FOCI: Fair Cross-Network Node Classification via Optimal Transport

Anna Stephens, Francisco Santos, Pang-Ning Tan, and Abdol-Hossein Esfahanian

Michigan State University, East Lansing, Michigan, USA 48825
{steph496, santosf3, ptan, esfahanian}@msu.edu

Abstract. Graph neural networks (GNNs) have demonstrated remarkable success in addressing a variety of node classification problems. Cross-network node classification (CNNC) extends the GNN formulation to a multi-network setting, enabling the classification to be performed on an unlabeled target network. However, applying GNNs to a multi-network setting in practice is a challenge due to the possible presence of concept drift and the need to account for link biases in the graph data. In this paper we present `FOCI`, a powerful, model-agnostic approach for cross-network node classification that enables the GNN to overcome the concept drift issue while mitigating potential biases in the data. `FOCI` utilizes a fair Sinkhorn distance function with optimal transport to learn a fair yet effective feature embedding of the nodes in the source graph. We experimentally demonstrate the effectiveness of `FOCI` at addressing the CNNC task while simultaneously mitigating unfairness compared to other baseline methods.

1 Introduction

The proliferation of graph-based data in various application domains has motivated the need to develop more advanced techniques based on deep learning to harness the network data from multiple sources in order to improve the performance of node classification algorithms. Current graph neural network techniques would learn a feature embedding of the nodes from multiple graphs, which are then presented to a fully-connected network layer to perform the final classification. However, despite the notable advances in graph neural networks, there are still numerous challenges that must be addressed in order for the techniques to be successfully deployed to solve real-world problems.

First, existing techniques often assume that the graphs share similar distributional properties, thus enabling us to apply a model trained on one graph, *a.k.a.* the *source graph*, to the nodes in another, *a.k.a.* the *target graph*. Unfortunately, in practice, one would likely encounter some form of distributional shift, where the training data only captures the essence of a particular graph but fails to account for some unforeseen differences in another graph. Such type of concept drift is illustrated in Fig. 1, where the decision boundary induced from the learned representation of a source graph (left) does not reflect the class separation of the nodes in the target graph (right). Second, incorporating fairness into node classification is another concern to prevent the model from generating biased prediction results. Graph neural networks (GNNs) are especially prone to fairness issues due to an artifact of the homophily principle, which states that similar nodes are likely to be connected to each other [14]. Past research on graph fairness have

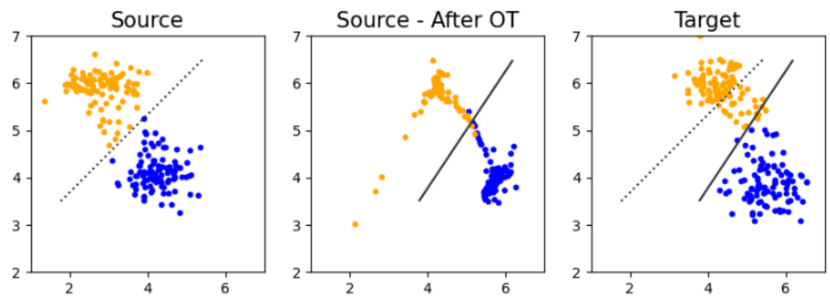


Fig. 1: An illustration of OT for domain adaptation, where the color represents class labels. The diagram on the left is the original source dataset and on the far right is the target dataset. The dotted line is the decision boundary of logistic regression trained on the source dataset. The middle diagram shows the transported source dataset using OT, with the solid line representing the decision boundary obtained by logistic regression.

shown that neighborhood structure is often more dependent on the protected attributes than the classification labels [7][16]. As a consequence, GNNs have the potential to exacerbate unfairness as the learned embeddings may capture more information about a node's protected group than its class label [9], a phenomenon known as *link bias* [6].

