Transferability Prediction for Model Recommendation: A Graph Learning based Approach

Xiangyu Chen, Yanru Wu, Jian Xu, Yang Li
yangli@sz.tsinghua.edu.cn
Tsinghua Shenzhen International Graduate School, Tsinghua University
Shenzhen, Guangdong, China

ABSTRACT

Transfer learning has emerged as a popular approach to improving the performance of target tasks using knowledge from related source models. However, it remains challenging to select the most suitable pre-trained models, especially with multiple heterogeneous model architectures. As a solution to this scenario, in this paper, we formulate the transferability prediction problem as a bipartite graph learning problem, and propose a source data-free transferability prediction method. Leveraging task metadata as node information, a customized Graph Attention Network (GAT) is employed to predict the transferability among tasks, i.e., the graph edges. By learning low-dimensional task embeddings and predicting edges in the latent space, our method effectively captures intrinsic task relationships and infers transferability for unseen task pairs. Experimental results on general image classification tasks and segmentation tasks in autonomous driving scenarios demonstrate the effectiveness of our method, showcasing improvements of 15% and 12% compared to state-of-the-art methods, respectively. The code is available in https://anonymous.4open.science/r/ModelRecommend-9DBB.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

KEYWORDS

Transfer Learning, Graph Neural Network, Task Embeddings

ACM Reference Format:

Xiangyu Chen, Yanru Wu, Jian Xu, Yang Li. 2024. Transferability Prediction for Model Recommendation: A Graph Learning based Approach. In . ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The development of deep learning methods has led to the extensive accumulation of pre-trained models, such as those found in repositories like Hugging Face [1], PyTorch Hub [2] and TensorFlow Hub [3]. Popularized by this trend, The "pre-train then fine-tune" paradigm of transfer learning has gained widespread adoption [27, 28, 32]. This line of methods leverages the embedded knowledge within pre-trained models (PTMs) through fine-tuning for the purpose of enhancing computational efficiency and ensuring effectiveness, especially in the context of large models and data scarcity in downstream tasks [22]. However, the growing number of PTMs makes it impractical to fine-tune all of them and then choose the best one. Therefore, model recommendation from a large model zoo has gained increasing popularity [4, 14]. Consequently, a pivotal research question emerged: ***How can we accurately assess the performance of PTMs without directly fine-tuning them on the target dataset?***

Model recommendation could be regarded as a special application of transfer learning, where a crucial step is to identify the most suitable source models to maximally benefit the training of target task [12]. Most existing approaches make their source selection relying on the pairwise transferability estimation, which measures the expected transfer performance between source and target tasks [12]. Current transferability metrics typically approximate the optimal target performance based on source pretrained features [5, 24, 29], source predictor [17], or model gradient [21]. However, they still require evaluating source embedding or logits, hence do not scale to large model zoos and large volumes of target tasks. Recent works favor a learning-based approach to predict transfer performance efficiently based on metadata of models and datasets, leveraging historical fine-tuning results over a zoo of models and tasks [4, 14, 15, 20, 31]. Nevertheless, their effectiveness varies when dealing with a large number of heterogeneous models and target tasks, due to **1)** The intrinsic, high-order relationships among tasks, beyond mere pairwise connections, are often overlooked, while such relationships have been demonstrated in Taskonomy [30]. **2)** Model and dataset attributes used in previous works are coarse-grained, and can not capture the semantic label and domain similarity between tasks well.

To overcome those limitations, we formulate the transferability estimation problem as a bipartite graph link prediction problem, and propose a data-free transferability prediction method. Specifically, the graph represents tasks and transferabilities among them as nodes and edge weights respectively. A GNN is applied to predict the unknown transferabilities, i.e., the missing edges. On each node, we introduce the metadata to depict the model/task properties. By learning the low-dimensional embeddings of tasks and predicting edges in the latent space using the powerful graph representation capability of GNN, our method can effectively capture the intrinsic relationships among tasks so as to infer the transferability for unseen task pairs from the training history. Our contributions are three-fold:

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

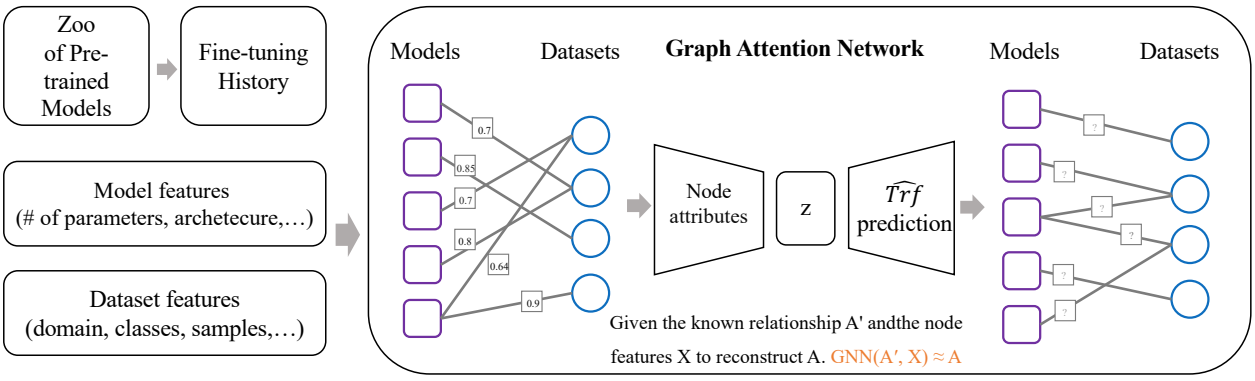

Figure 1: An illustration of model and dataset with different meta data in relationship graph.

- We formulate the model recommendation problem as a transferability prediction task by using bipartite graph link prediction, validating the effectiveness of incorporating model and dataset metadata.
- By leveraging graph structure learning instead of traditional regression methods, our approach captures high-order intrinsic relationships among tasks, thus improving the accuracy of transferability prediction.
- Experiments on cross-domain and cross-task scenarios, including the image classification and semantic segmentation datasets, show that our method outperforms state-of-the-art approaches with non-trivial improvements.

2 PROBLEM SETUP

Consider a set of pre-trained models $\mathcal{M} = \{M_i\}_{i=1}^m$ and a target task dataset $D_T = \{(x_T^i, y_T^i)\}_{i=1}^n$, where each model M_i is composed of a feature extraction layer θ_i and a decoder layer h_i. Given a fixed budget of model candidates for fine-tuning, model recommendation focuses on maximizing the transferability of selected pre-trained models on the target task. In particular, the empirical transferability is a standard way to measure the transfer learning effect as defined in the following [24].

Definition 2.1 (Empirical Transferability). By employing the widely used transfer learning strategy known as "retrain head", where the feature extraction layer θ_i is retained while the decoder readout function h_t is fine-tuned on the target task, the resulted model performance on the target task is adopted as a score of empirical transferability.

$$\mathrm{Trf}(M_i \to D_T) \triangleq \mathbb{E}\left[\mathrm{acc}\left(y_T, x_T; \theta_i, h_t\right)\right] \tag{1}$$

Straightforward as it is, the high time and resource consumption of deriving empirical transferability gives rise to the problem of transferability estimation.

Definition 2.2 (Transferability Estimation). This task is to measure the empirical transferability without directly fine-tuning on the target task through the function $\widehat{\mathrm{Trf}}(M_i \to D_T) = \tau(M_i, D_T, \theta_i)$, where $\tau(\cdot)$ is a kind of pre-defined estimation function.

In this paper, we formulate the transferability estimation problem as a bipartite graph link prediction problem, i.e., deriving the transferability between models and datasets using link prediction. Particularly, instead of a single target T, we have l target datasets $\{D_{T_j}\}_{j=1}^l$, and the transferability graph is written as $G = (V_M, V_D, A)$, where V_M represents model nodes, V_D represents target datasets and $A = \{\mathrm{Trf}(M_i \to D_{T_j})\}_{i,j=1}^{m,l}$ with $|A| = |V_M| \times |V_D|$ represents the transferability matrix among tasks. In the link prediction problem, we only have access to the node information (source model information from V_m and target dataset information from V_d) and the partially known relation matrix A' acquired from historical empirical transfer performance, and need to leverage this information to predict the missing edges. With a graph learning model GNN, we recover the full adjacency matrix A by

$$A = GNN(A', X), \tag{2}$$

where X denotes the feature of nodes (covering information from V_m and V_d), whose detailed design will be introduced later.

