem on those unlabeled target data or leverage knowledge distillation to transfer knowledge between two models. Neither of them explicitly considers how to guide the adaptation in the absence of source data. In response to this, we propose two methods based on the similarity graph built upon latent features to perform neighborhood searching for unsupervised clustering. The proposed learning frameworks start with a feature graph whose node and edge are represented by each data sample and its feature similarity with other data samples, respectively. Two methods are developed based on the built feature graph, 1) a contrastive objective that transforms the target feature space into a discriminative one under the guidance of the feature graph; 2) a distribution alignment between a data sample's classification logits and node embeddings to encourage a prediction consistency between the classification network and graph encoder. Extensive experiments demonstrate that our proposed method achieves state-of-the-art performance on multiple DA benchmarks even compared with the traditional domain adaptation methods which have access to the source data.

1 Introduction

Leveraging massive amounts of carefully labeled data, supervised learning has achieved great success in training machines to mimic human behaviors, e.g., manipulation [Mnih et al., 2015], recognition [Russakovsky et al., 2015], and understanding [Fawzi et al., 2022]. Nevertheless, most supervised models assume that the training and testing data samples will be drawn from the same data distribution, which does not hold in many practical applications. Due to the existence of domain shifts, the performance will degrade significantly when directly testing the models trained on one data domain (e.g., computer-generated images) on another (e.g., real images taken by cameras). Domain adaptation (DA) methods propose to solve this problem by learning a representation that is shared by the two data domains and is invariant to the change in data distribution [Ganin and Lempitsky, 2014]. One way to learn such a shared representation is to align the latent feature distributions of the two domains [Wang et al., 2021]. In this project, we will consider a more challenging but realistic case, i.e., source-free domain adaptation (SFDA), where the shared representation is expected to

Preprint. Under review.

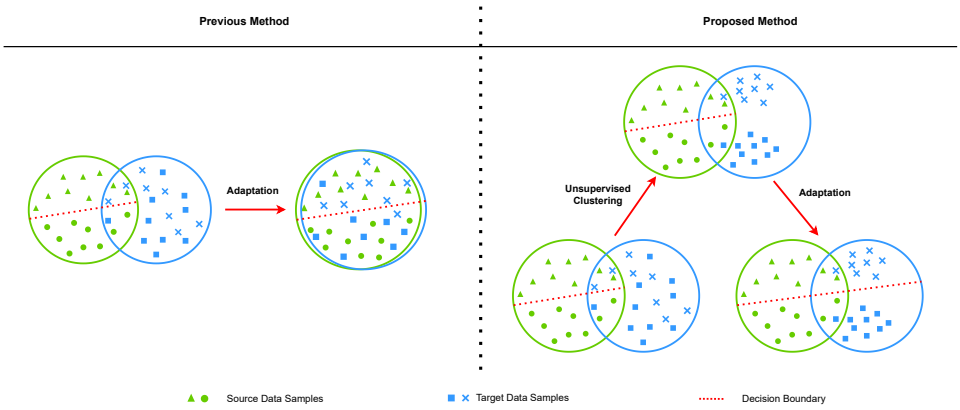

Figure 1: **Our Novelty.** Illustration of solving SFDA by our graph-guided unsupervised clustering. Unlike previous methods that attempt to align the predictions of the source and target models directly, we propose to first perform an unsupervised clustering using neighborhood searching on the unlabeled target domain. Then, the decision boundaries on those discriminative target clusters could be extended from the source model using a feature graph.

be learned in the absence of the labeled source data. Because we have no access to the source data during adaptation, the domain shifts are not likely to be analytically estimated in the SFDA setting.

The existing SFDA methods either treat the model pre-trained on the source domain as initialization to perform unsupervised clustering on those unlabeled target data or solve the problem by directly aligning the output probabilities from the source and target models. Source HypOthesis Transfer (SHOT) [Liang et al., 2020] assumes that the fixed source classifier could retain the information about the source features and introduces an alignment method to maximize such information in the learned target features. To avoid wrong matching of the target data, SHOT also utilizes a neighborhood searching method, i.e., a weighted k-means clustering, to provide pseudo labels for the alignment. G-SFDA proposes a local structure clustering based on k-nearest neighbors (kNNs), aiming to cluster the target features with their semantically similar neighbors, by encouraging consistency between the data sample prediction and its kNN's prediction using Kullback–Leibler (KL)-divergence.

