Cross-Domain Relation Adaptation

Ido Kessler[1] KESSIDO@GMAIL.COM
Omri Lifshitz[1] OMRI.LIFSHTZ@GMAIL.COM
Sagie Benaim[2] SAGIE.BENAIM@MAIL.HUJI.AC.IL
Lior Wolf[1] WOLF@CS.TAU.AC.IL
[1] *School of Computer Science, Tel Aviv University*
[2] *Department of Computer Science, Hebrew University*

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

We consider the challenge of establishing relationships between samples in distinct domains, A and B, using supervised data that captures the intrinsic relationships within each domain. In other words, we present a semi-supervised setting in which there are no labeled mixed-domain pairs of samples. Our method is derived based on a generalization bound and incorporates supervised terms for each domain, a domain confusion term on the learned features, and a consistency term for domain-specific relationships when considering mixed-domain sample pairs. Our findings showcase the efficacy of our approach in two disparate domains: (i) Predicting protein-protein interactions between viruses and hosts by modeling genetic sequences. (ii) Forecasting link connections within citation graphs using graph neural networks.

Keywords: Unsupervised Domain Adaptation, Cross-domain learning, Protein-Protein Interactions, Graph Neural Networks.

1. Introduction

A key element to the success of deep learning methods is the advent of effective transfer learning. While many classical machine learning methods notoriously require that the test and training distributions be as close as possible, the generalization ability of deep neural networks meant that supervised or unsupervised learning in a source domain would greatly benefit downstream learning in a target domain.

Naturally, some of the knowledge gained in the source domain is irrelevant to the target domain and one has to selectively employ such knowledge, e.g., by only using the lower layers, fine-tuning the knowledge to align with the target domain, or tailoring the initial training in the source domain to suit the characteristics of the target domain.

The final category of techniques is known as domain adaptation. They have solid theoretical foundations, with many of the leading methods derived based on generalization bounds that employ terms arising from the triangular inequality: the error of the learned function on the target domain is smaller than the sum of (i) the error of that function on the source domain, and (ii) a term quantifying the maximal discrepancy between the domains for the relevant family of functions.

In conventional learning in the first domain, the primary focus is on minimizes the first term. The second term is minimized by providing the learned function with features that are domain-agnostic,

i.e, features that do not disclose the domain from which the sample originates. Many contemporary methods enforce this constraint through the incorporation of adversarial terms.

We present a new type of domain adaptation problem, where we learn a relation between pairs of samples. We focus on symmetric relations and discrete labels. The pair of samples can be, for example, 'equivalent' or not, 'same' or 'different', and 'connected' or not.

Three sets of training samples are given: (i) a set of labeled pairs in a first domain A (all samples of all pairs are from this domain), (ii) a set of similar pairs in a second domain B, and (iii) a set of unlabeled mixed pairs, where one sample is from each domain. For the third set, we do not need to know which sample in each pair comes from which domain. If the third set cannot be obtained, we can create a virtual set by mixing samples from the first two sets to create mixed-domain pairs.

The goal is to learn a classifier that estimates the relation for mixed pairs of samples, from the distribution from which the third set is obtained.

As an example, consider the problem of protein interaction networks in humans and viruses. Biologists have constructed graphs that represent the interactions between proteins within an organism. These graphs indicate whether two proteins interact or not. Given such connectivity maps for humans and viruses, our objective is to deduce whether a specific human protein can interact with a particular virus protein. This information, which one can validate using several advanced lab methods, is relevant for understanding infection mechanisms.

Other computational problems that share our settings include, for example, knowledge-graphs relationships between different languages, cross-domain recommendations (e.g., books to movies), connecting user data across multiple services, or identifying related content in news-sites' articles and forums.

Our algorithm is based on training two relation-classifiers, one in each domain, and applying both to the mixed samples, considering their averaged prediction. In this method, domains A and B are treated in a symmetrical manner.

The method proposed is based on a new generalization bound that we derive. The error of the combined classifier on mixed samples is bounded by terms that evaluate the performance of each classifier on its domain, a term that assesses the consistency between the two learned classifiers in the mixed domain, and additional terms that ensure that the encoding of pairs of samples is agnostic to their domain: A, B, or mixed.

The terms of the generalization bound are readily converted to loss terms, which are then applied to a compound neural network. This network includes a feature extraction network, which processes pairs of samples as input, two classifiers that are applied on top of the resulting representations, and a domain-confusion network that is trained through adversarial techniques.

We demonstrate the effectiveness of our method in two markedly diverse problem domains: the virus-host problem mentioned above, and graph link prediction, where we separate the vertices based on category and predict inter-group links after training on labeled intra-group links. In each case, our proposed method is superior to the alternatives by a sizable margin.

2. Related Work

2.1. Unsupervised Domain Adaptation

In unsupervised domain adaptation, the algorithm is tasked with learning a labeling function over a target domain based on two inputs: (i) a dataset of labeled samples from a source domain and (ii) a dataset of unlabeled samples from a target domain. Earlier attempts suggested various statistical

methods for selecting features that are predictive and meaningful in both domains (Blitzer et al., 2006; Chelba and Acero, 2006; Ando and Zhang, 2005). Theoretical analysis of domain adaptation generalization bounds by (Ben-David et al., 2006; Mansour et al., 2009; Ben-David et al., 2010) suggested minimizing the discrepancy between the distribution of the features on both domains.

Following the rise of neural networks and their success in various predictive tasks, approaches trying to reduce the discrepancy between domains using a neural network and based on back-propagation optimization have arisen, e.g., (Long et al., 2015, 2017) have suggested minimizing the discrepancy itself using an efficient statistical estimation.

Following the Generative Adversarial Network (GAN) framework (Goodfellow et al., 2014), (Ganin et al., 2016; Tzeng et al., 2017) minimize the discrepancy using adversarial learning. These methods employ a domain classifier that discriminates between the encoded features extracted from the two domains, and a feature extraction mechanism that tries to fool this classifier while maximizing the classification score on the source domain. Adversarial methods have led to a remarkable increase in performance over the previous methods.