To address the concept drift problem, domain adaptation and transfer learning methods have been developed. For graph data, transfer learning is typically studied under the guise of *cross-network node classification* (CNNC) problems [17,18,20], in which there is a source graph, with fully or mostly labeled nodes, and a target graph, which has either a few or no class labels. Existing CNNC methods are primarily model-dependent, tailored towards the specific neural architecture, and thus, limiting their flexibility. Optimal transport (OT) [8,5] is another widely-used transfer learning approach. Specifically, OT would learn a transportation plan that maps data points from a source domain to a target domain in a way that minimizes the total cost [3]. The primary advantage of employing OT lies in its model-agnostic nature, allowing it to be seamlessly integrated into any graph neural network model. The impact of OT becomes evident in the middle diagram of Fig. 1, which shows the decision boundary obtained after applying OT to the source dataset depicted in the left diagram. The modified decision boundary post-OT demonstrates superior alignment with the dataset in the target domain as shown in the right diagram. However, similar to CNNC, there has yet been any studies integrating fairness into OT formulation.

In this paper, we present a novel, model-agnostic framework called `FOCI` (**F**air Cr**O**ss-Network Node **C**lass**I**fication) that performs fair optimal transport while mitigating the concept drift issue in CNNC task. Specifically, `FOCI` considers the nodes' protected attribute information when transporting the source nodes to their corresponding target nodes when learning their feature embedding. We introduce a fair Sinkhorn distance measure for OT to encourage diversity in the mapping of the source nodes to target nodes. This strategy ensures that the learned features are oblivious to the protected attribute information. Finally, we also proposed `FastFOCI`, an extension of `FOCI`, which provides an improved run-time performance.

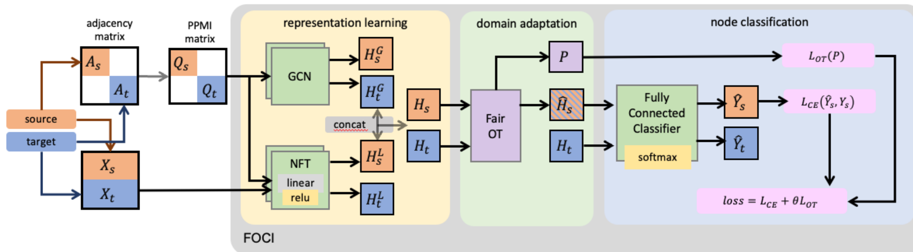


Fig. 2: A schematic illustration of the FOCI framework. The framework contains integrated modules for representation learning (using 2-layer GCN with PPMI matrix along with 2-layer NFT), fair OT for domain adaptation, and a fully connected node classification layer.

2 Problem Statement

Consider a set of graphs, $\mathcal{G}_1, \mathcal{G}_2, \cdots \mathcal{G}_n$, where each $\mathcal{G}_k(V_k, E_k, X_k, Y_k)$ is an attributed graph, with a node set V_k, edge set $E_k \subseteq V_k \times V_k$, node feature matrix X_k, and node label vector Y_k. Let $X_k = X_k^{(p)} \cup X_k^{(u)}$, where $X_k^{(p)}$ corresponds to the set of protected attributes (e.g., gender, race, or age group). Let $A^{(k)}$ denote the adjacency matrix representation of E_k, where $A_{ij}^{(k)} > 0$ if $(v_i, v_j) \in E_k$ and 0 otherwise. Assume the set of graphs are divided into two disjoint groups—source graph $\mathcal{G}_s(V_s, E_s, X_s, Y_s)$, where the node labels in Y_s are known, and a target graph, $\mathcal{G}_t(V_t, E_t, X_t, Y_t)$, where the node labels in Y_t are unknown. The adjacency matrices corresponding to the source and target graphs are denoted as $A_s \in R^{n_s \times n_s}$ and $A_t \in R^{n_t \times n_t}$, respectively, where n_s and n_t are the number of source and target nodes. Assuming the graphs have identical node features and class labels, therefore $X_s \in R^{n_s \times d}$ and $X_t \in R^{n_t \times d}$, $Y_s \in \{0, 1, \cdots, k-1\}^{n_s}$ and $Y_t \in \{0, 1, \cdots, k-1\}^{n_t}$, where k is the number of classes.

Given a source graph, $\mathcal{G}_s = (V_s, E_s, X_s, Y_s)$ and target graph, $\mathcal{G}_t = (V_t, E_t, X_t, Y_t)$, the goal of fair CNNC is to learn a function, $f : V \to \{0, 1, \cdots, k-1\}$, that accurately classifies the labeled nodes in Y_s and the unlabeled nodes in Y_t while minimizing the disparity in classification performance for different groups of the protected attributes.

3 Proposed FOCI Architecture

Fig. 2 shows a high-level overview of `FOCI`, consisting of modules for representation learning, OT for domain adaptation, and fully connected layers for node classification.

