Amalgamating Knowledge from Heterogeneous Graph Neural Networks

Yongcheng Jing[1], Yiding Yang[2], Xinchao Wang[3,2], Mingli Song[4], Dacheng Tao[1]
[1]The University of Sydney, [2]Stevens Institute of Technology,
[3]National University of Singapore, [4]Zhejiang University
{yjin9495, dacheng.tao}@sydney.edu.au, yyang99@stevens.edu,
xinchao@nus.edu.sg, brooksong@zju.edu.cn

Abstract

In this paper, we study a novel knowledge transfer task in the domain of graph neural networks (GNNs). We strive to train a multi-talented student GNN, without accessing human annotations, that "amalgamates" knowledge from a couple of teacher GNNs with heterogeneous architectures and handling distinct tasks. The student derived in this way is expected to integrate the expertise from both teachers while maintaining a compact architecture. To this end, we propose an innovative approach to train a slimmable GNN that enables learning from teachers with varying feature dimensions. Meanwhile, to explicitly align topological semantics between the student and teachers, we introduce a topological attribution map (TAM) to highlight the structural saliency in a graph, based on which the student imitates the teachers' ways of aggregating information from neighbors. Experiments on seven datasets across various tasks, including multi-label classification and joint segmentation-classification, demonstrate that the learned student, with a lightweight architecture, achieves gratifying results on par with and sometimes even superior to those of the teachers in their specializations. Our code is publicly available at https://github.com/ycjing/AmalgamateGNN.PyTorch.

1. Introduction

An increasing number of pre-trained deep neural networks (DNNs) have been generously released online for the sake of handy reproducibility [49]. As such, reusing these pre-trained models to alleviate training effort or to enhance performance, has emerged as a trending research topic in recent years. The seminal work of Hinton *et al.* [12], for instance, first raises *Knowledge Distillation*, where a pre-trained teacher model is utilized to generate soft labels so as to learn a lightweight student model with competent performance. Following this student-teacher paradigm, many other distillation-based approaches have been applied to various domains and have demonstrated promising results [7, 30, 47, 48, 55].

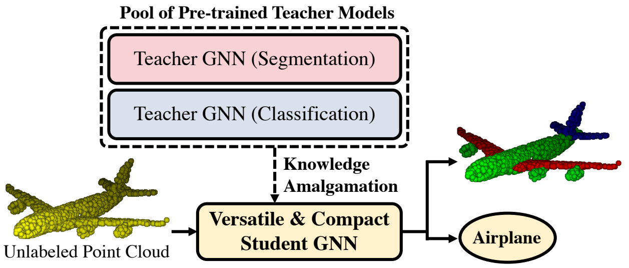

Pool of Pre-trained Teacher Models
Teacher GNN (Segmentation)
Teacher GNN (Classification)
Knowledge Amalgamation
Unlabeled Point Cloud
Versatile & Compact Student GNN
Airplane

Figure 1. Illustrations of amalgamating knowledge from heterogeneous teacher GNN models. "Teacher GNN (Segmentation)" and "Teacher GNN (Classification)" are pre-trained point cloud part segmentation and classification models, respectively. Knowledge amalgamation aims to learn a multi-talented and lightweight student GNN from teacher GNNs without human annotations.

Almost all existing approaches on knowledge transfer from pre-trained models have been focused on convolutional neural networks (CNNs), which take data in regular domains, like images, as input. Nevertheless, many other data samples take irregular forms and thereby resort to graph representations, calling for graph neural networks (GNNs). The work of [43], as the first attempt, generalizes knowledge distillation to GNNs, and introduces a customized approach tailored for irregular data. In spite of the improved performance, this approach is limited to the scenario where the student learns from a single teacher, and meanwhile holds a homogeneous architecture and tackles the same task as the teacher does.

In this paper, we strive to make one step further towards knowledge transfer from pre-trained GNNs, by studying a novel *knowledge amalgamation* task. Our goal is to train a multi-talented student GNN, from a couple of pre-trained teacher GNNs with heterogeneous architectures and specializes in different tasks, for example one working on point cloud segmentation and the other on classification, as shown in Fig. 1. We further assume that, in the knowledge amalgamation process, no human annotations are available. The student learned in this way is anticipated to integrate both teachers' expertise yet comes with a compact size, making it competent for resource-constrained applications such as edge computing.

Nevertheless, such an ambitious goal is accompanied with challenges. The first challenge regards handling graph features with varying dimensions. Unlike CNNs that take as input grid-structured data with *fixed channel numbers*, such as RGB images, in our scenario, GNNs pre-trained on different datasets work with distinct feature dimensions. For example, nodes in the citation network dataset *Cora* have 1433 features, while those in *Citeseer* have 3703 features. The student GNN would therefore have to accommodate the diverse feature dimensions. The second challenge lies in encoding topological semantics of graphs. As GNNs are designed to explicitly account for the topological information concealed in the graph data, aligning the topological semantics between teachers and the student emerges as a critical issue to be addressed in GNN knowledge amalgamation.