2.2. Protein-Protein Interactions (PPIs)

Proteins are sequences of amino-acids, which are used for various functions inside organisms. Many proteins interact with one another. It was estimated by (Stumpf et al., 2008) that there are $650,000$ PPIs in the human body alone, which is still relatively sparse when considering that there are approximately $400,000$ proteins in the human body. Aside from intra-species PPI, many viruses and bacteria have proteins that interact with their host's proteins. The understanding of those cross-domain interaction networks sheds light on infection mechanisms and on host cell disruption and can promote drug design efforts.

Viewing proteins as sequences of amino-acids, one may use various tools developed for 1D sequence analysis in Natural Language Processing (NLP) to study proteins, including CNNs, LSTMs, and multi-layer perceptrons. Following the emerging dominance of pretrained transformers in NLP (Vaswani et al., 2017), (Rao et al., 2019; Rives et al., 2019; Elnaggar et al., 2020) pretrained transformers for protein amino-acid sequences using masked language modeling and showed their efficiency for various protein prediction tasks.

Earlier attempts to employ machine learning for virus-host PPIs prediction relied on classic machine learning approaches, looking only at small interaction networks of humans with specific families of viruses. More recently, (Ahmed et al., 2018) used a multi-layered neural network to predict interactions between humans and Bacillus Anthracis, and (Yang et al., 2020) employed a doc2vec embedding along with a random forest classifier to infer the interaction network of virus-humans, using the amino-acid sequences as doc2vec inputs. All of these previous attempts relied on supervised learning of the virus-host relations, while our work performs this task without inter-species labels.

2.3. Graph link prediction

Graph link prediction is the task of predicting whether two vertices in a graph are connected via an edge, using previously known existing edges. Given the generality of the problem, many tasks can be described using those specifications, e.g. the relationship between entities in knowledge graphs, user preferences in various recommendation systems, and friend and family suggestions and identifications among users of social networks.

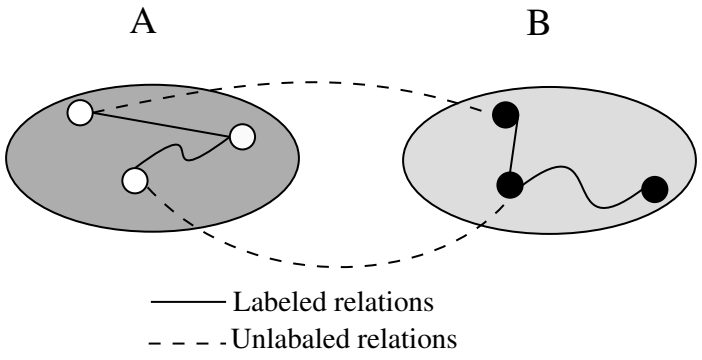


Figure 1: The cross-domain adaptation setting.

With the recent advances in neural network methods, various algorithms have been used to harness their effectiveness for graph-related problems. (Kipf and Welling, 2016) presented a way to use Convolutional Neural Network(CNN) and Variational-Autoencoder(VAE) on graphs using the symmetric normalized graph's Laplacian. (Hettige et al., 2020) presented an efficient graph embedding method that scales to large graphs and preserves both local and global features.

In our work, we focus on the problem of inferring inter-group links given intra-group links, which was not studied in the literature as far as we can ascertain.

3. Problem Setting

We consider a problem with two domains A and B and a set of labels $\mathcal{Y}$ used to categorize the relations between unordered pairs of samples. During training, the pairs are homogeneous, i.e., we have training samples from $\mathcal{A} \times \mathcal{A} \times \mathcal{Y}$ and $\mathcal{B} \times \mathcal{B} \times \mathcal{Y}$, but no samples from $\mathcal{A} \times \mathcal{B} \times \mathcal{Y}$, see Fig. 1 for visualization. Our goal is to learn a function $y_T : (\mathcal{A} \times \mathcal{B}) \to \mathcal{Y}$. We assume that this function is obtained by limiting a more general function $y : (\mathcal{A} \cup \mathcal{B})^2 \to \mathcal{Y}$ to the mixed domain. We also assume that the pairs are unordered $y(u, v) = y(v, u)$, regardless of the domain of samples u and v.

The cross-domain adaptation problem is formulated as a tuple, $(\mathcal{H}_1, \mathcal{H}_2, \mathcal{D}_{\mathcal{A}^2}, \mathcal{D}_{\mathcal{B}^2}, \mathcal{D}_\times, y, \ell)$:

- $\mathcal{H}_1 = \{f : (\mathcal{A} \cup \mathcal{B})^2 \to \mathcal{F}\}$, a set of feature maps.
- $\mathcal{H}_2 = \{g : \mathcal{F} \to [0, 1]^{|\mathcal{Y}|}\}$, a set of classifiers, returning a vector of pseudoprobabilities over the labels.
- $\mathcal{D}_{\mathcal{A}^2}, \mathcal{D}_{\mathcal{B}^2}$ distributions over the source domains $\mathcal{A}^2, \mathcal{B}^2$.
- $\mathcal{D}_\times$, a distribution over the target domain $\mathcal{A} \times \mathcal{B}$.
- $y : (\mathcal{A} \cup \mathcal{B})^2 \to \mathcal{Y}$, the labeling function over pairs mentioned above.
- $\ell : [0, 1]^{|\mathcal{Y}|} \times [0, 1]^{|\mathcal{Y}|} \to \mathcal{R}^+$, a loss function. The loss is also defined over the domain $[0, 1]^{|\mathcal{Y}|} \times \mathcal{Y}$, in which case we consider the one-hot vector of length $|\mathcal{Y}|$ for the label $t \in \mathcal{Y}$ as the second argument of ℓ.

The hypothesis class in this setting is $\mathcal{H} = \{g \circ f | f \in \mathcal{H}_1, g \in \mathcal{H}_2\}$. f is used to extracts features from a pair of samples, regardless of their domain, and g classifies the obtained features into a distribution over the labels.