3.1 Representation Learning & Pretraining

The GCN approach learns a feature embedding of the nodes in layer $l+1$, $H^{(l+1)}$, using the embedding from its previous layer $H^{(l)}$ as follows:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \tag{1}$$

where $\tilde{A} = A + I$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, and $\sigma(\cdot)$ is the ReLU activation function.

FOCI modifies the GCN architecture [10] to learn a joint embedding of the nodes in the source and target networks. The modifications are needed for two reasons. First, due to oversmoothing effect [13], current GCN cannot be easily extended beyond 2 or 3 layers, thus throttling the effective neighborhood size of its message passing. Because of its limited neighborhood size and the link bias problem noted in the introduction, the learned representation may inadvertantly encode the protected attribute information [6]. To overcome this limitation, we expand the neighborhood utilized by the GCN through the use of the *positive pointwise mutual information* (PPMI) matrix[12]. A PPMI matrix measures the topological proximity of nodes within some K-steps within the network and has been used with other GNN approaches [2,17,20] to expand the neighborhood of interest. By using a PPMI matrix, the embedding learned will more likely capture information from nodes beyond those from the same protected group.

The representation learning module also learns a separate nonlinear embedding of the node features. This allows us to maintain a node embedding that is free of link bias for later use. We refer to this as the *node feature transformation* (NFT) layer, which consists of a fully-connected linear layer with ReLU activation function. We pre-train the network to classify only the labeled nodes in the source network. Specifically, FOCI uses two GCN layers with the PPMI matrix and two NFT layers. The learned node embeddings are concatenated as shown in Fig. 2 before being sent to a fully connected classification layer, which is trained to minimize the following cross-entropy loss: $\mathcal{L}_{CE} = -\frac{1}{n_s}\sum_{x_i \in X_s}\sum_{j=0}^{l} Y_{ij}\log(\hat{Y}_{ij})$

3.2 Fair Optimal Transport

Our strategy to incorporate fairness into OT is by encouraging mappings between members of different protected groups. This heuristic helps to alleviate the link bias problem due to the inherent homophily effect in network data. Let $X_s^{(p)}$ and $X_t^{(p)}$ be the protected attributes of the nodes in the source and target networks, respectively. We first create two sparse matrices, $R \in [0,1]^{n_s \times n_t}$ and $S \in [0,1]^{n_s \times n_t}$, where

$$R_{ij} = \begin{cases} 1 & \text{if } X_{s,i}^{(p)} = X_{t,j}^{(p)} \\ 0 & \text{otherwise} \end{cases}, \quad S_{ij} = \begin{cases} 1 & \text{if } X_{s,i}^{(p)} \neq X_{t,j}^{(p)} \\ 0 & \text{otherwise} \end{cases}$$

Given a transportation plan matrix P, we compute the following fairness loss: $\ell_F = \frac{\langle P,R\rangle_F}{\langle R,R\rangle_F} - \frac{\langle P,S\rangle_F}{\langle S,S\rangle_F}$, where $\langle A,B\rangle_F = \sum_{ij} A_{ij}B_{ij}$ denotes the Frobenius inner product between two matrices. The smaller the fairness loss, the greater the emphasis is on transportation between samples in different protected groups. This enables us to incorporate fairness consideration into OT by introducing the following γ-fair Sinkhorn distance:

$$W_\gamma(\mu_s,\mu_t) = \langle P^\gamma, C\rangle \tag{2}$$

where $P^\gamma = \text{argmin}_{P\in U(\mu_s,\mu_t)}\mathcal{L}_{OT}$ and $\mathcal{L}_{OT} = \langle P,C\rangle - \lambda h(P) + \gamma\left[\frac{\langle P,R\rangle}{\langle R,R\rangle} - \frac{\langle P,S\rangle}{\langle S,S\rangle}\right]$.

Theorem 1. *Given the γ-fair Sinkhorn distance in Eqn.* (2)*, the solution for P^γ is*

$$P^\gamma = diag(u)K diag(v)$$

where $K = e^{-\frac{1}{\lambda}(C+\gamma(\frac{R}{n_1}-\frac{S}{n_2}))}, u = e^{-\frac{1}{2}-\frac{1}{\lambda}\alpha}, v = e^{-\frac{1}{2}-\frac{1}{\lambda}\beta}$.

