Cross-Domain Few-Shot Graph Classification

Kaveh Hassani

Autodesk AI Lab, Toronto, Canada
kaveh.hassani@autodesk.com

Abstract

We study the problem of few-shot graph classification across domains with nonequivalent feature spaces by introducing three new cross-domain benchmarks constructed from publicly available datasets. We also propose an attention-based graph encoder that uses three congruent views of graphs, one contextual and two topological views, to learn representations of task-specific information for fast adaptation, and task-agnostic information for knowledge transfer. We run exhaustive experiments to evaluate the performance of contrastive and meta-learning strategies. We show that when coupled with metric-based meta-learning frameworks, the proposed encoder achieves the best average meta-test classification accuracy across all benchmarks.

Introduction

In Few-shot learning a model learns to adapt to novel categories from a few labeled samples. Common practices such as using augmentation, regularization, and pretraining may help in such a data-scarce regime, but cannot circumvent the problem. Inspired by human learning (Lake, Salakhutdinov, and Tenenbaum 2015), meta-learning (Hospedales et al. 2020) leverages a distribution of similar tasks (Satorras and Estrach 2018) to accumulate transferable knowledge from prior experience which then can serve as a strong inductive bias for fast adaption to down-stream tasks (Sung et al.). In meta-learning, rapid learning occurs within a task whereas the knowledge about changes in task structure is gradually learned across tasks (Huang and Zitnik 2020). Examples of such learned knowledge are embedding functions (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Satorras and Estrach 2018; Sung et al.), initial parameters (Finn, Abbeel, and Levine 2017; Raghu et al. 2020), optimization strategies (Li et al. 2017), or models that can directly map training samples to network weights (Garnelo et al. 2018; Mishra et al. 2018).

A fundamental assumption in meta-learning is that tasks in meta-training and meta-testing phases are sampled from the same distribution, i.e., tasks are i.i.d. However, in many real-world applications, collecting tasks from the same distribution is infeasible. Instead, there are datasets available from the same modality but different domains. In transfer learning, this is referred as heterogeneous transfer learning where the feature/label spaces between the source and target domains are nonequivalent and are generally non-overlapping (Day and Khoshgoftaar 2017). It is observed that when there is a large shift between source and target domains, meta-learning algorithms are outperformed by pretraining/fine-tuning methods (Chen et al. 2019b).

A few work in computer vision addresses cross-domain few-shot learning by meta-learning the statistics of normalization layers (Tseng et al. 2020; Du et al. 2021). These methods are limited to natural images that still contain a high degree of visual similarity (Guo et al. 2020). Cross-domain learning is more crucial on variable-size order-invariant graph-structured data. Labeling graphs is more challenging compared to other common modalities because they usually represent concepts in specialized domains such as biology where labeling through wet-lab experiments is resource-intensive (Hu et al. 2020b) and labeling them procedurally using domain knowledge is costly (Sun et al. 2020). Furthermore, nonequivalent and non-overlapping feature spaces is common across graph datasets in addition to shifts on marginal/conditional probability distributions. As an example, one may have access to small molecule datasets where each dataset uses a different set of features to represent the molecules (Day and Khoshgoftaar 2017).

To the best of our knowledge, this is the first work pertaining cross-domain few-shot learning on graphs. To address this problem, we design a task-conditioned encoder that learns to attend to different representations of a task. Our contributions are as follows:

- we introduce three benchmarks for cross-domain few-shot graph classification and perform exhaustive experiments to evaluate the performance of supervised, contrastive, and meta-learning strategies.

- We propose a graph encoder that learns to attend to three congruent views of graphs, one contextual and two topological views, to learn representations of task-specific information for fast adaptation, and task-agnostic information for knowledge transfer.

- We show that when coupled with metric-based meta-learning frameworks, the proposed encoder achieves the best average meta-testing classification accuracy across all three benchmarks.

Related Work

Graph Neural Networks (GNNs) are a class of deep models that combine the expressive power of graphs in modeling interactions with the unparalleled capacity of deep learning in learning representations. GNNs learn node representations over order-invariant and variable-size data, structured as graphs, through an iterative process of transferring, transforming, and aggregating the representations from topological neighbors. The learned representations are then summarized into a graph-level representation (Li et al. 2015a; Gilmer et al. 2017; Kipf and Welling 2017; Veličković et al. 2018; Xu et al. 2019; Khasahmadi et al. 2020; Hassani and Khasahmadi 2020). GNNs are applied to non-Euclidean data such as point clouds (Hassani and Haley 2019), robot designs (Wang et al. 2019), physical processes (Sanchez-Gonzalez et al. 2020), molecules (Duvenaud et al. 2015), social networks (Kipf and Welling 2017), and knowledge graphs (Vivona and Hassani 2019). For an overview on GNNs see (Zhang, Cui, and Zhu 2020).

Meta-Learning on Graphs is an under-addressed problem. A few works including Meta-GNN (Zhou et al. 2019), G-Meta (Huang and Zitnik 2020), and GFL (Yao et al. 2020) focus on few-shot node classification via meta gradients, whereas other work such as MetaR (Chen et al. 2019a), GMatching (Xiong et al. 2018), and Meta-Graph (Bose et al. 2019) focus on few-shot link prediction and generalization over relations in knowledge graphs. More relevant to this study, MetaSpecGraph (Chauhan, Nathani, and Kaul 2020) uses a super-class prototypical network for few-shot graph classification where super-classes are constructed by clustering graphs using spectrum of normalized Laplacian. All these works assume i.i.d distribution across the tasks which makes them infeasible for real-world applications. On the other hand, we for the first time address the problem of few-shot graph classification within the cross-domain setting.

Domain Adaptation is a type of transductive transfer learning in which the source and target classes are equivalent but the domains are different, and the goal is to reduce the distribution shift between the two domains (Zhuang et al. 2020; Wilson and Cook 2020). Most work addresses it by reducing the shift in input, feature, or output spaces using adversarial training (Tzeng et al. 2017; Chen et al. 2018; Hoffman et al. 2018). A major drawback of these methods is that they require access to unlabeled samples from the target domain during the training which makes them less practical (Tseng et al. 2020). For a review see (Wilson and Cook 2020).