The fitting of some hypothesis h on the target distribution $\mathcal{D}_\times$ is measured by the *target generalization risk*:

$$\mathrm{R}_{\mathcal{D}_\times}[h, y] = \mathbb{E}_{x \sim \mathcal{D}_\times}\left[\ell\left(h(x), y(x)\right)\right] . \tag{1}$$

In other words, the goal of the learning algorithm is to pick $argmin_{h \in \mathcal{H}}(\mathrm{R}_{\mathcal{D}_\times}[h, y])$. Since the target labels are unknown at the learning phase, we wish to bound $\mathrm{R}_{\mathcal{D}_\times}[h, y]$ with terms that can be optimized during training.

Inspired by the generalization risk bound introduced by (Mansour et al., 2009) for the problem of unsupervised domain adaptation, we devise a theorem for the setting of cross-domain adaptation. The theorem relies on the notion of the *discrepancy* between two distributions $\mathcal{D}_1, \mathcal{D}_2$ over a hypothesis class C, which is defined as:

$$disc_C[D_1, D_2] = \sup_{c_1, c_2 \in C} |\mathrm{R}_{\mathcal{D}_1}[c_1, c_2] - \mathrm{R}_{\mathcal{D}_2}[c_1, c_2]| \tag{2}$$

The theorem also considers three "ideal" classifiers:

$$h^*_{\mathcal{A}^2} = \underset{h \in \mathcal{H}}{argmin}(\mathrm{R}_{\mathcal{D}_{\mathcal{A}^2}}[h, y]) \tag{3}$$

$$h^*_{\mathcal{B}^2} = \underset{h \in \mathcal{H}}{argmin}(\mathrm{R}_{\mathcal{D}_{\mathcal{B}^2}}[h, y]) \tag{4}$$

$$h^*_\times = \underset{h \in \mathcal{H}}{argmin}(\mathrm{R}_{\mathcal{D}_\times}[h, y]) . \tag{5}$$

Theorem 1. *Assume that the loss function ℓ is symmetric, convex, and obeys the triangle inequality. Then, for any two hypotheses $h_1 = g_1 \circ f \in \mathcal{H}$, $h_2 = g_2 \circ f \in \mathcal{H}$, and their ensemble $\frac{h_1+h_2}{2}(x) = \frac{h_1(x)+h_2(x)}{2}$, the following holds:*

$$\begin{aligned}\mathrm{R}_{\mathcal{D}_\times}\left[\frac{h_1+h_2}{2}, y\right] \leq\ & \mathrm{R}_{\mathcal{D}_{\mathcal{A}^2}}\left[h_1, h^*_{\mathcal{A}^2}\right] + \\ & \mathrm{R}_{\mathcal{D}_{\mathcal{B}^2}}\left[h_2, h^*_{\mathcal{B}^2}\right] + \mathrm{R}_{\mathcal{D}_\times}[h_1, h_2] + \underset{\mathcal{H}_2 \circ f}{disc}\left[\mathcal{D}_{\mathcal{A}^2}, \mathcal{D}_\times\right] \\ & + \underset{\mathcal{H}_2 \circ f}{disc}\left[\mathcal{D}_{\mathcal{B}^2}, \mathcal{D}_\times\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{A}^2}, h^*_\times\right] \\ & + \mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{B}^2}, h^*_\times\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h^*_\times, y\right]\end{aligned} \tag{6}$$

Proof.

$$\mathrm{R}_{\mathcal{D}_\times}\left[\frac{h_1+h_2}{2}, y\right] \leq \tag{7}$$

$$\text{(triangle)} \leq \mathrm{R}_{\mathcal{D}_\times}\left[\frac{h_1+h_2}{2}, \frac{h^*_{\mathcal{A}^2}+h^*_{\mathcal{B}^2}}{2}\right] \tag{8}$$

$$+ \mathrm{R}_{\mathcal{D}_\times}\left[\frac{h^*_{\mathcal{A}^2}+h^*_{\mathcal{B}^2}}{2}, h^*_\times\right] \tag{9}$$

$$
\begin{aligned}
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h^*_\times, y\right] && (10)\\
(\text{convexity}) \leq &\mathrm{R}_{\mathcal{D}_\times}\left[h_1, h^*_{\mathcal{A}^2}\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h_1, h^*_{\mathcal{B}^2}\right] && (11)\\
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h_2, h^*_{\mathcal{A}^2}\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h_2, h^*_{\mathcal{B}^2}\right] && (12)\\
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{A}^2}, h^*_\times\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{B}^2}, h^*_\times\right] && (13)\\
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h^*_\times, y\right] && (14)\\
(\text{triangle}) \leq &\mathrm{R}_{\mathcal{D}_\times}\left[h_1, h^*_{\mathcal{A}^2}\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h_1, h_2\right] && (15)\\
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h_2, h^*_{\mathcal{B}^2}\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h_2, h_1\right] && (16)\\
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h_1, h^*_{\mathcal{A}^2}\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h_2, h^*_{\mathcal{B}^2}\right] && (17)\\
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{A}^2}, h^*_\times\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{B}^2}, h^*_\times\right] && (18)\\
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h^*_\times, y\right] && (19)\\
(\text{definitions}) \leq &\mathrm{R}_{\mathcal{D}_{\mathcal{A}^2}}\left[h_1, h^*_{\mathcal{A}^2}\right] + \mathrm{R}_{\mathcal{D}_{\mathcal{B}^2}}\left[h_2, h^*_{\mathcal{B}^2}\right] && (20)\\
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h_1, h_2\right] + \underset{\mathcal{H}_2 \circ f}{disc}\left[\mathcal{D}_{\mathcal{A}^2}, \mathcal{D}_\times\right] && (21)\\
&+ \underset{\mathcal{H}_2 \circ f}{disc}\left[\mathcal{D}_{\mathcal{B}^2}, \mathcal{D}_\times\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{A}^2}, h^*_\times\right] && (22)\\
&+ \mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{B}^2}, h^*_\times\right] + \mathrm{R}_{\mathcal{D}_\times}\left[h^*_\times, y\right] && (23)
\end{aligned}
$$

□

Most of the terms in the RHS of Eq. 6 can be estimated and are converted to loss terms in our method.