Algorithm 1 Fair Sinkhorn

Require: $H_s, H_t, X_s^{(p)}, X_t^{(p)}, \lambda, \gamma$
0: $C =$ computeCost(H_s, H_t)
0: $\mu_s, \mu_t =$ computeUniformMarginals(H_s, H_t)
0: $R, S =$ getMasks(g_s, g_t)
0: $K = e^{-\frac{1}{\lambda}(C+\gamma(\frac{R}{n_1}-\frac{S}{n_2}))}$ {Modification}
0: $u =$ ones$(length(\mu_s))/length(\mu_s)$
0: $\tilde{K} =$ diag$(1/\mu_s)K$
0: **while** u changes **do**
0: $\quad u = 1/(\tilde{K}(\mu_t/(K'u))$
0: **end while**
0: $v = \mu_t/(K'u)$
0: $W =$ sum$(u \odot ((K \odot C)v))$ {Distance Measure}
0: $P =$ diag$(u)K$diag(v) {Transport Plan}
0: $\hat{H}_s = n_s P H_t$ =0

Proof. The Lagrangian of the function in Eqn. (2) is

$$\mathcal{L} = \sum_{ij} \left[P_{ij}C_{ij} + \lambda P_{ij} \log P_{ij} + \gamma P_{ij} \left(\frac{R_{ij}}{n_1} - \frac{S_{ij}}{n_2} \right) \right] + \alpha^T(P1_d - \mu_s) + \beta^T(P^T 1_d - \mu_t)$$

where $n_1 = \langle R, R \rangle$ and $n_2 = \langle S, S \rangle$. The solution can be found by taking the derivative of the Lagrangian function with respect to P_{ij} and setting it to zero.

$$P_{ij} = e^{-\frac{1}{2}-\frac{1}{\lambda}\alpha_i} e^{-\frac{1}{\lambda}(c_{ij}+\gamma(\frac{r_{ij}}{n_1}-\frac{s_{ij}}{n_2}))} e^{-\frac{1}{2}-\frac{1}{\lambda}\beta_j}$$

The theorem follows by replacing the terms for K, u, and v into the above equation.

Theorem 1 enables us to compute P^γ using the modified fair Sinkhorn algorithm shown in Algorithm 1, which in turn, allows us to find a fairness-aware transportation plan. These changes can be seen in Algorithm 1 and have no significant impact on computational complexity of existing Sinkhorn algorithm [5]. Additionally, since we are still utilizing squared l_2 loss for our cost and uniform marginals, the transported source node representation by OT remains unchanged, i.e., $\hat{H}^s = n_s P^\lambda H_t$

3.3 Cross-Network Node Classification

The fair OT module will generate a transported source node embedding matrix, $\hat{H}^s$. Next $\hat{H}_s$ and the un-transformed target embedding H_t are passed to a fully connected network for node classification. After pre-training, the entire network is trained end-to-end to optimize the following joint objective function, which is a combination of the cross-entropy loss and optimal transport loss.

$$\mathcal{L} = \mathcal{L}_{CE} + \theta \mathcal{L}_{OT} \tag{3}$$

where θ is a hyperparameter that controls the trade-off between the two losses.

Table 1: Breakdown of datasets used in experiments.

dataset	**nodes**	**edges**	$\#(Y = 1)$	$\#(X^{(p)} = 1)$
pokec-n	2933	16821	2145 (73%)	1722 (59%)
pokec-z	3285	22454	2137 (65%)	1844 (56%)
compas-0	2170	137473	1042 (48%)	1690 (78%)
compas-1	1434	148012	674 (47%)	1140 (79%)
Abide-large	804	93886	370 (46%)	685 (85%)
Abide-small	67	587	33 (49%)	42 (63%)
Credit-0	10468	75669	8029 (76%)	5462 (52%)
Credit-1	10169	29227	8384 (82%)	5360 (52%)

3.4 `FastFOCI`

A hindrance to performance in `FOCI` is that the representation learning step must evaluate both the source and target graphs at the same time. The first major improvement implemented in `FastFOCI` is that the representation learning phase will consider the source data only. Additionally, in `FastFOCI`, the source graph will be fed to source GCN and NFT layers, GCN$_s$ and NFT$_s$. In pretraining the output of these layers will be sent to a classifier and source predictions will be used to update the GCN$_s$, NFT$_s$, and classifier. GCN$_s$ and NFT$_s$ will be used to construct a source representation H_S while GCN$_t$ and NFT$_t$ is used to construct a target representation H_T. In this way we reduce the number of edges required by the GCN, which no longer requires full access to both source and target graphs at the same time. Finally, as the classifier does not need all node representations at the same time, we can improve the speed of the OT layer by inserting a sampling step after node representations are acquired and before OT occurs. Specifically, we randomly sample representations from both the source and target embeddings H_S and H_T described above and then pass those samples on to OT. We can then transport the sampled source representations to send on to the classifier.