3 METHODOLOGY

3.1 Graph Construction and Learning

Graph Auto-Encoders (GAEs) [13] based on GNNs have been widely adopted for graph link prediction, displaying effectiveness on bipartite graphs by leveraging side information [6, 13, 19]. After establishing the measurement of task interrelationship as a graph link prediction problem in Section 2, we propose a customized GAE framework to learn from the known relations among pairs of tasks (nodes) and try to recover the link weights in the whole transferability graph.

To be specific, to build the *encoder* in our framework, we customize a Graph Attention Network (GAT) [26] with combining multi-level information of nodes, which can capture the significance of various neighboring nodes and assign them with distinct attention weights, leading to more precise and robust representations of the graph structure. For the *decoder*, since our latent graph embeddings already encompass both content and structure information, we opt for a simple linear decoder. The decoder transforms the graph embeddings of the source and target nodes to predict the links between nodes, ensuring both efficiency and flexibility. This framework can support end-to-end training manner by minimizing the *reconstruction error*. As to the model input, the task feature is defined as a concatenation of multiple metadata representations,

each of them capturing different aspects of models and datasets. The specific formulation of key components are listed below.

Encoder: The vanilla GAT is capable of capturing information of node features and their interactions with attention mechanism:

$$h_j^{(l+1)} = \sigma\left(\sum_{i\in\mathcal{N}(j)} \alpha_{ij}^{(l)} W^{(l)} h_i^{(l)}\right), \tag{3}$$

where σ denotes a non-linear activation function such as ReLU, $\alpha_{ij}^{(l)}$ are the learned attention coefficients at layer l, and $W^{(l)}$ is a layer-specific trainable weight matrix. Specifically, the attention coefficients are computed using a learnable parameter vector $\vec{a}$, node features $\vec{h}_i$ and $\vec{h}_j$ as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\vec{a}^T[W_i\vec{h}_i \parallel W_j\vec{h}_j))}{\sum_{k\in\mathcal{N}_(j)} \exp(\text{LeakyReLU}(\vec{a}^T[W_i\vec{h}_i \parallel W_k\vec{h}_k]))}. \tag{4}$$

However, as the models and datasets are various, simply aggregating neighbors may lead to feature confusion and loss expressive ability. To address this, we propose to utilize a $COMBINE(\cdot)$ function that effectively aggregates information from higher-order neighborhoods. This approach would concatenate and gather information from the node's ego-feature, immediate neighbors, and higher-order neighbors, enhancing the distinction of node representations with multiple level of transferability.

$$z_i^{(l)} = COMBINE\left(W^{(0)}h_i^{(0)}, \ldots, h_i^{(l-1)}, h_i^l\right). \tag{5}$$

Decoder: The linear decoder utilizes the source and target node embeddings to predict the links between nodes:

$$\hat{A}_{ij} = \text{sigmoid}(W[z_i \| z_j]). \tag{6}$$

Here $\hat{A}$ represents the reconstructed adjacency matrix of the graph, W is the learnable weight matrix of the decoder.

Reconstruction Loss: The reconstruction error is evaluated by measuring the difference between the original adjacency matrix A and the reconstructed matrix $\hat{A}$: $L = \sum_{i=1}^{n} \text{MSE}(A_{ij}, \hat{A}_{ij})$.

3.2 Characterizing Datasets and Models

Prior studies [14, 15] have demonstrated the value of representing such characteristics with task metadata and integrating them into predictive models. In our study, we group meta attributes into a model feature set and a dataset feature set to reflect the asymmetry of transferability prediction. We further expand the task metadata by additionally incorporating domain attributes and label attributes. This allows the model to capture the fine-grained task similarity in the cross-domain and cross-task transfer setting. We elaborate these metadata as features in Table 1 and briefly explain them.