- $\mathrm{R}_{\mathcal{D}_{\mathcal{A}^2}}\left[h_1, h^*_{\mathcal{A}^2}\right]$ is the error between h_1 and the best hypothesis in H with respect to $\mathcal{D}_{\mathcal{A}^2}$. Since the classification error includes both this term and the modeling error of class $\mathcal{H}$ but the latter is fixed, this is estimated in domain adaptation methods using the training error in domain $\mathcal{A}$.
- $\mathrm{R}_{\mathcal{D}_{\mathcal{B}^2}}\left[h_2, h^*_{\mathcal{B}^2}\right]$ is the error between h_2 and the best hypothesis in H with respect to $\mathcal{D}_{\mathcal{B}^2}$. It is estimated by the training error in domain $\mathcal{B}$.
- $\mathrm{R}_{\mathcal{D}_\times}\left[h_1, h_2\right]$ is a consistency term between h_1 to h_2.
- $disc\left[\mathcal{D}_{\mathcal{A}^2}, \mathcal{D}_\times\right] + disc\left[\mathcal{D}_{\mathcal{B}^2}, \mathcal{D}_\times\right]$, measures the discrepancy between $\mathcal{D}_{\mathcal{A}^2}, \mathcal{D}_{\mathcal{B}^2}$, and $\mathcal{D}_\times$ on the hypothesis space $\mathcal{H}_2 \circ f$. This term can be estimated by employing a domain confusion term.

The term $\mathrm{R}_{\mathcal{D}_\times}\left[h^*_\times, y\right]$ denotes the expressiveness of the hypothesis class in terms of the target labeling. It is a fixed value, given a learning problem and the hypothesis class.

Finally, the terms $\mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{A}^2}, h^*_\times\right]$ and $\mathrm{R}_{\mathcal{D}_\times}\left[h^*_{\mathcal{B}^2}, h^*_\times\right]$ are the differences between the ideal classifiers in the source domains and the ideal one in the target domain, as measured on the target domain. This cannot be estimated. However, similar to the unsupervised domain literature (Mansour et al., 2009), which presents domains of the form $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$, where $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$ are the source and target domain distributions respectively, these terms are assumed to be small.

4. Method

Thm. 1 shows that we can bound the target generalization risk of two-classifier ensembles, by considering the source generalization risk, the discrepancy between the domains, the consistency between the two classifiers on the target domain, and the difference between the labeling functions in the different domains. We, therefore, seek to minimize the four optimizable components of the bound: (i) the risk of h_1 on $\mathcal{A}^2$, denoted by $\mathrm{R}_{\mathcal{D}_{\mathcal{A}^2}}\left[h_1, h^*_{\mathcal{A}^2}\right]$, (ii) the risk of h_2 on $\mathcal{B}^2$, denoted by $\mathrm{R}_{\mathcal{D}_{\mathcal{B}^2}}\left[h_2, h^*_{\mathcal{B}^2}\right]$, (iii) the consistency of h_1 and h_2 on $\mathcal{D}_\times$, denoted by $\mathrm{R}_{\mathcal{D}_\times}\left[h_1, h_2\right]$, and (iv) the discrepancy of $\mathcal{D}_{\mathcal{A}^2}$, $\mathcal{D}_{\mathcal{B}^2}$ with $\mathcal{D}_\times$ with regard to the hypothesis space $\mathcal{H}_2 \circ f$, $disc\left[\mathcal{D}_{\mathcal{A}^2}, \mathcal{D}_\times\right] + disc\left[\mathcal{D}_{\mathcal{B}^2}, \mathcal{D}_\times\right]$.

Naturally, we do not have access to the underlying distributions $\mathcal{D}_{\mathcal{A}^2}, \mathcal{D}_{\mathcal{B}^2}$ and $\mathcal{D}_\times$, only to the training samples. The training set $\mathcal{S}_\mathcal{A}$ ($\mathcal{S}_\mathcal{B}$) denotes the training samples from domain $\mathcal{A}$ ($\mathcal{B}$), each sample being a tuple in $\mathcal{A} \times \mathcal{A} \times \mathcal{Y}$ ($\mathcal{B} \times \mathcal{B} \times \mathcal{Y}$). In addition, we employ an unlabeled mixed training set $\mathcal{S}_\times$, specifically designed to depict the cross-domain distribution. This set comprises pairs of unlabelled samples, each pair containing one sample from each domain ($\mathcal{A} \times \mathcal{B}$).

The method learns three main networks: the classifiers g_1, g_2, and a common feature extraction backbone f. In addition, a single discriminator d is learned. Similar to Thm. 1, we denote $h_1 = g_1 \circ f \in \mathcal{H}$, $h_2 = g_2 \circ f \in \mathcal{H}$.

The terms $\mathrm{R}_{\mathcal{D}_{\mathcal{A}^2}}\left[h_1, h^*_{\mathcal{A}^2}\right]$, $\mathrm{R}_{\mathcal{D}_{\mathcal{B}^2}}\left[h_2, h^*_{\mathcal{B}^2}\right]$ are optimized by the following two loss terms, respectively:

$$\mathcal{L}_\mathcal{A} = \sum_{(a_1,a_2,y)\in\mathcal{S}_\mathcal{A}} \ell(g_1(f(a_1,a_2)), y) \tag{24}$$

$$\mathcal{L}_\mathcal{B} = \sum_{(b_1,b_2,y)\in\mathcal{S}_\mathcal{B}} \ell(g_2(f(b_1,b_2)), y)\,, \tag{25}$$

where for the loss ℓ, cross-entropy is used.

Denote by ℓ^{sym} the symmetric cross-entropy loss $\ell^{sym}[a,b] = \frac{\ell[a,b]+\ell[b,a]}{2}$. The consistency term $\mathrm{R}_{\mathcal{D}_\times}\left[h_1, h_2\right]$ can be readily estimated, by employing the set $\mathcal{S}_\times$ as follows:

$$\mathcal{L}_\times = \sum_{(a,b)\in\mathcal{S}_\times} \ell^{sym}(g_1(f(a,b)), g_2(f(a,b)))\,, \tag{26}$$

For the discrepancies terms, $disc\left[\mathcal{D}_{\mathcal{A}^2}, \mathcal{D}_\times\right]$, $disc\left[\mathcal{D}_{\mathcal{B}^2}, \mathcal{D}_\times\right]$, we follow (Ganin et al., 2016) and optimize a domain confusion term that includes the discriminator d. Similar to (Taigman et al., 2017), we employ a three-way discriminator trained to distinguish between $f \circ \mathcal{D}_{\mathcal{A}^2}$, $f \circ \mathcal{D}_{\mathcal{B}^2}$ and $f \circ \mathcal{D}_\times$ (the composition of a function with a distribution denotes the distribution obtained by applying the function to the samples of the distribution).