4 Experimental Evaluation

We consider the following datasets for our experiments (see Table 1 for more details):

- **Pokec**[11] is a popular social media platform in Slovakia. The node feature information is obtained from the user profile data while the link structure represents relationships between users. The classification task is to predict whether a user smokes with the user's sex as protected attribute. We use data from two geographical regions—Pokec-n and Pokec-z—to form the source and target networks.
- **Compas**[1] is a recidivism dataset in which each node corresponds to an incarcerated individual while an edge is formed by connecting individuals who were incarcerated during an overlapping period. The classification task is to predict whether an individual will re-offend again in the future with race as the protected attribute. The source and target networks are created by applying spectral clustering to split the network into 2 subgraphs, denoted as Compas-0 and Compas-1, respectively.

Table 2: Comparison of `FOCI` and baselines ACDNE and ASN in terms of F1-Score (F1) and statically parity (SP). First and second place values are highlighted in gold and silver respectively.

Source	**Target**		ACDNE	ASN	`FOCI`
abide-large	abide-small	F1	0.944 +/- 0.038	0.318 +/- 0.201	0.970 +/- 0.008
		SP	0.110 +/- 0.012	0.008 +/- 0.008	0.102 +/- 0.012
pokec-n	pokec-z	F1	0.836 +/- 0.002	0.748 +/- 0.022	0.790 +/- 0.004
		SP	0.080 +/- 0.015	0.008 +/- 0.006	0.007 +/- 0.006
pokec-z	pokec-n	F1	0.836 +/- 0.003	0.750 +/- 0.021	0.784 +/- 0.008
		SP	0.148 +/- 0.005	0.012 +/- 0.009	0.056 +/- 0.016
compas-0	compas-1	F1	0.634 +/- 0.011	0.028 +/- 0.058	0.560 +/- 0.187
		SP	0.172 +/- 0.007	0.001 +/- 0.002	0.122 +/- 0.041
compas-1	compas-0	F1	0.604 +/- 0.008	0.318 +/- 0.201	0.638 +/- 0.009
		SP	0.170 +/- 0.013	0.008 +/- 0.008	0.163 +/- 0.011

- **ABIDE**[4] is a popular dataset for studying Autism Spectrum Disorder (ASD). Simular to previous work [15], we construct a population graph by using phenotypic subject data as node features and resting-state fMRI similarity to construct the edges. Here the presence of ASD is node label and sex is the protected attribute. The dataset was split into two separate graphs according to their data collection sites. Given the need to have a large enough sample to train GNN models, ABIDE will only be trained with the larger graph as source and the smaller graph as target.
- **Credit** [19] is a financial dataset where the task is to predict whether an individual will default on a loan. The protected attribute is the age of the individual, and edges are formed between individuals with similar spending and payment patterns. Credit-0 was created by selecting the highest degree node and then repeatedly adding all immediate degree neighbors until the graph has over 10,000 nodes. Once the nodes in Credit-0 were removed, the process was repeated to obtain Credit-1.

The source code of `FOCI` can be found at `https://github.com/ajoystephens/foci`. We compared FOCI against the following CNNC baseline methods.

- **ACDNE** [17]: This approach uses an adversarial deep network embedding approach for domain adaptation. It learns a separate embedding from the node features and link structure before sending them to a discriminator.
- **ASN** [20]: This approach uses a series of 2-layer GCNs and GCN variational autoencoders (VAE) to learn the feature embedding. It addresses the domain adaptation issue using an adversarial discriminator.

For every method, we performed 10-fold cross fold validation on the source dataset to select their best hyperparameters. For the baseline methods, the range of their hyperparameter values include those reported in their authors' published papers and source code. We repeated our experiment 10 times with different random seeds and recorded the average F1-score and statistical parity values.