Towards this end, we propose a slimmable graph convolutional operation that enables adaptive activation or deactivation of layer channels; graph data of different input channels can therefore be simultaneously accounted for under one student model. Furthermore, we introduce *topological attribution map* (TAM), a general graph representation scheme to highlight structural saliency in terms of information propagation from neighbors. The derived student model is enforced to produce a TAM that resembles those from the teachers, in which way the student imitates the teachers' fashions of aggregating features to the center node. Notably, TAM is free of data labels and readily applied to heterogeneous GNN architectures.

Our contribution is therefore a novel GNN-based knowledge amalgamation approach to train a versatile student model that covers the specialties from heterogeneous-task teachers, without human annotations. This is typically accomplished through a slimmable graph convolutional operation to accommodate varying-dimension features from teachers, together with a TAM scheme for learning the teachers' topological semantics. We evaluate the proposed method on four different tasks across various domains, including single- and multi-label node classifications, 3D object recognition, and part segmentation. Experimental results demonstrate that, the learned student GNN model is competent to handle all different tasks of the heterogeneous teachers, sometimes with a performance even superior to those of the teachers, and meanwhile comes at a significant reduction in computational cost.

2. Related Work

Graph Neural Network. Graph neural networks (GNNs) have achieved unprecedented advances in recent years, showing promising performance in handling graph data lying in the non-Euclidean domain [16, 37, 53, 44, 6, 41, 19, 23, 36, 13, 18, 25, 42]. Since the seminal work of Kipf *et al.* [16], a large number of approaches have been proposed for an enhanced GNN model. Specifically, GraphSAGE, proposed by Hamilton *et al.* [9], provides a general inductive framework towards scalable GNN for huge graphs. Graph attention network (GAT) [35], on the other hand, focuses on introducing a novel attention mechanism for GNN, allowing for efficient graph processing without knowing the graph structure upfront. Furthermore, [52] and [29] propose a novel PairNorm layer and a DropEdge strategy to alleviate the oversmoothing problem in GNNs. Despite the encouraging progress in the field of GNN, there is a lack of research on reusing pre-trained GNN models.

Multi-task Learning. The proposed task of graph knowledge amalgamation is also related to multi-task learning. Multi-task learning aims to leverage task relatedness to jointly learn a group of tasks with shared architectures [1, 4, 5, 8, 17, 50]. In the past few years, multi-task learning has been widely studied in various areas, such as bioinformatics [10, 20], ubiquitous computing [39, 40], natural language processing [4, 38] and computer vision [21, 54, 15, 14]. Specifically, He *et al.* [11] develop a multi-task framework that combines object detection and segmentation. Also, in [51], Zhang *et al.* devise a convolutional neural network architecture for joint face detection, pose estimation, and landmark localization. Another work in [3] constructs a multi-task recurrent neural network, of which the output layer has multiple units to simultaneously estimate the relative distance, interactions, and standing orientations. A more recent work [21] further proposes a multi-task collaborative network that achieves joint learning of referring expression comprehension and segmentation.

Model Reusing. Reusing pre-trained models has become increasingly prevalent in recent years. The seminal work of Hinton *et al.* [12] proposes the concept of knowledge distillation, where the soft labels obtained from a cumbersome teacher model are used for training a compact student model. Following this pioneering teacher-student framework, plenty of algorithms are proposed to fully utilize the knowledge concealed in the pre-trained teachers [30, 28, 48, 7, 55, 26, 33]. In particular, Rusu *et al.* [30] propose a novel progressive neural network to learn useful features from multiple teachers. The work in [26], on the other hand, proposes an Actor-Mimic scheme to reuse several teacher models specializing in diversified tasks. Also, the works in [32, 34, 22, 45] propose to reuse multiple trained teacher CNNs, working on different tasks, to learn a versatile student model, but built upon a strong assumption that the teacher models share the same CNN architecture. Given the promising advances of model reusing techniques, however, none of these existing algorithms investigates a solution for reusing heterogeneous GNN models.