The discriminator d, is implemented as a neural network with the same architecture as the classifiers g_1, g_2. The loss term used for the optimization is then:

$$\mathcal{L}_{disc} = \sum_{(a_1,a_2)\in\mathcal{S}_\mathcal{A}} \left[\ell\left(d(f(a_1,a_2)), 0\right)\right] + \sum_{(b_1,b_2)\in\mathcal{S}_\mathcal{B}} \left[\ell\left(d(f(b_1,b_2)), 1\right)\right] + \sum_{(a,b)\in\mathcal{S}_\times} \left[\ell\left(d(f(a,b)), 2\right)\right] \tag{27}$$

```
VSYASSSRLL   SNKNQFKSFL   RTIPNDEHQA   TAMADIIEYF
RWNWVGTIAA   DDDYGRPGIE   KFREEAEERD   ICIDFSELIS
QYSDEEEIQH   VVEVIQNSTA   KVIVVFSSGP   DLEPLIKEIV
RRNITGKIWL   ASEAWASSSL   IAMPQYFHVV   GGTIGFALKA
GQIPGFREFL   KKVHPRKSVH   NGFAKEFWEE   TFNCHLQEGA
KGPLPVDTFL   RGHEESGDRF   SNSSTAFRPL   CTGDENISSV
ETPYIDYTHL   RISYNVYLAV   YSIAHALQDI   YTCLPGRGLF
TNGSCADIKK   VEAWQ
```

Figure 2: The amino-acid sequence of human's calcium-sensing receptor.

Table 1: Average prediction accuracy (%) on a balanced positive/negative virus-host PPI dataset extracted from viruses.STRING after reducing near-duplication, over all backbone models.

Method	TAPE	ProtBert	ProtAlbert	ProtXLNet
Unsupervised	63.5	66.9	67.6	66.0
Host-host model	63.2	65.6	66.1	67.9
Virus-Virus model	63.4	67.3	66.4	68.6
Conventional DA	64.7	73.6	**69.3**	71.1
Our Full Method	**69.9**	**80.7**	69.1	**78.6**

The discriminator d minimizes $\mathcal{L}_{disc}$, while the backbone feature extractor f maximizes it in an adversarial manner. Following (Ganin et al., 2016) this is obtained by employing a gradient reversal layer. Note that the feature extractor can be either trained from scratch or one can finetune a pretrained backbone that is learned, for example, using self-supervised training.

The final training objective of $f \in \mathcal{H}_1$, $g_1, g_2 \in \mathcal{H}_2$ is

$$\mathcal{L} = \mathcal{L}_{\mathcal{A}} + \mathcal{L}_{\mathcal{B}} + \lambda_1 \mathcal{L}_{\times} - \lambda_2 \mathcal{L}_{disc}\,, \tag{28}$$

for some weight parameters λ_1 and λ_2.

5. Experiments

We evaluate our method on datasets from two different domains: proteins described as a text sequence of amino-acids, and citation networks represented as attributes graphs. In each of these domains, we demonstrate the effectiveness of our algorithm, using alternatives state of the art domain-specific learning techniques. In other words, we apply our method as a template to which we incorporate state of the art methods, demonstrating its real-world applicability.

We also experiment using various settings of the two weight parameters λ_1, λ_2 (Fig.3), showing the low sensitivity of our method to the different hyper-parameters.

All the experiments were performed on an Intel® Core™ i7-5960X processor and four NVIDIA TITAN X GPUs.

5.1. Protein-Protein Interactions

We employ the viruses.STRING dataset (Cook et al., 2018), containing 9 million proteins described by their amino-acid sequence (see Fig. 2), and 652 million interactions. The proteins come from

Table 2: An ablation study showing average prediction accuracy (%) on viruses.STRING using TAPE as the backbone.

Method	Accuracy
Full method	**69.9**
Method without consistency ($\lambda_1 = 0$)	66.8
Method without DA ($\lambda_2 = 0$)	66.0
Method with backbone frozen	65.0
Method with no pretraining	65.4

239 viruses and 319 hosts. The domains $\mathcal{A}$ and $\mathcal{B}$ are then, respectively, host proteins and virus proteins.

The dataset viruses.STRING provide a score for each interaction, depicting its confidence when integrating knowledge from both experiments and the relevant literature. Following (Szklarczyk et al., 2015), we filter interactions by their score at a threshold of 500, in order to consider the interactions which are more likely to be valid. To refrain from training on redundant data, as many proteins have several near-duplicates that evolved from close ancestors, we filter the training set such that no two proteins have sequence-identity larger than 25%(Brenner et al., 1998). For negative samples, we randomly select two proteins from the whole dataset and assert there is no interaction between them. For all datasets we use a balanced positive/negative ratio. For proteins longer than 500 amino-acids, as they might overrun the GPU memory, we randomly select a substring of length 500.

Table 3: AUC of heterogeneous link prediction using GLACE. Reported are mean AUC averaged over 16 runs. Bold fonts are used to denote the best results for the unsupervised methods, whereas an underline emphasizes the best supervised method.