Source	Target	Method	F1-Score	SP	Epoch Time (s)
pokec-n	pokec-z	FOCI	**0.790 +/- 0.004**	**0.007 +/- 0.006**	0.569 +/- 0.067
		FastFOCI	0.773 +/- 0.041	0.013 +/- 0.014	**0.117 +/- 0.042**
pokec-z	pokec-n	FOCI	**0.784 +/- 0.008**	0.056 +/- 0.016	0.627 +/- 0.065
		FastFOCI	0.756 +/- 0.071	**0.047 +/- 0.027**	**0.110 +/- 0.029**
compas-0	compas-1	FOCI	0.560 +/- 0.187	**0.122 +/- 0.041**	0.277 +/- 0.024
		FastFOCI	**0.599 +/- 0.085**	0.158 +/- 0.027	**0.108 +/- 0.022**
compas-1	compas-0	FOCI	**0.638 +/- 0.009**	**0.163 +/- 0.011**	0.177 +/- 0.021
		FastFOCI	0.622 +/- 0.037	0.169 +/- 0.026	**0.084 +/- 0.023**
credit-0	credit-1	FastFOCI	**0.883 +/- 0.001**	**0.014 +/- 0.001**	**0.123 +/- 0.089**
credit-1	credit-0	FastFOCI	**0.868 +/- 0.009**	**0.006 +/- 0.004**	**0.145 +/- 0.195**

Table 3: Comparison of FOCI and FastFOCI and in terms of F1-Score, statically parity (SP), and epoch time in seconds. Best values are **bold**.

4.1 Experimental Results

The results comparing FOCI to other CNNC approaches are shown in Table 2. Here the goal is to achieve high node classification results, as measured by F1-score, while simultaneously improving fairness, as measured by the statistical parity measure:

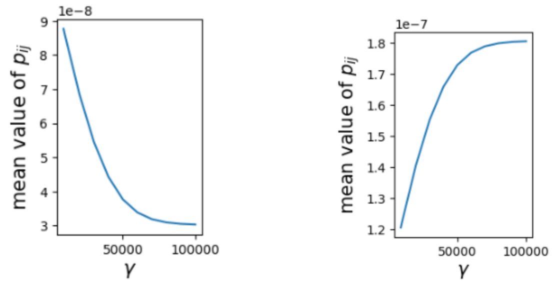
$$SP = |P(\hat{Y} = 1|X^{(p)} = 0) - P(\hat{Y} = 1|X^{(p)} = 1)|$$

The results in Table 2 suggest FOCI is consistently one of the best methods in both F1-Score and statistical parity. ACDNE has the best F1-Score in most cases, but the worst statistical parity. ASN has often improved statistical parity but this coincides with reduced F1-Score. FOCI, however, manages to maintain a high F1-Score while reducing statistical parity, balancing the fairness/utility trade off. Note that the credit dataset is not considered in Table 2 as methods aside from FastFOCI were unable to process the large credit dataset.

Table 3 compares results from FOCI and FastFOCI in terms of F1-score, statistical parity, and run time. FOCI results for the credit datasets are excluded from this table because the FOCI method was unable to process the credit dataset. FOCI's optimal transport layer encountered numerical errors when attempting to transport the entire source and target credit datasets. Observe that FastFOCI provides significant run time improvements over FOCI, but with a slight decline in model utility and fairness. This suggests that FastFOCI may be the best option with larger datasets, but that FOCI may be the better choice if dataset size and runtime are not significant concerns.

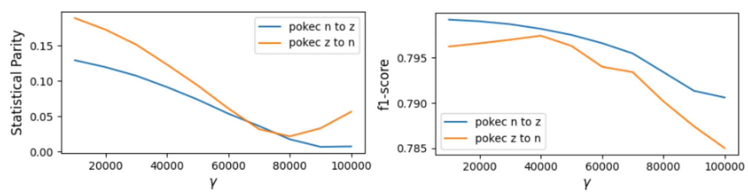
4.2 Ablation Study

We investigate the impact of the hyperparameter γ on the OT transport plan and model outcome. Here, we restrict our discussion to the Pokec datasets and the FOCI approach. First, we vary γ and evaluate the resulting transport plan matrix P^{γ}. In these and all other experiments involving OT we held $\lambda = 0.03$. Fig. 3 shows these results by plotting γ against the mean value of p_{ij} where nodes i and j do or do not share protected groups respectively. Note that mean values for p_{ij} are very small as $P^{\gamma} \in R^{m \times n}$. Next we



(a) same protected groups (b) different protected groups

Fig. 3: Impact of γ on transport plan matrix when mapping from pokec-n to pokec-z. As γ increases transport plan values decrease between nodes which share a protected group and increase for nodes which are in different protected groups.