Domain Generalization aims to generalize from a set of seen domains to unseen domains without knowledge about the target distribution during training (Dou et al. 2019; Tseng et al. 2020). Different strategies such as adversarial data augmentation (Volpi et al. 2018; Shankar et al. 2018), extracting task-specific domain-invariant features (Li et al. 2018b; Akuzawa, Iwasawa, and Matsuo 2019), or learning-to-learn strategies to simulate the generalization process (Dou et al. 2019; Li et al. 2018a, 2019) are used to tackle this problem. Domain generalization is more challenging in few-shot setting. On proper *cross-domain few-shot* benchmarks, meta-learning methods are outperformed by simple fine-tuning and even in some cases by networks with random weights (Guo et al. 2020). Only a few methods are introduced for cross-domain few-shot learning. Feature-wise transform (FWT) (Tseng et al. 2020) uses feature-wise transform layers to encourage learning representations with an improved ability to generalize whereas MetaNorm (Du et al. 2021) learns to infer adaptive statistics for batch normalization. We use a combination of learning task-specific domain-invariant features and inductive normalization layers to achieve domain generalization in cross-domain few-shot graph classification setting.

Problem Formulation

A *domain* $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}, P_{\mathcal{X}, \mathcal{Y}}\}$ is defined as a joint distribution $P_{\mathcal{X}, \mathcal{Y}}$ over the feature space $\mathcal{X}$ and label space $\mathcal{Y}$. We denote the marginal distribution over feature space as $P_\mathcal{X}$ and a parametric model over joint distribution as $f_\theta : \mathcal{X} \longmapsto \mathcal{Y}$ where $f_\theta(x) = \{P(y_k|x, \theta) \mid y_k \in \mathcal{Y}\}$. The model parameters are learned by minimizing the expected error over loss function $\mathcal{L}$: $\mathbb{E}_{(x,y)\sim P_{\mathcal{X}, \mathcal{Y}}}[\mathcal{L}(f_\theta(x), y)]$. In cross-domain few-shot learning, it is assumed that two domains exist: source domain $\mathcal{D}_S$ and target domain $\mathcal{D}_T$ such that their marginal distributions are different $P_{\mathcal{X}_S} \neq P_{\mathcal{X}_T}$, and $\mathcal{Y}_S$ and $\mathcal{Y}_T$ are disjoint. The source domain is available during meta-training phase whereas the target domain is only seen in meta-testing phase. During meta-training, tasks $\{\mathcal{T}_i | i = 1...N\}$ are drawn from a distribution of tasks defined over $\mathcal{D}_S$, i.e., $\mathcal{T}_i \sim P_S(\mathcal{T})$, where each task consists of two non-overlapping small datasets: $D_i^{support} = \{(x_j, y_j)\}_{j=1}^{k \times n}, D_i^{query} = \{(x_j, y_j)\}_{j=1}^{k \times m}$. k denotes the number of sampled classes and n and m are number of examples per category, i.e., *k-way n-shot learning*.

In the meta-training phase, the model's error on the support set provides *task-level* update signals, while error on the query set after the model adapts to the support set, provides *meta-level* update signals. During the meta-testing stage, the model is expected to quickly adapt to task $\mathcal{T}_j \sim P_T(\mathcal{T})$ by only accessing the support set for that task. The tasks in the meta-testing phase are sampled from $\mathcal{D}_T$ and $P_S(\mathcal{T}) \neq P_T(\mathcal{T})$. It is noteworthy that learning in cross-domain few-shot setting is more difficult than learning in the transductive setting of traditional meta-learning where $\mathcal{D}_S = \mathcal{D}_T$ and as a result $P_{\mathcal{X}_S} = P_{\mathcal{X}_T}$ and $P_S(\mathcal{T}) = P_T(\mathcal{T})$.

In cross-domain few-shot learning, it is assumed that: (1) there is a domain shift between the source and target domains ($P_{\mathcal{X}_S} \neq P_{\mathcal{X}_T}$), and (2) the feature spaces are equivalent between domains ($\mathcal{X}_S = \mathcal{X}_T$). This makes sense in computer vision where various image acquisition methods such as satellite images, dermatology images, and radiology images share a similar feature space with natural images (Guo et al. 2020). However, in domains where data is represented as graphs, this assumption does not hold. For example, two molecular property prediction datasets may have different node/edge feature spaces due to the method used to generate the datasets (e.g., one may contain additional atom features such as formal charge and whether the atom is in the ring) (Hu et al. 2020a).

As such, we go beyond cross-domain few-shot learning and investigate its heterogeneous variant for graph classifi-

cation. In this setting, we assume that: (1) Each task is possibly sampled from a dedicated domain different from all other tasks, either in meta-training or meta-testing phase, i.e. if there are N tasks in meta-training and M tasks in meta-testing phases, there exist $|\mathcal{D}| \leq N + M$ domains. (2) Tasks are heterogeneous, which means they may have nonequivalent non-overlapping feature spaces, i.e., different dimensions in addition to distribution differences and disjoint label spaces: $\forall i \neq j : \mathcal{X}_{\mathcal{T}_i} \neq \mathcal{X}_{\mathcal{T}_j}, P_{\mathcal{X}_{\mathcal{T}_i}} \neq P_{\mathcal{X}_{\mathcal{T}_j}}, \mathcal{Y}_{\mathcal{T}_i} \neq \mathcal{Y}_{\mathcal{T}_j}$. (3) Tasks can be grouped based on their *meta-domains* which essentially defines the conceptual domain of a task. For example, all datasets that represent social networks can be grouped under a *social network* meta-domain despite the fact that they may have different feature and label spaces. We investigate whether there exists underlying knowledge that can be transferred across these meta-domains.