Method	Cora	Cora-ML	DBLP	CiteSeer	PubMed
Supervised (Single Head) (our re-run)	.956	<u>.939</u>	<u>.908</u>	.898	<u>.924</u>
Supervised (Head Per Group)	<u>.963</u>	.938	.903	<u>.924</u>	.918
Unsupervised (Single Head)	.661	.654	.780	.620	.861
Unsupervised (Head Per Group)	.682	.646	.764	.633	.861
Conventional domain adaptation	.696	.787	.721	.623	**.910**
Our full method	**.752**	**.825**	**.812**	**.715**	.851
Our method without consistency	.750	.808	.761	.572	.837
Our method without DA	.685	.809	.810	.713	.858

The representation we employ for the amino-acid sequences is based on a pretrained backbone network f. We use four different transformer models: (i) TAPE, which is a pretrained BERT (Devlin et al., 2019) model, published by (Rao et al., 2019), as well as three models published by (Elnaggar et al., 2020) (ii) ProtBert (Devlin et al., 2019) (iii) ProtAlbert (Lan et al., 2019), and (iv)

ProtXLNet (Yang et al., 2019). We follow the common practice for classification of pairs of strings using transformer-like models (Devlin et al., 2019; Yang et al., 2019), and provide a concatenated version of both proteins, relative positional encodings, and type encodings, and use the final layer representation of the first 'start' token as the embedding.

We report results for multiple methods. (i) The unsupervised method, using a pretrained backbone f and a classification head on the labeled pairs in $\mathcal{S}_\mathcal{A}$ and $\mathcal{S}_\mathcal{B}$, (ii) A method in which we employ the host-host PPI prediction model h_1 as is, trained using $\mathcal{S}_\mathcal{A}$, (iii) reporting similarly for the virus-virus PPI model h_2, trained using $\mathcal{S}_\mathcal{B}$, (iv) the application of conventional domain adaptation, in which we add a discriminator to the unsupervised method, and train the backbone using both the supervised loss $\mathcal{L}_\mathcal{A}$, $\mathcal{L}_\mathcal{B}$, and adversarially using the domain confusion loss $\mathcal{L}_{disc}$ using a single head, similarly to (Ganin et al., 2016), (v) our full model.

Tab. 1 depicts the results of all these methods under different backbones. Evidently, under almost all backbones our method improves the baseline results by a sizable margin. Using stronger unsupervised representations (ProtBert and ProtXLNet), our method achieves $\sim$80% classification rate.

We also run an ablation study, using TAPE as the backbone, where we explore the following variations of our method: (i) our method with the consistency term removed ($\lambda_1 = 0$), (ii) our method with the domain confusion term removed ($\lambda_2 = 0$) (iii) our method with backbone frozen, i.e., initializing f by TAPE's weights and not updating it (iv) our method where we train the backbone f from scratch, using the same architecture as TAPE.

Tab. 2 presents the results of the ablation study. It is evident that both the consistency loss and the domain adaptation terms contribute to the success of the method. It can be observed that the pretraining of the backbone has a significant impact on the accuracy achieved by the model and so does finetuning. Without finetuning, the representation produced by the mask language modeling alone is not sufficient, and results in a sub-optimal solution.

5.2. Graph Link Prediction

We test our method for link prediction in citations graphs using several well-known datasets: Cora, CiteSeer, DBLP, and PubMed. In these datasets, each node represents a publication associated with a group identifier denoting the publication's research topic, and each edge represents a citation.

Each publication in the citation graphs also contains attributes regarding the content of the publication. We follow the common prepossessing procedure (Kipf and Welling, 2016; Hettige et al., 2020) and use the TF-IDF of a publication's abstract as its attribute vector.

In this cross-domain adaptation task, unlike the PPI domain which only had two groups (hosts and viruses), each of the citation graphs contain multiple group identifiers per graph, each representing a different research topic. We look at each group of nodes as a different domain, where the homogeneous edges between nodes of the same group are labeled (intra-group), and the heterogeneous edges between nodes of different groups (inter-groups) are unlabeled and are only used for unsupervised learning.

To verify the generality of our algorithm, we experiment with two different approaches to link inference. In each case, we adapt our method to the exact framework of the approach. The first approach uses GLACE (Hettige et al., 2020), a graph embedding neural networks, which uses Gaussians as its embeddings space. The second approach uses Graph Convolutional Networks (GCN)

(Kipf and Welling, 2016) to embed nodes using both their feature vector and their neighborhoods, using the symmetric normalized graph's Laplacian.

In this setting, as there are multiple groups for each of the graphs, we generalize our method to a many-domains setting: (i) for the domain confusion term, we train the intermediate representations of each method to be indistinguishable between the different groups. This is done using a discriminator that is trained in an adversarial manner to classify each representation to the group it originated from. (ii) for the classification heads, we implement a classification head per group, where each group classification head is trained in a supervised manner using homogeneous edges that connects nodes of this group. (iii) To implement the consistency term, we simply use the squared difference between predictions of different heads on heterogeneous edges.

5.2.1. GLACE

GLACE encodes each attributes vector using a neural network into an intermediate representation, which we treat as the backbone f, and then embeds it into a Gaussian distribution with a diagonal covariance matrix using another linear layer, which we treat as the classifier g. We denote the output Gaussian GLACE outputs for node i as z_i.

The algorithm in GLACE encourage nodes that are connected by an edge to be close in the representation space while distancing nodes that are not connected. Let E be the set of (labeled) edges of the graph G, and $D(z_i, z_j)$ be a dissimilarity measure between the Gaussians. The following objective is used in GLACE:

$$\mathcal{L}_1 = \sum_{(i,j)\in E} \log \sigma(-D(z_i, z_j)) + \sum_{(i,j)\in E_N} \log \sigma(D(z_i, z_j)) \,, \tag{29}$$

where E_N denotes the non-existing edges set of graph G.

As the graphs are sparse, the number of negative edges is extremely large. To solve this, we instead sample a subset of the negative edges for each batch. We use the exact same random sampling used in GLACE, where the negative edges are sampled proportionally to its nodes degrees.

In line with GLACE, we use the symmetric KL divergence to measure the dissimilarity between the embeddings, as they represent Gaussians distributions. Denote $D_{KL}(z_i, z_j)$ as the Kullback–Leibler divergence of the two distribution z_i, z_j, then:

$$D(z_i, z_j) = D_{KL}(z_i, z_j) + D_{KL}(z_j, z_i) \tag{30}$$

To apply our method, we employ a different g for each group. We define, for each group k, the functions g_k, which result in a Gaussian embedding z_i^k for each node i.