(a) impact of γ on statistical parity (b) impact of γ on f1-score

Fig. 4: Impact of γ on model outcomes.

examine the impact of γ on statistical parity and F1-score. Here, γ was varied within the range that produces an impact on P^{γ} in Fig. 3, while all other hyperparameters were kept the same. The results shown in Fig 4 suggest that the statistical parity and F1-scores do not vary significantly within the given range.

5 Conclusion

This paper presents a framework called `FOCI` for fair cross-network node classification and a framework called `FastFOCI` which offers improved efficiency. Both frameworks use a novel fair Sinkhorn distance measure to encourage mapping between members of different protected groups in the source and target networks. We have experimentally shown that the proposed fair Sinkhorn distance helps to mitigate unfairness while maintaining high accuracy, comparable to other state-of-the-art CNNC baselines.

Acknowledgment

This material is based upon work supported by NSF under grant #IIS-1939368 and #IIS-2006633. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias. ProPublica (2016)
2. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: Proceedings of the AAAI conference on artificial intelligence. vol. 30 (2016)
3. Courty, N., Flamary, R., Tuia, D.: Domain adaptation with regularized optimal transport. In: Proc of ECML PKDD. pp. 274–289 (2014)
4. Craddock, C., Sikka, S., Cheung, B., Khanuja, R., Ghosh, S.S., Yan, C., Li, Q., Lurie, D., Vogelstein, J., Burns, R., et al.: Towards automated analysis of connectomes: The configurable pipeline for the analysis of connectomes (c-pac). Front Neuroinform **42**, 10–3389 (2013)
5. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Advances in Neural Information Processing Systems. vol. 26 (2013)
6. Dai, E., Wang, S.: Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 680–688 (2021)
7. Dong, Y., Lizardo, O., Chawla, N.V.: Do the young live in a "smaller world" than the old? age-specific degrees of separation in a large-scale mobile communication network. arXiv preprint arXiv:1606.07556 (2016)
8. Kantorovitch, L.: On the translocation of masses. Management Science **5**(1), 1–4 (1958). `https://doi.org/10.1287/mnsc.5.1.1`, `https://doi.org/10.1287/mnsc.5.1.1`
9. Karimi, F., Génois, M., Wagner, C., Singer, P., Strohmaier, M.: Homophily influences ranking of minorities in social networks. Scientific reports **8**(1), 11077 (2018)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
11. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data` (Jun 2014)
12. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. Advances in neural information processing systems **27** (2014)
13. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning (2018)
14. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. Annual review of sociology **27**(1), 415–444 (2001)
15. Parisot, S., Ktena, S.I., Ferrante, E., Lee, M., Moreno, R.G., Glocker, B., Rueckert, D.: Spectral graph convolutions for population-based disease prediction. In: Proc of MICCAI. pp. 177–185 (2017)
16. Rahman, T., Surma, B., Backes, M., Zhang, Y.: Fairwalk: Towards fair graph embedding. In: Proc of IJCAI. pp. 3289–3295 (2019)
17. Shen, X., Dai, Q., Chung, F.l., Lu, W., Choi, K.S.: Adversarial deep network embedding for cross-network node classification. In: Proc of AAAI Conference on Artificial Intelligence. vol. 34, pp. 2991–2999 (2020)
18. Shen, X., Dai, Q., Mao, S., Chung, F.l., Choi, K.S.: Network together: Node classification via cross-network deep network embedding. IEEE Transactions on Neural Networks and Learning Systems **32**(5), 1935–1948 (2020)
19. Yeh, I.C., Lien, C.h.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert systems with applications **36**(2), 2473–2480 (2009)
20. Zhang, X., Du, Y., Xie, R., Wang, C.: Adversarial separation network for cross-network node classification. In: Proc of ACM Int'l Conf on Info & Knowl Mgmt. pp. 2618–2626 (2021)