Method

Graph-structured data can be analyzed from two congruent views: a contextual view and a topological view. The contextual view is based on initial node or edge features (for simplicity and without loss of generality, we only consider node features) and carries task-specific information. The topological view, on the other hand, represents topological properties of a graph which are task-agnostic and hence can be used as an anchor to align graphs from various domains in the feature space. We exploit this dual representation and explicitly disentangle them by designing dedicated encoders for each view which in return imposes the needed inductive bias to learn task-specific domain-invariant features. In a heterogeneous few-shot setting, the topological features can help with knowledge transfer across tasks whereas the contextual features can help with fast adaptation. We also use an attention mechanism that is implicitly conditioned on the tasks and learns to aggregate the learned features from the two views. We use a meta-learning strategy that simulates the generalization process by jointly learning the parameters of the encoders and the attention mechanism. As shown in Figure 1, our method consists of the following components:

- An augmentation mechanism that transforms a sampled graph into one contextual view and two topological views. The augmentations are applied to the initial node feature and graph structure.

- An encoder consisting of two dedicated GNNs, i.e., graph encoders, and an MLP for the contextual and topological views, respectively, and an attention mechanism to aggregate the learned features.

- A meta-learning mechanism to jointly learn the parameters of the dedicated encoders and attention model based on error signals from the query set.

Augmentations

Recent works on self-supervised learning on graphs suggest that contrasting graph augmentations allows encoders to learn rich node/graph representations (Hassani and Khasahmadi 2020). In this work, we are specifically interested in task-specific and domain-agnostic views of graphs to help

the meta-learner to gradually accumulate domain-agnostic knowledge while utilizing task-specific information for fast adaptation. We use both feature-space and structure-space augmentations as follows. For the contextual view, we considered three feature-space augmentations on the initial node features including: (1) Heterogeneous feature augmentation (Duan, Xu, and Tsang 2012) where the initial feature and its projection by a linear layer are concatenated and padded to a predefined dimension, (2) Deep set (Zaheer et al.) approach in which we considered the initial node feature space as a set, projecting each dimension independently to a new space using a linear layer, and aggregating them by a permutation-invariant function. This augmentation can capture the shared information across tasks with overlapping features when the alignment among the features is not available. (3) Simple padding of the features to a predetermined dimension. Surprisingly, we observed that the simplest augmentation achieves better results. We speculate this is because the tasks are not sharing overlapping features.

For the topological view, we apply one feature-space and one structure-space augmentation. In the feature-space augmentation, we replace the task-dependent node features with sinusoidal node degree encodings which allow the model to extrapolate to node degrees greater than the ones encountered during meta-training stage (Vaswani et al. 2017). Because node degrees are universal properties of graph nodes, encoding a graph with such initial features will capture task-agnostic geometric structure of the graph. We also use graph sub-sampling to keep the degree distribution in a similar order of magnitude across domains. For the structure-space augmentation, we compute graph diffusion to provide a global view of the graph's structure. We used Personalized PageRank (PPR) (Page et al. 1999), a specific instantiation of the generalized graph diffusion. We compute the eigenvalues of the diffusion matrix, sort them in a descending order, and select the top-k eigenvalues as the structural representation. We also experimented heat kernel diffusion, eigen values of normalized graph Laplacian, and shortest path matrix, and found that diffusion produced better results.

Encoders

Assume a support set $D_i^{sup} = [g_1, g_2, ..., g_N]$ of N graphs belonging to a randomly sampled task i. Augmenting each graph g produces three views: a contextual view represented as graph $g_c = (\mathbf{A}, \mathbf{X})$ where $\mathbf{A} \in \{0, 1\}^{n \times n}$ and $\mathbf{X} \in \mathbb{R}^{n \times d_x}$ denote the adjacency matrix and the task-specific node features, a topological view represented as graph $g_g = (\mathbf{A}, \mathbf{U})$ where $\mathbf{U} \in \mathbb{R}^{n \times d_u}$ denotes sinusoidal node degree encodings, and another topological view represented as vector $\mathbf{z} \in \mathbb{R}^{d_z}$ denoting the sorted eigenvalues of the corresponding diffusion matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$. Our framework allows various choices of network architecture without any constraints. For encoding graph-structured views, we opted for expressive power and adopted graph isomorphism network (GIN) (Xu et al. 2019). The kth layer of our graph encoder consists of a GIN layer followed by a feature-wise transformation layer (FWT) (Tseng et al. 2020) and a swish activation. FWT layer simulates various feature distributions un-
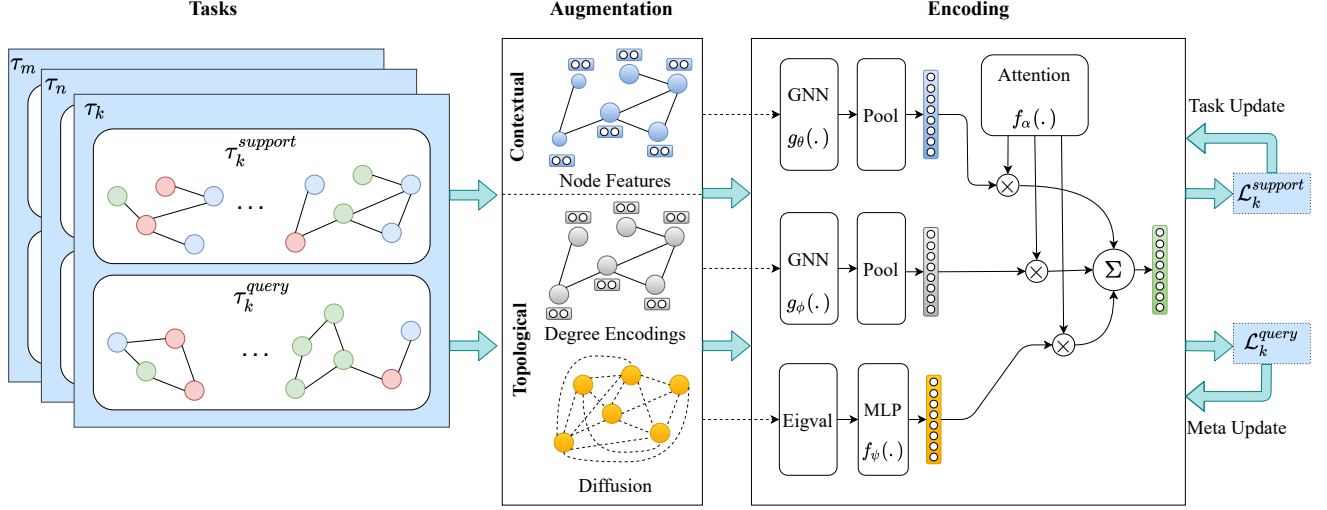
Figure 1: The proposed model for cross-domain few-shot graph classification. Graphs from sampled tasks are augmented to one contextual view and two geometric views and fed to three dedicated encoders resulting in three representations of the same graph. An attention mechanism is then used to aggregate the representations into a single graph representation. The parameters of the encoders and the attention mechanism are learned end-to-end using an arbitrary metric-based meta-learning approach.