Model Metadata as Node Features (m) Key attributes of models (source) that are indicative of their learning capabilities and architectural complexities, including quantities related to the model architecture and pre-training datasets.

Dataset Metadata as Node Features (d) Characteristics of datasets (target), including sample size and the number of categories as well as their semantic meaning (e.g. domain for classification).

Table 1: Metadata as features of models and datasets

Attribute Name	Description
Architecture (m_{arch})	The architectural design of a model
Input Size (m_{size})	Dimensionality of the model inputs
Model Capacity (m_{cap})	The number of parameters
Pre-trained Dataset (m_{pre})	The dataset for pre-training the model
Model Performance (m_{perf})	An empirical measure of model accuracy
Number of Classes (d_{class})	Indicates the categorical complexity.
Categories (d_{cat})	A high-level description of categories
Sample Size (d_{size})	The total number of samples
Domain/Modality (d_{dom})	Reflects the specific area or type of data.

3.3 Workflow

From a high-level point of view, our proposed approach involves the following steps during the algorithm implementation:

1. **Task Embedding:** Derive the model and dataset features V_M and V_D with concatenated metadata of dataset(d) and model(m).
2. **Graph Learning:** Apply the GAT encoder to the partially known adjacency matrix A' and node features to obtain the node latent embeddings z.
3. **Link Prediction:** Use the linear decoder to predict the missing links and construct the transferability estimation matrix $\hat{A}$.

In particular, we name our Graph-learning based Transferability prediction framework as *G4Trf* and and its variant as *SCG4Trf*, which means a GNN model w/ or w/o the Skip Combination are applied for the prediction of transferability. The overall framework is illustrated in Fig. 1 and will be elaborated in the following sections.

4 EXPERIMENTS

4.1 Experimental Setup

For evaluation, we conduct extensive experiments using various pre-trained models as *Source Models* and data from both fine-grained classification and segmentation tasks as *Target Datasets*, where the target datasets cover multiple domains within the DomainNet dataset [18] and the Cityscapes dataset [7]. Due to space limitation, we elaborate on the setup of Classification task on DomainNet, and the settings of Segmentation task follows [25].

Datasets To explore the effectiveness of our method in cross-domain and cross-task scenarios, we divide the DomainNet dataset into a series of sub-datasets denoted as $\mathcal{T} = (\mathcal{D}, \mathcal{N}, \mathcal{S}, \mathcal{L})$, with $\mathcal{D}$ = {Real, Sketch, Clipart, Infograph, Painting, Quickdraw} as the target data domain, $\mathcal{N} = \{2, 4, 8, 16, 32\}$ as the number of classes in the classification task, $\mathcal{S} = \{10, 30, 50\}$ as the sample size of each class, and $\mathcal{L} = \{A, B, C\}$ denoting a task division based on the overlapping of classification labels. We ensure that the classes in tasks A and C are mutually exclusive, while the classes in task B consist of half of task A and half of task C. Consequently, we perform evaluations on $|\mathcal{D}| \times |\mathcal{N}| \times |\mathcal{S}| \times |\mathcal{L}| = 270$ target datasets.

Models In the DomainNet task, we adhere to the model selection criteria outlined in the state-of-the-art (SOTA) study in [14]. We select 10 models close to the Pareto-optimal curves, embracing a diverse range of common architecture families, denoted by the

Table 2: Different missing edge ratios for each task are evaluated. The average Pearson correlation coefficient between the predicted and ground truth performance is reported, which measures the linear correlation between prediction results and the ground truth. Higher values indicate better performance. Each experiment is repeated 10 times, and the mean and standard deviation (mean/std) of the results are reported.