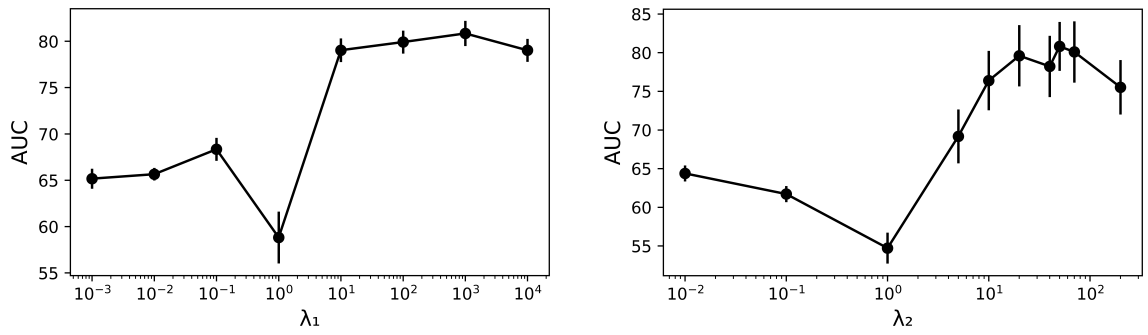
Denote i, j as any two nodes, and k_i and k_j as the groups that nodes i and j belong to respectively. We define our dissimilarity prediction, by generalizing GLACE as follows:

$$D(i, j) = \frac{D(z_i^{k_i}, z_j^{k_i}) + D(z_i^{k_j}, z_j^{k_j})}{2} \tag{31}$$

To optimize the domain confusion terms from the theorem, we use domain adaptation training. A discriminator d with similar architecture to g, classifies each node to its respected group, while the backbone f is trained in an adversarial way using gradient reversal to confuse d. To use a similar loss structure to the classification loss that is used to train g, the discriminator d is tasked to embed the nodes' representation after the backbone in a way where the Gaussian embeddings of intra-group

Table 4: AUC of link prediction using GCN.

Method	Cora-ML	DBLP	CitSeer	PubMed
Supervised (Single Head)	.873	<u>.958</u>	<u>.947</u>	<u>.955</u>
Supervised (Head Per Group)	<u>.907</u>	.937	.880	.922
Unsupervised (Single Head)	.697	.838	.725	.850
Unsupervised (Head Per Group)	.739	.888	.783	.902
Conventional domain adaptation	.769	.882	.911	.912
Our full method	**.803**	**.890**	**.942**	.908
Our method without consistency	.780	.870	.827	**.919**
Our method without DA	.705	.882	.820	.850



(a) (b)

Figure 3: Hyper parameter sensitivity analysis over the Cora_ML graph. (a) varying (a) λ_1 or (b) λ_2.

pairs are close, and the Gaussian embeddings of inter-group pairs are further apart. Denote z_i^d as the Gaussian embedding produced by the discriminator d on node i, then the loss is defined as:

$$\mathcal{L}_d = \sum_{k\in K}\sum_{i,j\in V_k} \log\sigma(-d(z_i^d, z_j^d)) + \sum_{k\in K}\sum_{i\in V_k, j\in V\setminus V_k} \log\sigma(d(z_i^d, z_j^d))\,, \tag{32}$$

where K is the set of groups in the graph and V_k are the vertices of group $k \in K$.

To fit our method's consistency term (Eq. 26) to GLACE, we use the squared error measure between the predictions of the different classifiers on heterogeneous edges:

$$\mathcal{L}_c = \sum_{(i,j)\in V_{k_1}\times V_{k_2}, k_1\neq k_2} (\log\sigma(d(z_i^{k_1}, z_j^{k_1})) - \log\sigma(d(z_i^{k_2}, z_j^{k_2})))^2 \tag{33}$$

Link prediction experiments with GLACE The results of the link prediction experiments with GLACE are reported in Tab. 3. We can see that in almost all the datasets our method outperforms the unsupervised baseline and the conventional domain adaptation method by a sizable margin.

Removing either the consistency or the domain adaptation term from our method hurt results in almost all benchmarks. Removing the consistency term hinders the performance in all benchmarks by a significant margin, except for on the full Cora benchmark, where the difference is smaller.

The sensitivity of our method to its hyper-parameters is explored in Fig. 3, where we vary either the consistency coefficient λ_1, or the domain confusion coefficient λ_2. These experiments are performed on the Cora-ML benchmark. As can be seen, performance is stable for a wide range of $\lambda_1 \geq 10$. For λ_2 performance is stable over a smaller region, with the best results in the range of $10 < \lambda_2 < 100$.

5.2.2. GRAPH CONVOLUTIONAL NETWORKS

For the second experiment, we used a GCN (Kipf and Welling, 2016) to model our network. Denote $G = (V, E)$ as the graph the convolution layer works on, $X \in \mathbb{R}^{|V| \times F_0}$ as representation vectors of some size F_0 for all nodes in V, A as the adjacency matrix of the graph G, $\hat{A} = I + A$ as the adjacency matrix with self-loops added, and $D_{ii} = \sum_{j=1}^{|V|} \hat{A}_{ij}$ as the diagonal matrix of the degree of each node. Then a graph convolution layers, parameterized by some Θ^l, can be described using the following notion:

$$X' = GCN(X; \Theta^l) = D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}} X \Theta^l \,, \tag{34}$$

where $\Theta^l \in \mathbb{R}^{F_0 \times F_1}$ and $X' \in \mathbb{R}^{|V| \times F_1}$ where F_1 is the dimension of the representations of the nodes after the layer.

Denoted by K the number of categories in the graph. Our method employs $2 + K$ GCN layers. One represents f, the backbone, which reduces the TD-IDF representation to an intermediate representation. The other K are g_k, the embedding layers of each category $k = 1..K$, which transform the intermediate representation into a final representation. The last one is the discriminator d which is represented by a similar GCN layer.

The adjacency matrix used in the GCN layers only contains labeled homogeneous (intra-group) edges, which reduces each group in the graph into a separate component.

For the classification loss, we employ a standard reconstruction loss, using the inner product between the embeddings to denote the logits of an edge between two nodes.