In this report, we propose to solve the SFDA problem using a feature graph in two ways: 1) discriminative feature clustering and 2) network prediction alignment. As shown in Figure 1, unlike the existing works [Yang et al., 2021, Liang et al., 2020] that transform the unsupervised clustering problem into the neighborhood searching problem, we propose to perform such discriminative clustering under the guidance of neighborhood searching, which is motivated by the fact that the contrastive-based clustering could be explicitly guided by the definitions of its positive and negative keys. Meanwhile, we also propose a graph regularization method to align the outputs of the classification network with the node embeddings of the graph encoder to allow more interpretability. Our proposed two methods are extensively evaluated on three benchmark datasets on DA: Office-31 [Saenko et al., 2010], Office-Home [Venkateswara et al., 2017], and VisDA-C [Peng et al., 2017]. The experiment shows that our graph-guided contrastive framework outperforms the existing SFDA methods and becomes the new state-of-the-art in SFDA with a lift of 5.6% on VisDA-C dataset.

Contributions:

- We propose a feature graph built on the unlabeled target domain, whose nodes are the target data samples and edges are the cosine distance between two nodes.
- We propose a novel graph-guided contrastive learning, whose positive key is defined as the nearby node of the query sample found in the feature similarity graph, for source-free domain adaptation.
- We also propose another framework to directly align the logit distribution of the classification network with node embeddings from the graph encoder.

2 Related Work

2.1 Source-Free Domain Adaptation

In the traditional domain adaptation setting, the source data are required to evaluate the domain shifts across the two domains [Ben-David et al., 2010]. Due to data privacy concerns, in most applications at the industrial deployment stage, only the model pre-trained on the source domain is available. Thus, a more realistic but challenging source-free domain adaptation problem arises lately, where only the source pre-trained model and the raw target data are available during adaptation [Yang et al., 2021]. A way to solve SFDA is to use adversarial learning [Xia et al., 2021], where a minimax game is set between the feature extractor and the task-specific classifier. By training both networks in an adversarial manner, the feature extractor can learn to extract features that are informative for classifying those unlabeled target data. BAIT [Yang et al., 2020] extends the maximum classifier difference by introducing a second classifier to play the minimax game and modifying the decision boundaries based on the learned target feature clusters. SHOT [Liang et al., 2020] proposes to freeze the source classifier as an anchor while fine-tuning the source feature extractor to maximize the mutual information between the latent feature distribution and the output probability prediction. The idea that SFDA can be treated as an unsupervised clustering problem gives us two insights:

- the feature similarities of the two domains can be naturally formed as a feature graph, in which data samples with more similar features connect closer to each other;
- the inherent connection between self-supervised learning and SFDA enables us to leverage the contrastive methods to such unsupervised clustering problems.

2.2 Contrastive learning

Contrastive learning is a powerful framework to train models to recognize and differentiate between distinct, which can be applied in both supervised (e.g., to ameliorate feature clusters [Khosla et al., 2020]) and unsupervised settings (e.g., to form discriminative feature clusters [Liu et al., 2021]). In the literature review, we limit our attention to its applications in unsupervised learning to be consistent with our proposal. Contrastive Predictive Coding (CPC) [van den Oord et al., 2018] encodes high-dimensional data into an autoregressive sequence of one-dimensional points and maximizes the consistency between a point and its earliest point in the future. Momentum Contrast (MOCO) [He et al., 2019] uses the concept of momentum, i.e., a measure of the movement of a system to continue in its current state, to improve the performance of contrast. The idea behind Momentum Contrast is to use the model's previous predictions as a reference point and to update the model's parameters based on how well it predicts the current data sample compared to its previous predictions. Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [Chen et al., 2020] simply maximizes the consistency between a query data sample and its augmentations in a contrastive manner. Our contrastive learning framework is built on SimCLR because of its simplicity and efficacy.

2.3 Graph Representation Learning

Graph representation learning can generate low-dimension embeddings for nodes and edges of the graph structural data by leveraging neighbor information. Shallow embedding methods, such as node2vec [Grover and Leskovec, 2016] and Deepwalk [Perozzi et al., 2014], use probabilistic approaches to optimize embeddings to encode the statistics of random walks, decode pairwise node similarities in the original graph, and generate node embedding lookup. Specifically, node2vec [Grover and Leskovec, 2016] proposes a controllable sampling method to maximize the likelihood of keeping the neighborhood information for the target node. Deep neural graph representation uses neighborhood autoencoder [Kipf and Welling, 2016a], neighborhood aggregation [Scarselli et al., 2008], and graph convolutional encoder [Kipf and Welling, 2016b] to generate informative feature vectors with both local and global neighbor information. Those general graph encoders can compress the node's neighborhood information (a node's similarities to all other nodes) into a low-dimensional vector.