der different domains: $h_v^{(k)} = \gamma^{(k)} \times h_v^{(k)} + \beta^{(k)}$ where $\gamma \sim \mathcal{N}(1, \text{softplus}(\theta_\gamma))$ and $\beta \sim \mathcal{N}(0, \text{softplus}(\theta_\beta))$. $\theta_\gamma, \theta_\beta$ are the standard deviations of the Gaussian distributions for sampling the affine transformation parameters.

We use a dedicated graph encoder for each view: $g_\theta(.)$: $\mathbb{R}^{n \times d_x} \times \mathbb{R}^{n \times n} \longmapsto \mathbb{R}^{n \times d_h}$ and $g_\phi(.) : \mathbb{R}^{n \times d_u} \times \mathbb{R}^{n \times n} \longmapsto \mathbb{R}^{n \times d_h}$ resulting in two sets of node representations $\mathbf{H}^x$, $\mathbf{H}^u \in \mathbb{R}^{n \times d_h}$ corresponding to the contextual and the topological views of the sampled graph. For each view, we aggregate the node representations into a graph representation using a pooling (readout) function $\mathcal{R}(.) : \mathbb{R}^{n \times d_h} \longmapsto \mathbb{R}^{d_h}$. We experimented with global soft attention pooling (Li et al. 2015b), jumping knowledge network (Xu et al. 2018), and summation and mean pooling layers, and found that they produce similar results. Therefore, we opted for simplicity and used a simple mean pooling layer. This results in two graph representations: $\mathbf{h}^x, \mathbf{h}^u \in \mathbb{R}^{d_h}$. We also feed the topological view from the eigenvalues of the graph diffusion into a projection head $f_\psi(.) : \mathbb{R}^{d_z} \longmapsto \mathbb{R}^{d_h}$, modeled as an MLP resulting in the third representation: $\mathbf{h}^z \in \mathbb{R}^{d_h}$.

To aggregate the learned representations, we feed the concatenation of the learned representations into an attention module $f_\omega(.) : \mathbb{R}^{3 \times d_h} \longmapsto \mathbb{R}^3$ that generates attention scores for each representation. The attention module is modeled as a single-layer MLP followed by a softmax function:

$$\alpha = \text{Softmax}\left(\text{ReLU}\left(\left[\mathbf{h}^x \parallel \mathbf{h}^u \parallel \mathbf{h}^z \right] \mathbf{W_1} \right) \mathbf{W_2} \right) \quad (1)$$

where $\mathbf{W_1} \in \mathbb{R}^{(3 \times h_d) \times h_d}$ and $\mathbf{W_2} \in \mathbb{R}^{h_d \times 3}$ are network parameters. The attention scores are then used to aggregate the learned features into a final graph representation.

The attention mechanism gates the representations and decides if the model should rely more on contextual or topological representations. If the samples are from a task that is similar to seen tasks, the model will pay more attention to contextual representation whereas if there is a drastic shift in feature space, the model will rely more on geometric representations. We assume that the target domain is not available during training and hence there is no information in advance about whether there are shared features among tasks. If there is, the attention module will pass the shared contextual information through, otherwise it wont attend to the contextual features and will let the learner to learn them from scratch during the meta-test adaptation phase. Hence, rather than naively throwing the information away and learning from scratch, we let the model decide if it can use the information. It is noteworthy that we are not introducing a new meta-learning framework. Instead we are introducing an encoder with attention module that can seamlessly be integrated into any meta-learning framework. As an example, we show the training procedure of the encoder within a mini-batch of tasks using prototypical approach (Snell, Swersky, and Zemel 2017) in Algorithm 1. Depending on the meta-learner, the aggregated representation can be then fed into a linear classifier or a non-parametric classifier such as a prototypical classifier.

Benchmarks

We also introduce three new few-shot graph classification benchmarks with fixed meta–train/val/test splits constructed from publicly available graph datasets. In all benchmarks, the source meta-domain consists of molecule classification tasks. We made this decision because most of the available graph classification datasets are molecule datasets and hence using them as the source meta-domain can provide suffi-

Algorithm 1: Training the proposed encoder with prototypical approach for one mini-batch of tasks. $\mathbf{A}_g, \mathbf{S}_g, \mathbf{X}_g, \mathbf{U}_g$ denote adjacency, diffusion, node features, and node degree encodings of graph g.

Input: Concatenation operator $\|$, readout function $\mathcal{R}$, eigenvalue function $\mathcal{E}$, prototype function $\mathcal{P}$, prototypical loss $\mathcal{L}$, encoders g_θ, g_ϕ, f_ψ, attention module f_ω, and meta-training task batch $\{\mathcal{T}_j | \mathcal{T}_j = D^s_{\mathcal{T}_j} \cup D^q_{\mathcal{T}_j}\}^N_{j=1}$