The domain discriminator d is implemented using a GCN layer similar to the classifiers, and outputs, for each node, a distribution over the K categories of the graph nodes. We train it using the cross-entropy loss.

For the consistency loss, we use the square of the difference between predictions of different heads on heterogeneous edges. Let $\mathcal{S}_\times$ be the unlabeled set of inter-category edges (positive and negative) and Z_i^k as the final representation of node i using group's k embedding layer g_k.

Link prediction experiments with GCN We use the exact same baselines and variants of our method, as used in the GLACE experiments. For GCN, we did not test on the full Cora dataset due to scalability issues: Cora contains 70 classes, which implies computing 70 embeddings for all nodes in each step, which overrun the GPU. The results of our experiments are presented in Tab. 4. With the exception of PubMed, our method outperforms the unsupervised baselines and the simplified variants of our method. On PubMed, the variant without the consistency term and the conventional domain adaptation outperform our full method. On CiteSeer, our unsupervised method almost matches the performance of the fully supervised method.

6. Conclusions

We present a new type of transfer learning problem. This novel setting is relevant to many real-world scenarios, such as integrating multiple datasets, wherein each dataset, information on the relation between pairs of samples is available.

The setting we explore is a specific unsupervised domain adaptation problem. The target domain is the product of two domains A and B, where there are two source domains $A \times A$ and $B \times B$. We derive a new generalization bound for this setting and develop a method that implements the terms of the bound. The result is a method that differs from the one obtained when considering this as a vanilla domain adaptation problem: first, multiple source classifiers are used; second, the source classifiers are required to be consistent; third, the domain confusion term takes a specific form.

We present experiments on two very different learning problems: virus-host protein interactions prediction based on their amino-acid sequences, and inter-category citation prediction based on graph algorithms. In both, we explore the different state of the art representations and neural frameworks and show that the proposed method consistently outperforms the domain adaptation baseline, as well as other semi-supervised methods.

References

Ahmed, I., Witbooi, P. and Christoffels, A. (2018), 'Prediction of human-bacillus anthracis protein–protein interactions using multi-layer neural network', *Bioinformatics* **34**(24), 4159–4164.

Ando, R. K. and Zhang, T. (2005), 'A framework for learning predictive structures from multiple tasks and unlabeled data', *Journal of Machine Learning Research* **6**(Nov), 1817–1853.

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

Ben-David, S., Blitzer, J., Crammer, K. and Pereira, F. (2006), Analysis of representations for domain adaptation, *in* 'Proceedings of the 19th International Conference on Neural Information Processing Systems', NIPS'06, MIT Press, Cambridge, MA, USA, p. 137–144.

Blitzer, J., McDonald, R. and Pereira, F. (2006), Domain adaptation with structural correspondence learning, *in* 'EMNLP'.

Brenner, S. E., Chothia, C. and Hubbard, T. J. (1998), 'Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships', *PNAS* **95**(11), 6073–6078.

Chelba, C. and Acero, A. (2006), 'Adaptation of maximum entropy capitalizer: Little data can help a lot', *Computer Speech & Language* **20**(4), 382–399.

Cook, H. V., Doncheva, N. T., Szklarczyk, D., Von Mering, C. and Jensen, L. J. (2018), 'Viruses. string: A virus-host protein-protein interaction database', *Viruses* **10**(10), 519.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019), BERT: Pre-training of deep bidirectional transformers for language understanding, *in* 'ACL'.

Elnaggar, A., Heinzinger, M. et al. (2020), 'Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing', *arXiv preprint arXiv:2007.06225* .

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. and Lempitsky, V. (2016), 'Domain-adversarial training of neural networks', *The journal of machine learning research* **17**(1), 2096–2030.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014), Generative adversarial nets, *in* 'NIPS'.

Hettige, B., Li, Y.-F., Wang, W. and Buntine, W. (2020), Gaussian embedding of large-scale attributed graphs, *in* 'Australasian Database Conference', Springer, pp. 134–146.

Kipf, T. N. and Welling, M. (2016), 'Semi-supervised classification with graph convolutional networks', *arXiv preprint arXiv:1609.02907* .

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. and Soricut, R. (2019), 'Albert: A lite bert for self-supervised learning of language representations', *arXiv preprint arXiv:1909.11942* .

Long, M., Cao, Y., Wang, J. and Jordan, M. I. (2015), 'Learning transferable features with deep adaptation networks'.

Long, M., Zhu, H., Wang, J. and Jordan, M. I. (2017), Deep transfer learning with joint adaptation networks, *in* 'International conference on machine learning', PMLR, pp. 2208–2217.

Mansour, Y., Mohri, M. and Rostamizadeh, A. (2009), 'Domain adaptation: Learning bounds and algorithms'.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P. and Song, Y. (2019), Evaluating protein transfer learning with tape, *in* 'NeurIPS'.

Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J. and Fergus, R. (2019), 'Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences', *bioRxiv* p. 622803.

Stumpf, M. P., Thorne, T., de Silva, E., Stewart, R., An, H. J., Lappe, M. and Wiuf, C. (2008), 'Estimating the size of the human interactome', *Proceedings of the National Academy of Sciences* **105**(19), 6959–6964.

Szklarczyk, D., Franceschini, A. et al. (2015), 'STRING v10: protein–protein interaction networks, integrated over the tree of life', *Nucleic Acids Research* **43**.

Taigman, Y., Polyak, A. and Wolf, L. (2017), Unsupervised cross-domain image generation, *in* 'ICLR'.

Tzeng, E., Hoffman, J., Saenko, K. and Darrell, T. (2017), 'Adversarial discriminative domain adaptation', *CVPR* .

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017), Attention is all you need, *in* 'NeurIPS'.

Yang, X., Yang, S., Li, Q., Wuchty, S. and Zhang, Z. (2020), 'Prediction of human-virus protein-protein interactions through a sequence embedding-based machine learning method', *Computational and structural biotechnology journal* **18**, 153–161.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R. and Le, Q. V. (2019), Xlnet: Generalized autoregressive pretraining for language understanding, *in* 'NeurIPS'.