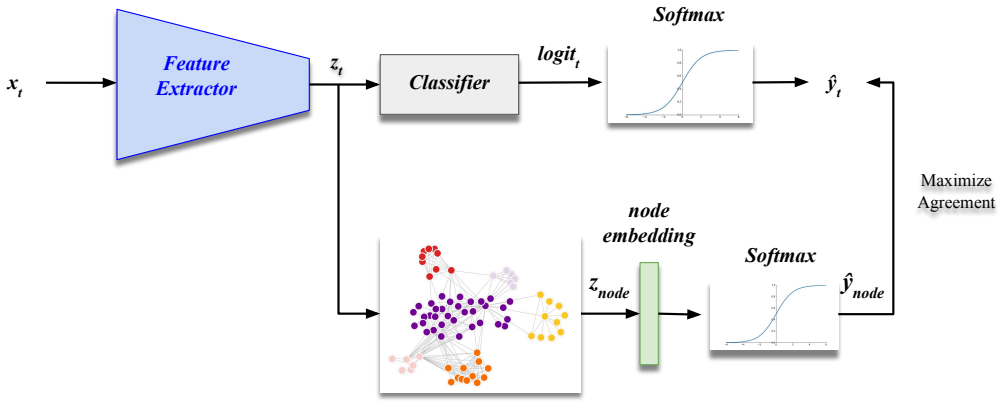

Figure 2: **Overall idea.** Illustration of our general idea to solve the source-free domain adaptation problem by consistency maximization.

3 Problem Formulation

In source-free domain adaptation, we have two training stages: source pretraining (*procurement stage*) and target adaptation (*deployment stage*), which is identical to the current industrial protocols where only pre-trained models will be released for a specific task.

Procurement stage: in this stage, N raw images and their corresponding labels are sampled from the source space $\{X_S, Y_S\}$ to form the source domain $\mathfrak{D}_S = \{(\mathbf{x_s}^{(i)}, \mathbf{y_s}^{(i)})\}_{i=1}^{N}$. The goal of this stage is to pre-train the model on the source domain so that we do not need to access source data in the next stage. We follow the standard pipeline to optimize a cross-entropy objective based on labeled source data for this training stage:

$$\mathcal{L}_{CE}(X_S, Y_S) = -\frac{1}{N}\sum_{i=1}^{N} I(i = \mathbf{y_s}) \log F(G(\mathbf{x_s})), \tag{1}$$

where $I(i = \mathbf{y_s})$ is the binary indicator which outputs 1 when i equals $\mathbf{y_s}$; G is the feature extractor to encode the high-dimensional data into a low-dimensional latent feature vector; F is the task-specific classifier that transforms the latent feature vector into a class probability distribution.

Deployment stage: after obtaining a model pre-trained on the source domain, i.e., $f_s = F_s(G_s(\cdot))$, we sample M raw images from the target space $\{X_T, Y_T\}$ to form the target domain $\mathfrak{D}_T = \{(\mathbf{x_t}^{(i)})\}_{i=1}^{M}$. In this stage, the model is expected to be updated in a way that it could ultimately adapt to the target domain and accurately predict those unlabeled images, i.e., $f_t(\mathbf{x_t}^{(i)}) \approx \mathbf{y_t}^{(i)}$, by leveraging the knowledge learned by the source pre-trained model f_s.

4 Method

In this section, we present two methods: 1) contrastive-based framework and 2) distribution alignment framework to maximize the consistency between classification network prediction and graph prediction, which is shown in Figure 2. The details of our two frameworks are shown in Appendix 8.3. We first introduce the feature graph construction process in our work and then describe our two frameworks built upon it and their components in detail.

4.1 Feature Graph Construction

To make our framework take advantage of graph representation learning, we construct a feature graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by using normalized extracted features of target domain samples $\{G_s(\mathbf{x_t}^{(i)})\}_{i=1}^{N}$ as

features of graph nodes. Edges are the cosine distance matrix of those nodes, and the edge connects any pair of samples representing the similarity of the pair, which can be captured by $e_{ij} = \mathcal{V}_i \mathcal{V}_j^T$. The larger e_{ij} denotes that the i-th and j-th samples belong to the same category with higher probability, and vice versa. We assume that the target samples belonging to the same class should have similar features in the latent feature space even though the source model cannot classify them correctly due to the domain variance.

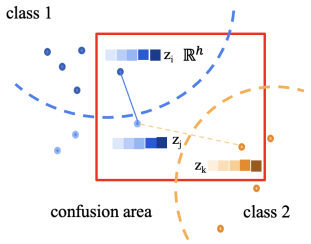

Figure 3: **Motivation of Feature Graph.** Illustration of the class clustering assumption.