for $\mathcal{T}$ *in a task batch* $\{\mathcal{T}_j\}^N_{j=1}$ **do**
   $[\mathbf{H}^s, \mathbf{H}^q, \mathcal{Y}^s, \mathcal{Y}^q] \leftarrow \emptyset$
   for (g, y) *in* $D^s_{\mathcal{T}} \cup D^q_{\mathcal{T}}$ **do**
      $\mathbf{h}_x \leftarrow \mathcal{R}\left(g_\theta\left(\mathbf{A}_g, \mathbf{X}_g\right)\right)$
      $\mathbf{h}_u \leftarrow \mathcal{R}\left(g_\phi\left(\mathbf{A}_g, \mathbf{U}_g\right)\right)$
      $\mathbf{h}_z \leftarrow f_\psi\left(\mathcal{E}\left(\mathbf{S}_g\right)\right)$
      $\alpha \leftarrow f_\omega\left([\mathbf{h}_x \| \mathbf{h}_u \| \mathbf{h}_z]\right)$
      $\mathbf{h} \leftarrow \alpha_0 \mathbf{h}_x + \alpha_1 \mathbf{h}_u + \alpha_2 \mathbf{h}_z$
      if $g \in D^s_{\mathcal{T}}$ **then**
         $\mathbf{H}^s \leftarrow \mathbf{H}^s \| \mathbf{h} , \mathcal{Y}^s \leftarrow \mathcal{Y}^s \| y$
      else
         $\mathbf{H}^q \leftarrow \mathbf{H}^q \| \mathbf{h} , \mathcal{Y}^q \leftarrow \mathcal{Y}^q \| y$
      end
   end
   $\mathbf{C}_{\mathcal{T}} \leftarrow \mathcal{P}\left(\mathbf{H}^s, \mathcal{Y}^s\right) \mathbf{L}_{\mathcal{T}} \leftarrow \mathbf{L}_{\mathcal{T}} + \mathcal{L}\left(\mathbf{C}_{\mathcal{T}}, \mathbf{H}^q, \mathcal{Y}^q\right)$
end
$[\theta, \omega, \phi, \psi] \leftarrow$
   $[\theta, \omega, \phi, \psi] - \gamma \nabla_{[\theta, \omega, \phi, \psi]} \frac{1}{N \times |D^q|} \sum\limits_{j=1}^{N} \left[\mathbf{L}_{\mathcal{T}_j}\right]$

---

cient tasks during meta-training. The target meta-domains are molecule, bioinformatics, and social networks. Note that although both source and target meta-domains in the first benchmark pertain to molecule, the tasks differ in both feature and class spaces. The process of creating these benchmark was as follows.

We collected all the datasets from TUDataset (Morris et al. 2020) and OGB (Hu et al. 2020a). We kept graphs with maximum node degree of 50, maximum node feature dimension of 100, and graphs with a minimum of two nodes and one edge and ignored the rest. We also filtered out graphs with disconnected components. For graphs with more than 500 nodes, we sorted the nodes by their harmonic centrality (Boldi and Vigna 2014) in a descending order and used the sub-graphs containing the top 500 nodes. For multi-task datatsets, we drew samples from each task without replacement and split them into several single-task datasets without sharing any data samples. Because the majority of datasets in TUDataset and OGB are binary classification tasks, we opted for a "k-shot 2-way" setting and split the few remaining multi-class datasets into binary datasets by sampling without replacement. We then randomly selected 20 and 50 samples per class as support and query sets, respectively.

For the second and third benchmarks, we used the processed datasets from bioinformatics and social network categories as the meta-testing tasks. For the first benchmark, we split the tasks such that if a task is originated from a multi-

task or a multi-class dataset, it is allocated to a split with all of the tasks that also originated from the same dataset. The statistics of the proposed benchmarks are shown in Table 1. We believe these benchmarks will help the community to drive further advances in heterogeneous meta-learning.

Experimental Results

We exhaustively perform empirical evaluations to answer the following questions: (1) what is the the empirical upper bound of the classification accuracy on the meta-test set for the benchmarks? (2) does any knowledge transfer occur across meta-domains? If not, does negative transfer occur? (3) How well does pre-training based on contrastive methods perform? (4) How do metric-based meta-learning methods perform compared to optimization-based methods? (5) what is the effect of using the proposed encoder?

To estimate the empirical upper bound of the classification accuracy, we approximated how fully-supervised models with access to all data would perform. To accomplish this, we used all the data available within the datasets from which the few-shot tasks in the meta-testing phase are sampled and trained one classifier per meta-testing task in a fully supervised fashion. We evaluated the classifiers on the same query sets used in the few-shot setting, i.e., query samples are fixed, but the support set is replaced with all data available in the original dataset. To investigate knowledge transfer, we trained independent classifiers on the support set of the meta-testing tasks and evaluated them on their corresponding query set. Negative transfer occurs when the performance on meta-testing tasks is negatively effected by knowledge transferred from meta-training tasks (Pan and Yang 2009). Therefore, if the independently trained models outperform models that utilize knowledge transfer, it implies negative transfer. We also investigated two strategies to utilize knowledge transfer: pre-training using contrastive methods and learning-to-learn strategies. For contrastive methods, we aggregated support and query samples from the meta-training tasks into a unified dataset, pretrained a GNN, and then fine-tuned it to the meta-test tasks. Finally, we investigated the effectiveness of the proposed encoder by comparing the performance of the meta-learning models with and without using it.

We report the mean classification accuracy with standard deviation over query samples of the meta-testing tasks after ten runs. For estimating empirical upper bound accuracy, we used a GIN (Xu et al. 2019) classifier. For independent classifiers, we used GCN (Kipf and Welling 2017), GAT (Veličković et al. 2018), and GIN layers. For contrastive methods, we used GCC (Qiu et al. 2020), GSFE (Hu et al. 2019), InfoGraph (Sun et al. 2020), and MV-GRL (Hassani and Khasahmadi 2020). During the meta-testing phase, we first fine-tuned the encoder on the support set using the same contrastive approach, and then trained an MLP on the graph representations. We found that concatenating the graph representation produced by the contrastive method with a mean pooling of initial node features improves the performance. For meta-learning methods, we used three metric-based techniques including matching network (Vinyals et al. 2016), prototypical network

Table 1: Statistics of the proposed benchmarks for heterogeneous cross-domain few-shot graph classification.