Type	Method	Meta feature	DomainNet Tasks						Seg. Tasks
			Avg	Cold Start	0% ~20%	20% ~40%	40% ~60%	<60% ~80%	Avg
Unsupervised	Random	-	0.27 ± 0.01	-0.01 ± 0.07	0.01 ± 0.04	0.10 ± 0.04	0.29 ± 0.03	0.56 ± 0.05	0.29 ± 0.01
	MC [8]	-	0.47 ± 0.02	0.39 ± 0.03	0.43 ± 0.03	0.32 ± 0.06	0.38 ± 0.03	0.56 ± 0.04	0.38 ± 0.02
Feature-based	LogME [29]	-	0.28 ± 0.02	0.01 ± 0.06	0.03 ± 0.05	0.13 ± 0.06	0.29 ± 0.04	0.55 ± 0.04	0.57 ± 0.03
	LEEP[17]	-	0.39 ± 0.02	0.34 ± 0.06	0.26 ± 0.05	0.24 ± 0.06	0.33 ± 0.04	0.56 ± 0.04	0.65 ± 0.03
Learning-based	LR [14]	d,m	0.65 ± 0.02	0.56 ± 0.06	0.54 ± 0.04	0.56 ± 0.04	0.65 ± 0.04	0.77 ± 0.03	0.74 ± 0.03
	G4Trf	d,m	0.74 ± 0.01	0.61 ± 0.05	0.63 ± 0.03	0.68 ± 0.04	0.78 ± 0.03	0.87 ± 0.02	0.81 ± 0.03
	SCG4Trf	d,m	**0.76 ± 0.01**	**0.61 ± 0.05**	**0.63 ± 0.03**	**0.70 ± 0.03**	**0.79 ± 0.02**	**0.88 ± 0.02**	**0.83 ± 0.02**

model set [9–11, 16, 23]

$$\mathcal{M} = \left\{ \begin{array}{l} \text{ConvNeXt Small, ConvNeXt Tiny, ResNet18, ResNet50,} \\ \text{ResNet101, Wide ResNet101, DenseNet121, MobileNet,} \\ \text{EfficientNet b0, EfficientNet b3} \end{array} \right\}.$$

Groundtruth The empirical transferability score is used to evaluate the model M's performance on task T after a fine-tuning process.

4.2 Numerical Results

To investigate the effectiveness of meta-features and graph learning methods, we compare different approaches: unsupervised methods, feature-based methods that do not consider the meta-features of models and datasets (e.g., LogME and LEEP), and a state-of-the-art learning-based method (LR) for model selection in the model zoo. The latter method utilizes meta-data and partially known training history to predict model performance.

Table 2 summarizes the numerical results. We can observe that using meta-data to predict model performance outperforms feature-based transferability score methods in both classification and segmentation tasks. This is particularly evident in DomainNet tasks, where tasks vary in different domains and the number of classes, posing a challenge for feature-based methods due to the domain gap and task difficulty.

Further, in the learning based method, the graph learning methods outperform LR approach by a significant margin, verifying our motivation to capture high-order information into transferability estimation. To be more concrete, by enhancing the G4Trf model with such a combining mechanism, SCG4Trf significantly outperforms the baseline LR by over 15% in DomainNet tasks and 12% in Segmentation tasks.

4.3 Further Analysis

To analyze the latent representation Z, we utilize t-SNE to obtain low-dimensional node representations. We selectively plot the source models as red nodes and the target datasets as blue nodes. As shown in Figure 2, similar model architectures tend to cluster closely together. Empirically, in DomainNet tasks, the performance of the quickdraw domain is quite different from others due to the large domain gap. Meanwhile, the sketch, clipart, and painting domains perform similarly. This is evident in nodes and the target datasets as blue nodes. It also can be found that tasks with similar performance cluster more closely in the low-dimensional task space. This is consistent with our assumption that good representations for predicting task relationships are closer in low-dimensional space when model fine-tuning results are similar. These good representations are not only useful for better predicting model performance for model selection but also potentially beneficial for guiding model aggregation or model merging in broader applications.