To further reduce the complexity of graph construction, we introduce a time-variant threshold to cut edges between dissimilar nodes and emphasize similar nodes. All edges are divided into three categories: very similar, similar and dissimilar based on the time-variant upper and lower similarity bounds which are defined as follows:

$$\begin{cases} \ell(t) = \ell_0 + \lambda_\ell * t \\ 0 \leq \ell(t) \leq 0.9 \end{cases} \quad (2) \qquad e_{ij} = \begin{cases} 0, e_{ij} < \ell(t) \\ 1, \text{otherwise} \end{cases} \quad (3)$$

where both bounds are linear functions of iteration times t, and λ_μ and λ_ℓ control the increasing of bounds. We assume the samples from the same class will be more similar to each other during training time, so increasing the bound can be more strict with the definition of similar node pairs.

4.2 Feature Graph Guided Contrastive Learning

Contrastive learning is a promising method as explicit knowledge could be imposed to guide the formation of discriminative clusters by designing its positive and negative keys. The general idea for contrastive-based clustering is to push the latent features of the query sample close to those of positive keys and away from those of negative keys [Jaiswal et al., 2021]. In our work, the "*nearest*" neighbor of the query sample found in the feature graph, i.e., the node that is closely linked with the query sample node, will be used as the positive key, and the other samples in the same mini-batch will be regarded as the negative keys. Fortunately, as the target domain shares the same classification task with the source domain, i.e., the number of classes to be classified is known during adaptation, we can formulate the contrastive objective in the output probability space instead of the latent feature space to reduce the complexity of the optimization. The positive sample will then be the softmax output of the node embedding:

$$\mathcal{L}_{i,j} = -log \frac{exp(\frac{y_i^T y_j^{graph}}{\tau y_i y_j^{graph}})}{\sum_{k=1}^{2N} 1_{[k \neq i]} exp(\frac{y_i^T y_k}{\tau y_i y_k})}, \quad (4)$$

where y_j^{graph} is the softmax embedding of a node in the feature graph that is closely connected with the query sample node.

$$y_j^{graph} = \frac{exp(h_i)}{\sum_{i=j}^{C} exp(h_j)}, \quad (5)$$

where h_j is the node embedding of the positive key and C is the total number of classes to be classified.

4.3 Aligning Classification Logits with Node Embeddings

Based on the assumption that data samples from the same category should be geometrically close to each other in their latent feature space, thus their node embeddings encoded from the feature graph should also be similar to each other. To introduce more interpretability to the unsupervised clustering, we propose to minimize the mean squared error (MSE) between the output logits z_{x_t} and its corresponding node embeddings $ENC(s_{x_t})$. The reconstructed node embeddings will then be decoded by a fixed-weight classifier to produce the corresponding prediction.

$$DEC(ENC(s_{x_t})) = DEC(z_{x_t}) \approx s_{x_t} \tag{6}$$

$$\mathcal{L}_{MSE} = -\mathbb{E}_{x_t \in \mathcal{X}_t} ||ENC(s_{x_t}) - z_{x_t}||_2^2 \tag{7}$$

where $x_t \in \mathcal{X}_t$ and the s_{x_t} vector contains a node x_t's similarity with all other nodes in the graph and acts as a low-dimensional vector representation of x_t's neighborhood.

Graph convolutional networks (GCN) [Welling and Kipf, 2016, Kipf and Welling, 2016a, Schlichtkrull et al., 2018, Berg et al., 2017] and GraphSAGE [Abadi et al., 2016] use neighborhood aggregation methods to build the representation for a node in an iterative fashion, which could incorporate the information about both built nodes and their edges to generate meaningful node embeddings. Since such learned node embeddings will aggregate information from the built feature graph, ablation studies are performed on the way to constructing a feature graph - whether to use features or logits to build node features and similarity edges.

5 Experiments

5.1 Dataset

We evaluated our model on three different close-set benchmark datasets for DA: Office-31 [Saenko et al., 2010], Office-Home [Venkateswara et al., 2017], and VisDA-C [Peng et al., 2017]. All of them are image classification tasks, and the evaluation metrics would be classification accuracy in percentage. Office-31 has three domains: *Amazon*, *DLSR*, and *Webcam*, with 31 categories of images; medium-sized Office-Home has four domains: *Art*, *Clipart*, *Product*, and *RealWorld*, with 65 categories of images and consists of over 15,500 images; VisDA-C has two domains: *synthetic* and *real-world*, 12 categories of images. These datasets would be able to provide sufficient evaluation for our model from different scales.