META-DOMAIN		|TASK|		AVG. ON TARGET			|SHOT|	|CLASS|	|QUERY|
SOURCE	TARGET	TRAIN	DEV	TEST	NODE	EDGE	FEATURE			
MOLECULES	MOLECULES	169	5	18	26.6 ± 15.7	28.6 ± 16.6	18.1 ± 18.7	1,5,10,20	2	50
MOLECULES	BIOINFORMATICS	187	5	24	79.2 ± 58.5	406.6 ± 300.3	19.8 ± 15.1	1,5,10,20	2	50
MOLECULES	SOCIAL NETWORKS	187	5	12	54.1 ± 58.8	98.1 ± 117.9	0	1,5,10,20	2	50

Table 2: Mean and Standard Deviation of meta-test accuracy on Bioinformatics benchmark after ten runs.

	METHOD	1-SHOT	5-SHOT	10-SHOT	20-SHOT
	EMPIRICAL UPPER BOUND	66.78 ± 10.30			
SUP	GCN	54.88 ± 7.55	55.05 ± 8.85	55.03 ± 8.91	54.99 ± 8.82
	GAT	54.75 ± 8.85	54.69 ± 8.90	54.76 ± 8.97	54.63 ± 8.94
	GIN	55.37 ± 9.83	55.52 ± 9.79	55.47 ± 9.89	55.52 ± 9.65
UNS	INFOGRAPH	54.00 ± 6.65	53.67 ± 7.35	54.42 ± 6.41	54.96 ± 7.63
	MVGRL	57.12 ± 7.75	57.25 ± 9.04	57.17 ± 8.01	57.54 ± 8.06
	GSFE	52.84 ± 6.71	52.96 ± 7.82	53.06 ± 7.64	53.16 ± 7.87
	GCC	53.11 ± 6.51	53.17 ± 6.43	53.18 ± 7.18	53.35 ± 7.64
META	MATCHNET	54.83 ± 7.66	55.62 ± 7.60	55.92 ± 6.67	56.04 ± 7.78
	PROTONET	54.71 ± 8.86	55.75 ± 7.84	55.96 ± 6.73	55.50 ± 9.65
	RELATIONNET	54.93 ± 8.55	55.92 ± 8.69	56.02 ± 7.69	56.15 ± 7.81
	MAML	53.83 ± 9.62	54.46 ± 6.77	54.50 ± 8.77	54.79 ± 8.90
	METASGD	53.83 ± 8.79	54.21 ± 7.70	54.67 ± 9.72	54.71 ± 7.90
	METASPECGRAPH	$-$	55.47 ± 7.79	55.82 ± 8.91	55.97 ± 8.89
	MATCHNET + OUR ENCODER	$\mathbf{59.14 \pm 7.00}$	$\mathbf{59.19 \pm 9.77}$	59.22 ± 8.72	59.56 ± 6.97
	PROTONET + OUR ENCODER	57.17 ± 7.76	57.58 ± 8.89	57.79 ± 8.76	58.17 ± 7.88
	RELATIONNET + OUR ENCODER	58.83 ± 8.03	58.83 ± 9.68	$\mathbf{59.29 \pm 7.87}$	$\mathbf{59.82 \pm 7.93}$
	MAML + OUR ENCODER	56.00 ± 8.74	56.21 ± 7.76	56.37 ± 8.81	57.04 ± 7.85
	METASGD + OUR ENCODER	55.08 ± 8.67	56.12 ± 8.23	57.08 ± 8.86	57.33 ± 8.86

Table 3: Mean and Standard Deviation of meta-test accuracy on Social Networks benchmark after ten runs.

	METHOD	1-SHOT	5-SHOT	10-SHOT	20-SHOT
	EMPIRICAL UPPER BOUND	72.35 ± 12.38			
SUP	GCN	60.51 ± 10.54	60.54 ± 10.18	60.33 ± 10.19	60.35 ± 10.52
	GAT	61.32 ± 10.31	61.37 ± 10.20	61.17 ± 10.17	61.51 ± 10.13
	GIN	62.11 ± 10.11	62.98 ± 10.04	63.27 ± 10.05	63.24 ± 9.28
UNS	INFOGRAPH	61.92 ± 9.84	62.25 ± 7.12	62.58 ± 8.57	62.58 ± 7.32
	MVGRL	63.00 ± 10.70	63.75 ± 11.17	63.25 ± 11.69	63.75 ± 11.99
	GSFE	60.38 ± 9.74	60.45 ± 9.62	60.46 ± 9.95	60.55 ± 9.11
	GCC	60.61 ± 9.55	60.73 ± 9.74	60.81 ± 9.61	60.98 ± 9.97
META	MATCHNET	62.25 ± 9.14	62.42 ± 9.98	62.92 ± 9.22	63.33 ± 9.92
	PROTONET	60.50 ± 9.05	61.25 ± 10.01	61.75 ± 9.06	63.50 ± 9.97
	RELATIONNET	61.25 ± 10.04	61.08 ± 10.25	61.83 ± 10.17	62.00 ± 10.17
	MAML	58.33 ± 9.05	58.75 ± 10.84	59.00 ± 9.87	60.17 ± 10.99
	METASGD	59.25 ± 10.15	59.33 ± 9.50	59.83 ± 9.89	60.25 ± 9.88
	METASPECGRAPH	$-$	62.55 ± 9.79	62.74 ± 9.91	63.73 ± 9.89
	MATCHNET + OUR ENCODER	$\mathbf{67.40 \pm 10.37}$	$\mathbf{67.71 \pm 9.24}$	$\mathbf{68.34 \pm 9.09}$	68.90 ± 9.18
	PROTONET + OUR ENCODER	66.42 ± 10.21	66.92 ± 10.55	67.00 ± 9.13	67.50 ± 10.46
	RELATIONNET + OUR ENCODER	66.75 ± 9.89	67.08 ± 10.80	67.33 ± 10.33	$\mathbf{69.67 \pm 10.05}$
	MAML + OUR ENCODER	62.25 ± 10.12	64.75 ± 10.79	66.58 ± 10.93	66.83 ± 10.21
	METASGD + OUR ENCODER	61.92 ± 9.56	63.17 ± 10.51	64.42 ± 10.16	64.83 ± 10.06

(Snell, Swersky, and Zemel 2017), and relation network (Sung et al.), and two optimization-based methods including MAML (Finn, Abbeel, and Levine 2017) and MetaSGD (Li et al. 2017). For optimization-based methods, we used a linear classifier in both the meta-training and meta-testing phases whereas for metric-based methods, we found that using a cosine distance classifier (Chen et al. 2019b) for adaptation achieves better performance. We trained the meta-learning models with and without our proposed encoder. We also trained a specialized meta-learning approach for graph classification that uses spectral information (Chauhan, Nathani, and Kaul 2020). The results are shown in Tables 2 and 3 (for Molecule benchmark results, hyper-parameters and other details refer to Appendix).