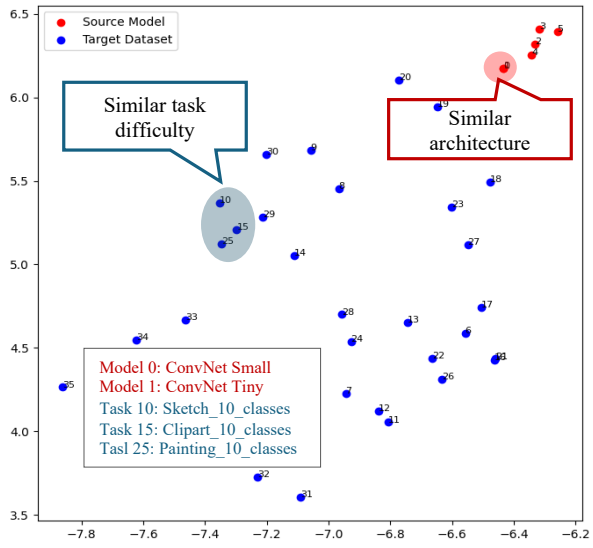

Figure 2: t-SNE visualization of the latent representation z.

5 CONCLUSION

In this work, we introduced a novel approach for transferability prediction by formulating it as a bipartite graph learning problem. Our method leverages the task metadata and combines graph attention mechanisms to learn more effective task embeddings for prediction. These innovations enable the effective characterizing of intrinsic task relationships and thus promoting the reliable prediction of transferability for unseen task pairs.

REFERENCES

[1] 2024. Hugging Face Models. Available at: https://huggingface.co/models.
[2] 2024. PyTorch Hub. Available at: https://github.com/pytorch/hub.
[3] 2024. TensorFlow Hub. Available at: https://github.com/tensorflow/hub.
[4] Jiameng Bai, Sai Wu, Jie Song, Junbo Zhao, and Gang Chen. 2024. Pre-Trained Model Recommendation for Downstream Fine-tuning. *arXiv preprint arXiv:2403.06382* (2024).
[5] Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. 2019. An information-theoretic approach to transferability in task transfer learning. In *2019 IEEE international conference on image processing (ICIP)*. IEEE, 2309–2313.
[6] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3213–3223.
[8] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2009. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. *arXiv preprint arXiv:0909.4061* 909 (2009).
[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[10] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.
[12] Junguang Jiang, Yang Shu, Jianmin Wang, and Mingsheng Long. 2022. Transferability in deep learning: A survey. *arXiv preprint arXiv:2201.05867* (2022).
[13] Thomas N Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *NIPS Workshop on Bayesian Deep Learning* (2016).
[14] Hao Li, Charless Fowlkes, Hao Yang, Onkar Dabeer, Zhuowen Tu, and Stefano Soatto. 2023. Guided recommendation for model fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3633–3642.
[15] Ziyu Li, Hilco van der Wilk, Danning Zhan, Megha Khosla, Alessandro Bozzon, and Rihan Hai. 2024. Model Selection with Model Zoo via Graph Learning. *arXiv preprint arXiv:2404.03988* (2024).
[16] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11976–11986.
[17] Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. 2020. Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*. PMLR, 7294–7305.
[18] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1406–1415.
[19] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*. Springer, 593–607.
[20] Wenqi Shao, Chonghe Jiang, Kaipeng Zhang, Yu Qiao, Ping Luo, et al. 2024. Foundation Model is Efficient Multimodal Multitask Model Selector. *Advances in Neural Information Processing Systems* 36 (2024).
[21] Jie Song, Yixin Chen, Xinchao Wang, Chengchao Shen, and Mingli Song. 2019. Deep model transferability from attribution maps. *Advances in Neural Information Processing Systems* 32 (2019).
[22] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. 2019. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 403–412.
[23] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.
[24] Yang Tan, Yang Li, and Shao-Lun Huang. 2021. Otce: A transferability metric for cross-domain cross-task representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 15779–15788.
[25] Yang Tan, Yicong Li, Yang Li, and Xiao-Ping Zhang. 2023. Efficient Prediction of Model Transferability in Semantic Segmentation Tasks. In *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 720–724.
[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[27] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3 (2016), 1–40.
[28] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *Advances in neural information processing systems* 27 (2014).
[29] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*. PMLR, 12133–12143.
[30] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3712–3722.
[31] Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. 2024. Model spider: Learning to rank pre-trained models efficiently. *Advances in Neural Information Processing Systems* 36 (2024).
[32] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.