The model is pre-trained on the source domain and adapted to the target domain. Taking the adaptation scenario from *Amazon* to *DLSR* as an example, we will train the model with the labeled data of *Amazon* with the standard *CrossEntropy* objective, we can get a *source-only* model (before adaptation); such pre-trained model will be used as the starting point for training in *DLSR*. We used Colab and local Nvidia 3090Ti for training the models. Due to the limited computing resources and time, we finished all experiments on Office-31 and most experiments on Office-Home and VisDA-C.

5.2 Baselines and Setup

We used SHOT [Liang et al., 2020] and G-SFDA [Yang et al., 2021] as our baselines. The mappings of our models $f = NetC(NetB(NetF(\cdot)))$ can be divided into three parts based on their functionalities: the feature extractor ($NetF$), the bottleneck of the classifier ($NetB$), and the last linear layer of the classifier ($NetC$). During adaptation, the parameters of $NetF$ and $NetB$ will be updated, while the parameters of $NetC$ are fixed. Due to the imbalanced data sample distribution on different categories observed in the three benchmark datasets, we follow the baselines on SFDA which incorporate the imbalanced entropy (IE) [Gong et al., 2012] into their overall loss function:

$$\mathcal{L}_{ent}(f_t; \mathcal{X}_t) = -\mathbb{E}_{x_t \in \mathcal{X}_t} \sum_{k=1}^{K} \delta_k(f_t(x_t)) log \delta_k(f_t(x_t)) \tag{8}$$

$$\begin{aligned} \mathcal{L}_{div}(f_t; \mathcal{X}_t) &= \sum_{k=1}^{K} \hat{p}_k * log \hat{p}_k \\ &= D_{KL}(\hat{p}_k, \frac{1}{K}\boldsymbol{I}_K - logK) \end{aligned} \tag{9}$$

$$\mathcal{L}_{IE} = \mathcal{L}_{ent} + \mathcal{L}_{div} \tag{10}$$

where δ is the softmax normalization; f_t is the model fine-tuned on the target domain; $\mathbf{1}_K$ is a K-dimensional vector with all ones and $\hat{p}_k = \mathbb{E}_{x_t \in \mathcal{X}_t}[\delta f_t^{(k)}(x_t)]$ represents the expectation of classifier's outputs of all target samples. We have reproduced the results of SHOT on Office-31, Office-Home, and VisDA-C (table 1-3). We use ResNet-50 and ResNet-101 as $NetF$ for the experiments on Office and VisDA-C datasets, respectively. The feature graph used to model the similarities among data samples is of size $N \times N$ whose parameters will only be built once in one epoch.

5.3 Experiments and Results

In the first experiment, we used graph-selected positive keys for contrastive learning (Code in Appendix 8.1, figure in Appendix 8.3). We update the feature graph in every epoch and use Node2Vec to re-sample walks with the built graph, and generate a fixed lookup for node embeddings. In every iteration, for each sample i in the mini-batch, we use Node2Vec to produce 5 nodes that are closely connected with the query sample and retrieve their node embeddings from lookup. We calculate the mean of selected neighbor node embeddings as the pseudo label. Then we use this super node with averaged embedding as the positive key, and all other samples in the same mini-batch as the negative keys of the query sample i. Contrastive objective is then built upon this to perform neighborhood searching and discriminative clustering simultaneously. We also tried using other unsupervised clustering methods, such as kNNs, to select positive keys to formulating our contrastive objective.

In the second experiment, we evaluated the second framework, where we wish to give output logits of $NetB$ with interpretable meaning as the node embeddings, and used the second layer of the classifier, $NetC$, for the classification head (Code in Appendix 8.1, figure in Appendix 8.3). In this experiment, the node embedding is expected to be aligned with the output logit distribution using MSE loss. Ablation studies include 1) using different distribution spaces (outputs of $NetF$ or $NetB$) to construct the graph (feature graph f.g., or logit graph l.g.), which to be encoded as node embeddings; and 2) generating node embeddings with different graph encoders. Table 1 shows our ablation studies results on different graph encoders with different feature graph settings. In all the experiments performed, weights of graph encoders are updated together with $NetB$ in each iteration. Due to the limited computing resources, we only did ablation studies on the small-size dataset *Office-31*.

Table 1: Accuracy(%) of the proposed framework on *Office-31* (ResNet-50).

	Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg
source only	ResNet-50	78.3	72.0	60.2	91.8	61.6	96.39	76.7
baselines	Imbalanced Entropy (IE)	90.6	90.7	74.2	97.1	72.1	99.8	87.4
	SHOT (IE + K-Means)	92.6	89.2	74.2	96.6	73.8	99.0	87.6
experiment 1	**IE+Node2vec+Contrastive**	93.7	92.2	73.5	97.4	72.7	99.6	88.2
abalation studies	**IE+kNNs+Contrastive**	**94.5**	**94.0**	**76.2**	**98.8**	**75.1**	**100**	**89.8**
	IE + GCN1layer (f.g.)	91.2	92.5	74.9	96.5	72.4	99.6	87.9
experiment 2	**IE + GCN1layer (l.g.)**	91.2	92.1	75.4	96.7	73.2	99.8	88.1
	IE + GCN2layers (f.g.)	89.4	91.8	75.3	96.9	72.6	99.6	87.6
abalation studies	**IE + GCN2layers (l.g.)**	90.4	91.0	75.7	97.6	70.8	99.8	87.6
	IE + GraphSAGE (f.g.)	89.4	91.8	74.5	97.1	73.7	99.6	87.7
	IE + GraphSAGE (l.g.)	90.0	91.7	73.6	97.9	69.8	99.4	87.0

The quantitative results on *Office-Home* and *VisDA-C* are shown in table Table 2 and Table 3. Firstly, we verify the importance of the domain adaptation problem by showing that the source-free domain adaptation methods outperform source-only models significantly. We also evaluated our proposed two methods on *Office-Home*, and our first framework on *VisDA-C*.

Table 2: Accuracy(%) of the proposed framework on *Office-Home* (ResNet-50).

	Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
source only	ResNet-50	44.7	66.1	73.5	53.1	60.6	64.2	50.8	39.5	72.5	64.2	44.4	76.6	59.1
baselines	SHOT (IE+K-Means)	57.0	77.8	**81.4**	**66.9**	**77.6**	78.0	**68.3**	53.0	82.0	**74.0**	**58.0**	83.2	71.4
	G-SFDA (Paper Reported)	**57.9**	**78.6**	81.0	66.7	77.2	77.2	65.6	**56.0**	82.2	72.0	57.8	83.4	71.3
experiment 1	**IE+Node2vec+contrastive**	54.6	76.4	78.9	65.6	74.5	76.1	64.4	54.2	81.7	73.6	57.5	83.3	70.1
	IE+kNNs+contrastive	57.0	78.2	81.2	66.4	76.5	**78.2**	68.0	52.5	**82.5**	73.4	57.8	**84.6**	**71.4**
experiment 2	**IE+GraphSAGE (f.g.)**	53.9	76.3	79.5	65.5	74.2	75.8	65.9	54.5	81.6	73.8	56.7	82.2	70.0

Table 3: Accuracy(%) of the proposed framework on *VisDA-C* (ResNet-101).

	Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
source only	ResNet-101	54.2	19.0	54.2	**79.9**	64.6	4.6	78.0	22.5	73.6	42.2	79.1	6.1	48.2
baselines	SHOT	94.2	56.3	68.1	29.8	88.9	85.2	58.4	73.6	90.8	81.1	77.64	32.3	70.0
	G-SFDA (Paper Reported)	96.1	88.3	85.5	74.1	97.1	95.4	89.5	79.4	95.4	92.9	89.1	42.6	85.4
experiment 1	**IE+Node2vec+Contrastive**	92.2	80.5	79.6	50.9	88.7	59.8	81.9	76.6	84.9	81.4	88.46	**46.3**	75.6
	kNNs+Contrastive	**96.6**	**89.3**	**88.3**	74.9	**97.7**	**94.7**	**83.9**	**81.6**	**94.2**	**87.0**	**90.2**	45.7	**85.3**

5.4 Visualization of Discriminative Feature Clusters

To validate the claim that our proposed method could effectively perform discriminative feature clustering under the guidance of the feature graph, we leverage t-SNE [Van der Maaten and Hinton, 2008] to visualize those latent features. We run the t-SNE visualizations of the adaptation scenarios from *Amazon* to *DLSR* and *Webcam* respectively on the *Office-31* dataset. As shown in Figure 4, our proposed method could form very discriminative feature clusters that any classifier could easily derive decision boundaries among them in both cases.