Results suggest that: (1) On all three benchmarks, there exists underlying knowledge that can be transferred. This is validated by the observation that both meta-learning and contrastive approaches outperform naive classifiers. (2) Contrastive approaches are competitive with meta-learning methods without the proposed encoder. As an example, on 20-shot bioinformatics benchmark, MVGRL outperforms the best performing meta-learning method by 1.57% absolute accuracy. (3) Coupling metric-based meta-learning methods with our proposed encoder significantly enhances performance. As an instance, on 1-shot setting, the best meta-learning methods coupled with our encoder outperform the best results achieved by regular meta-learning methods by 3.28%, 4.29%, and 5.17% absolute accuracy on molecules, bioinformatics, and social network benchmarks, respectively. (4) RelationNet coupled with our encoder and only trained with 20 examples is only 4.46%, 6.96%, and 2.68% less accurate than fully-supervised models trained on all available data of molecules, bioinformatics, and social network benchmarks, respectively. Note that some of these datasets have tens of thousands training samples. (5) We get the most improvement when we transfer knowledge from molecular meta-training to social network meta-testing. This is because social network tasks do not contain any initial node features, and hence classifying them completely depends on task-agnostic geometric features. This suggests that our encoder is able to learn expressive geometric representations on one domain and generalize to another domain.

Ablation Study

Effect of Views. To investigate the effect of views and the proposed encoder, we run the experiments with: (1) Only node features $\mathbf{X}$ which is equivalent to traditional meta-learning, (2) a combination of node features $\mathbf{X}$ and node degree encodings $\mathbf{U}$, (3) a combination of node features $\mathbf{X}$ and eigenvalues of diffusion matrix $\mathbf{Z}$, and (4) combination of all the features. The result (see Tables 6-8 in Appendix) suggest that: (1) both topological views contribute complementary enhancements to the performance, and (2) the proposed encoder has a better performance when coupled with metric-based methods compared to optimization-based methods. Also we observe that the diffusion based view has a slightly larger effect on performance. We speculate this is because of the inductive bias that diffusion provides (it directly encodes graph encoding from global information).

Why topological properties are task-agnostic? Node degree distributions are different across domains (e.g., social networks and molecules). We are addressing this by a few tricks: (1) we use sinusoidal node degree encodings which allow the model to extrapolate to unseen node degrees, (2) we use graph sub-sampling to keep the degree distribution in a similar order of magnitude, and most importantly, (3) we are interleaving FWT and GNN layers to address the distribution shift. This allows us to first relax the heterogeneity by relying on topological signals and then use the mentioned tricks to control the distribution shifts that may occur. It is noteworthy that we found the encodings from diffusion to be less sensitive to the shifts in graph size and node degrees distributions.

Effect of Meta-Learning Approach. We considered optimization-based and metric-based approaches. Results suggest that without using our encoder, metric-based approaches are competitive with contrastive methods and also show some degree of knowledge transfer, whereas optimization-based approaches are outperformed by independent classifiers. That suggests that optimization-based approaches lead to negative knowledge transfer, confirming the observation in (Guo et al. 2020). When coupled with our encoder, optimization-based approaches become competitive with contrastive methods and show some knowledge transfer but are still outperformed by metric-based approaches (see Appendix). As suggested in (Huang and Zitnik 2020), this is likely because metric-based approaches leverage the inductive bias between representation and labels to effectively propagate scarce label information.

Effect of Task Encoding. We attempted to explicitly condition the attention module on the task encodings as follows. We pretrained an encoder using MVGRL and encode the support set of each task. We then aggregated the representation into a task encoding and stored meta-training task encodings within a memory-bank. We trained the attention module by feeding it with the graph views, the task encodings, and the encodings of the top-3 similar tasks. During meta-testing, we retrieve the top-3 similar tasks from the memory bank. Surprisingly, we found that using task encodings produces nearly identical results to not conditioning the attention module on task encodings.

Conclusion

We investigated the problem of cross-domain few-shot graph classification by introducing three new benchmarks. We performed exhaustive experiments using independent classifiers, contrastive methods, and meta-learning strategies. We also introduced a simple yet powerful multi-view graph encoder with an attention-based aggregation mechanism for better knowledge transfer and adaptation. We showed that metric-based meta-learning approaches coupled with the proposed encoder achieve the best performance across all benchmarks. We also showed that optimization-based meta learning methods struggle in cross-domain setting. In future work, we plan to investigate black-box and hybrid adaptations of meta-learning strategies.

References

Akuzawa, K.; Iwasawa, Y.; and Matsuo, Y. 2019. Domain Generalization via Invariant Representation under Domain-Class Dependency.

Boldi, P.; and Vigna, S. 2014. Axioms for centrality. *Internet Mathematics*, 10(3-4): 222–262.

Bose, A. J.; Jain, A.; Molino, P.; and Hamilton, W. L. 2019. Meta-Graph: Few Shot Link Prediction via Meta Learning. *arXiv preprint arXiv:1912.09867*.

Chauhan, J.; Nathani, D.; and Kaul, M. 2020. Few-Shot Learning on Graphs via Super-Classes Based on Graph Spectral Measures. In *International Conference on Learning Representations*.

Chen, M.; Zhang, W.; Zhang, W.; Chen, Q.; and Chen, H. 2019a. Meta Relational Learning for Few-Shot Link Prediction in Knowledge Graphs. In *Conference on Empirical Methods in Natural Language Processing*, 4208–4217.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C. F.; and Huang, J.-B. 2019b. A Closer Look at Few-shot Classification. In *International Conference on Learning Representations*.

Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; and Van Gool, L. 2018. Domain adaptive faster r-cnn for object detection in the wild. In *IEEE Conference on computer vision and pattern recognition*, 3339–3348.

Day, O.; and Khoshgoftaar, T. M. 2017. A survey on heterogeneous transfer learning. *Journal of Big Data*, 4(1): 29.

Dou, Q.; de Castro, D. C.; Kamnitsas, K.; and Glocker, B. 2019. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, 6450–6461.

Du, Y.; Zhen, X.; Shao, L.; and Snoek, C. G. M. 2021. MetaNorm: Learning to Normalize Few-Shot Batches Across Domains. In *International Conference on Learning Representations*.

Duan, L.; Xu, D.; and Tsang, I. W. 2012. Learning with augmented features for heterogeneous domain adaptation. In *International Conference on Machine Learning*, 667–674.

Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; and Adams, R. P. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, 2224–2232.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*, 1126–1135.

Garnelo, M.; Rosenbaum, D.; Maddison, C.; Ramalho, T.; Saxton, D.; Shanahan, M.; Teh, Y. W.; Rezende, D.; and Eslami, S. A. 2018. Conditional Neural Processes. In *International Conference on Machine Learning*, 1704–1713.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, 1263–1272.

Guo, Y.; Codella, N. C.; Karlinsky, L.; Codella, J. V.; Smith, J. R.; Saenko, K.; Rosing, T.; and Feris, R. 2020. A broader study of cross-domain few-shot learning. In *European Conference on Computer Vision*, 124–141. Springer.

Hassani, K.; and Haley, M. 2019. Unsupervised Multi-Task Feature Learning on Point Clouds. In *International Conference on Computer Vision*, 8160–8171.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive Multi-View Representation Learning on Graphs. In *Proceedings of the 37th International Conference on Machine Learning*, 4116–4126.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, 1989–1998.

Hospedales, T.; Antoniou, A.; Micaelli, P.; and Storkey, A. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020a. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2020b. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*.

Hu, Z.; Fan, C.; Chen, T.; Chang, K.-W.; and Sun, Y. 2019. Pre-training graph neural networks for generic structural feature extraction. *arXiv preprint arXiv:1905.13728*.

Huang, K.; and Zitnik, M. 2020. Graph meta learning via local subgraphs. *Advances in Neural Information Processing Systems*.

Khasahmadi, A. H.; Hassani, K.; Moradi, P.; Lee, L.; and Morris, Q. 2020. Memory-Based Graph Networks. In *International Conference on Learning Representations*.

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

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266): 1332–1338.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. 2018a. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Li, Y.; Tarlow, D.; Brockschmidt, M.; and Zemel, R. 2015a. Gated graph sequence neural networks. In *International Conference on Learning Representations*.

Li, Y.; Tarlow, D.; Brockschmidt, M.; and Zemel, R. 2015b. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

Li, Y.; Tian, X.; Gong, M.; Liu, Y.; Liu, T.; Zhang, K.; and Tao, D. 2018b. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision*, 624–639.

Li, Y.; Yang, Y.; Zhou, W.; and Hospedales, T. 2019. Feature-Critic Networks for Heterogeneous Domain Generalization. In *International Conference on Machine Learning*, 3915–3924.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2018. A Simple Neural Attentive Meta-Learner. In *International Conference on Learning Representations*.

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond*.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Pan, S. J.; and Yang, Q. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. *GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training*, 1150–1160.

Raghu, A.; Raghu, M.; Bengio, S.; and Vinyals, O. 2020. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. In *International Conference on Learning Representations*.

Sanchez-Gonzalez, A.; Godwin, J.; Pfaff, T.; Ying, R.; Leskovec, J.; and Battaglia, P. 2020. Learning to Simulate Complex Physics with Graph Networks. In *Proceedings of the 37th International Conference on Machine Learning*, 8459–8468.

Satorras, V. G.; and Estrach, J. B. 2018. Few-Shot Learning with Graph Neural Networks. In *International Conference on Learning Representations*.

Shankar, S.; Piratla, V.; Chakrabarti, S.; Chaudhuri, S.; Jyothi, P.; and Sarawagi, S. 2018. Generalizing Across Domains via Cross-Gradient Training. In *International Conference on Learning Representations*.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, 4077–4087.

Sun, F.-Y.; Hoffman, J.; Verma, V.; and Tang, J. 2020. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *International Conference on Learning Representations*.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. ???? Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tseng, H.-Y.; Lee, H.-Y.; Huang, J.-B.; and Yang, M.-H. 2020. Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation. In *International Conference on Learning Representations*.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *IEEE conference on computer vision and pattern recognition*, 7167–7176.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29: 3630–3638.

Vivona, S.; and Hassani, K. 2019. Relational Graph Representation Learning for Open-Domain Question Answering. *Advances in Neural Information Processing Systems, Graph Representation Learning Workshop*.

Volpi, R.; Namkoong, H.; Sener, O.; Duchi, J. C.; Murino, V.; and Savarese, S. 2018. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems*, 5334–5344.

Wang, T.; Zhou, Y.; Fidler, S.; and Ba, J. 2019. Neural Graph Evolution: Automatic Robot Design. In *International Conference on Learning Representations*.

Wilson, G.; and Cook, D. J. 2020. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology*, 11(5): 1–46.

Xiong, W.; Yu, M.; Chang, S.; Guo, X.; and Wang, W. Y. 2018. One-Shot Relational Learning for Knowledge Graphs. In *Conference on Empirical Methods in Natural Language Processing*, 1980–1990.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *International Conference on Machine Learning*, 5453–5462.

Yao, H.; Zhang, C.; Wei, Y.; Jiang, M.; Wang, S.; Huang, J.; Chawla, N.; and Li, Z. 2020. Graph few-shot learning via knowledge transfer. In *AAAI Conference on Artificial Intelligence*, 6656–6663.

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R. R.; and Smola, A. J. ???? Deep sets. In *Advances in neural information processing systems*.

Zhang, Z.; Cui, P.; and Zhu, W. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Geng, J. 2019. Meta-GNN: On Few-shot Node Classification in Graph Meta-learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2357–2360.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1): 43–76.