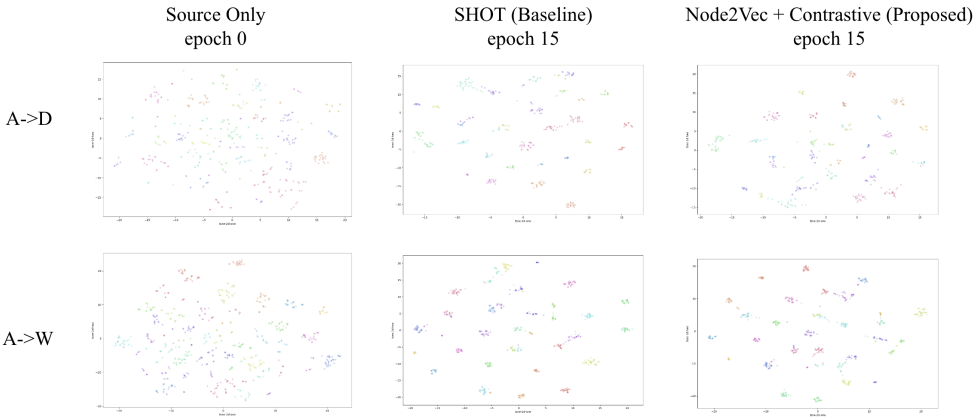

Figure 4: **t-SNE visualization on the feature space of Office-31.**

6 Conclusion

We proposed a few unsupervised classification models that perform well on the unlabeled target domain without access to the labeled source data during adaptation. First, we proposed a feature graph with the dimension $N \times N$ where N is the total number of input samples in the target dataset to provide information on sample-wise similarity. Then we proposed a feature graph-guided contrastive learning framework for source-free domain adaptation, with a contrastive objective optimized on the output probability level with the positive key defined on the embedding of the most similar node of the query sample node. We also proposed a reconstruction-based regularization to compress a sample's neighborhood information gathered from the feature graph into the classification network, which gives more interpretability to the unsupervised clustering. Extensive experiments and ablation studies have been performed to validate the efficacy of our proposed method.

7 Future Work

In the present work, we utilized graph regularization to directly align the classification outputs with the node embeddings from the feature graph. In the future, this graph regularization could be used in a way to learn better similarities among data samples (nodes) so that our feature graph-guided contrastive model could yield competitive results over those of our kNN-based contrastive model. Meanwhile, we could use different graph encoders to provide better node embeddings as the positive keys for contrastive-based clustering. Moreover, we could construct a feature graph of the size $D \times D$ where D is the dimension of the latent feature vector to measure the feature channel correlation. Such a feature graph could be shared by different data domains (as its size is independent of the number of data samples) and could be potentially used to explicitly measure domain shifts in the absence of source data in SFDA settings.

8 Appendix

8.1 Code Repository

Github Files

Code repository - https://github.com/dukedudu/571f_project

Adaptation Framework for Experiment 1 - Contrastive learning on graph selected positive pairs
Github Python file 1 - https://github.com/dukedudu/571f_project/blob/main/image_target_cuda_colab_framework.py

Adaptation Framework for Experiment 2 - Learning graph representations through node embeddings' alignment
Github Python file 2 - https://github.com/dukedudu/571f_project/blob/main/image_target_pyg_cuda.py

Colab Notebooks

Experiment 1 related colab files
colab notebook1 (VisDA-C), Click to access link
colab notebook2 (Office-31 & Office-Home), Click to access link

Experiment 2 related colab files
colab notebook 3, Click to access link

t-SNE Visualization
colab notebook 3, Click to access link

GDrive Link
Click to see: source pre-trained and target domain fine-tuned models

8.2 Contribution

8.2.1 Overall Contribution

Jing Wang: the idea of using feature graph-guided contrastive learning for source-free domain adaptation, the idea of constructing the graph in terms of feature-wise similarity, source-free domain adaptation problem formulation, contrastive learning idea and implementation (Github folder "knn_based"), code framework and troubleshooting, experiment 1 results, and final report writing.

Mingyuan Du: experiment 1 results (Github python fle 1 and colab notebooks 1 & 2) & experiment 2 results, code framework and troubleshooting, feature graph implementation, node2vec idea and implementation, node embedding alignment implementation, time-variant threshold implementation, GitHub, Colab and code framework management, and final report writing.

Shurui Feng: experiment 2 results (Github python file 2 and colab notebook 3), the idea of constructing the graph in terms of sample-wise similarity, feature graph implementation, graph representation learning (node embedding) idea and implementation, time-variant threshold idea, t-SNE visualization, framework and troubleshooting, and final report writing.

Weiye Liang: proposal, presentation, and final report writing.

8.2.2 Report Writing Contribution

Jing Wang: writing: abstract, 1 introduction, 2.1 source-free domain adaptation, 2.2 contrastive learning, 3 problem formulation, 4 methods, 4.2 feature graph guided contrastive learning, 6 conclusions, and 7 future work.

Mingyuan Du: writing: 2.3 graph representation learning and 4.1 feature graph construction, 4.3 aligning classification logits with node embeddings, 5 experiments.

Shurui Feng: writing: abstract, 1 introduction, 2.3 related work on graph representation learning, 4.3 aligning classification logits with node embeddings, 5 experiments.

Weiye Liang: writing: 1 introduction, 2 related works, 2.1 source-free domain adaptation, 2.2 contrastive learning, and 7 future work, review of whole report

8.3 Figures for Our Two Frameworks

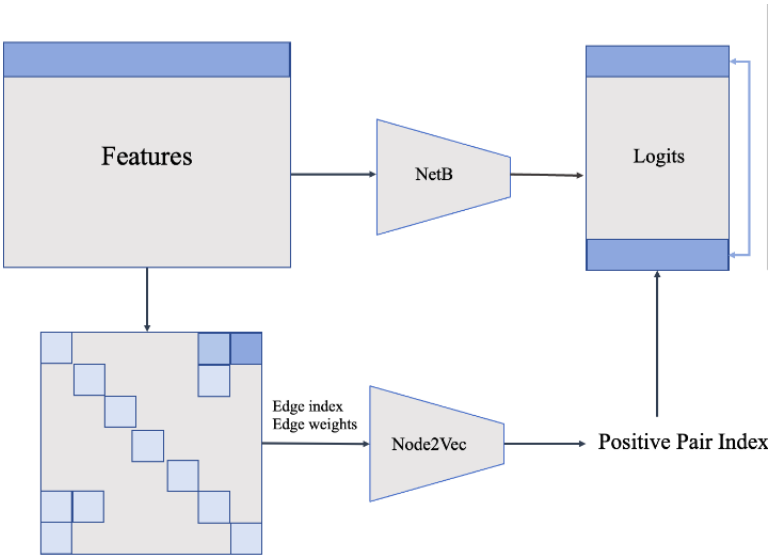

Figure 5: **Experiment 1 Schema - Selected Positive Pairs Contrastive.** Illustration of feature graph guided contrastive learning framework for source-free domain adaptation.

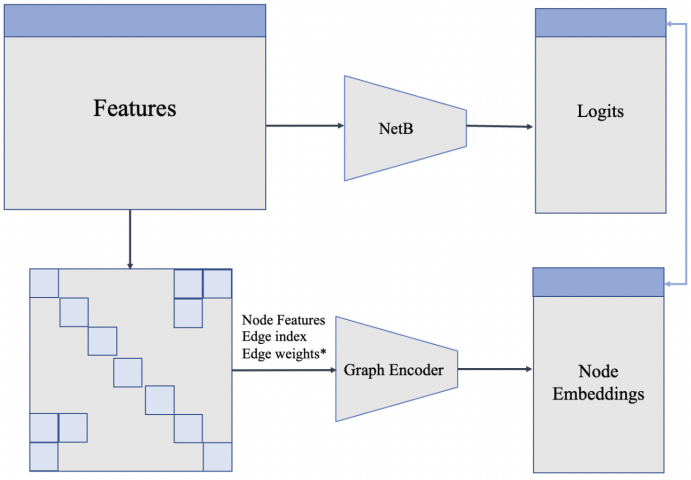

Figure 6: **Experiment 2 Schema - Node Embeddings Learning** Illustration of logits' distribution alignment for source-free domain adaptation.

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

R. v. d. Berg, T. N. Kipf, and M. Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R Ruiz, J. Schrittwieser, G. Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation, 2014.

B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019.

A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021.

P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

T. Kipf and M. Welling. Variational graph auto-encoders. nips workshop on bayesian deep learning, 2016a.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016b.

J. Liang, D. Hu, and J. Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 6028–6039, 2020.

X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko. Visda: The visual domain adaptation challenge. *CoRR*, abs/1710.06924, 2017.

B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, page 213–226, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 364215560X.

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. *CoRR*, abs/1706.07522, 2017.

J. Wang, J. Chen, J. Lin, L. Sigal, and C. W. de Silva. Discriminative feature alignment: Improving transferability of unsupervised domain adaptation by gaussian-guided latent alignment. *Pattern Recognition*, 116:107943, 2021.

M. Welling and T. N. Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

H. Xia, H. Zhao, and Z. Ding. Adaptive adversarial network for source-free domain adaptation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8990–8999, 2021. doi: 10.1109/ICCV48922.2021.00888.

S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui. Casting a bait for offline and online source-free domain adaptation. *arXiv*, 2020.

S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021.