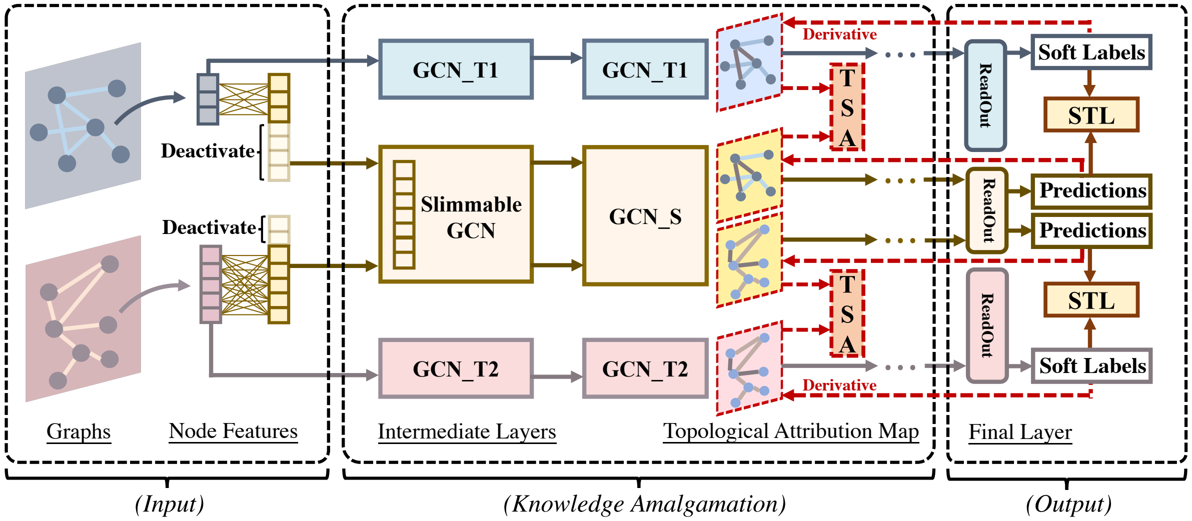


Figure 2. The overall framework of the proposed knowledge amalgamation method tailored for GNNs. For illustration, we take two pre-trained teacher GCNs as an example. On the input side, the dimensions of input node features would vary with different graph samples. **GCN_T1**, **GCN_S** and **GCN_T2** represent the graph convolutional layers from pre-trained teacher #1, lightweight student, and pre-trained teacher #2, respectively. **TSA** and **STL** denote the proposed topological semantics alignment module and the soft target learning module, respectively. The topological attribution map is obtained by computing the edge gradients of the constructed unary edge features, as explained in Sect. 4.3.

3. Problem Definition

The problem we aim to address here is to learn a versatile and lightweight student GNN model, with only unlabeled graph data, that amalgamates topology-aware knowledge from multiple task-wise heterogeneous teachers. Specifically, assume that we are given N pre-trained GNN models $G = \{g_1, g_2, \cdots, g_N\}$, each of which specializes in different tasks, such as paper classifications on specific topics [31] or predictions of various protein functions [56]. We use $T(g_i)$ to represent the specific task handled by teacher g_i. The goal of knowledge amalgamation is then formulated as learning a student GNN model g_s that has the following three properties:

- The student g_s covers the expertise of all heterogeneous teachers.
- The model size of the student is smaller than the sum of teachers, preferably even smaller than a single teacher.
- Learning of g_s requires only raw graph data without human-labeled annotations.

The target student GNN model is therefore expected to be capable of simultaneously handling heterogeneous tasks, and meanwhile more portable for deployment on the mobile-terminal side.

Also, for different pre-trained teachers g_i, we impose *no constraints* on g_i's architectures being the same, meaning that g_i can have diversified layer numbers, different feature dimensions, or even distinct layer mechanisms, such as graph convolutional layers by Kipf *et al.* [16] and graph attention layers by Veličković *et al.* [35].

4. Proposed Method

Towards addressing the proposed problem of knowledge amalgamation, we introduce the proposed dedicated approach tailored for GNN models. In what follows, we start by giving an overview of the proposed method, and then detail the key modules. Finally, we propose a dedicated training strategy that trains the student GNN intertwined with teacher GNNs.

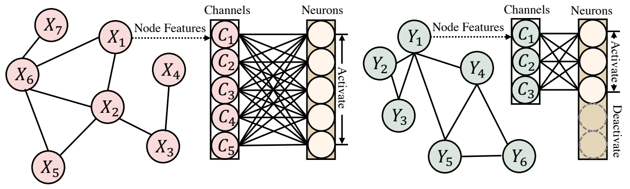
4.1. Overview

The overall workflow of the proposed method is shown in Fig. 2. The task of knowledge amalgamation imposes three major challenges, respectively on input data, intermediate features, and output labels. The challenge on the input side concerns handling multiple teacher GNNs with different feature dimensions. This dilemma is solved by equipping the student with the proposed slimmable graph convolutions (Fig. 2 *(Input)*).

The second challenge lies in the effective extraction and transfer of topological information from teachers. In our proposed approach, this issue is tackled by the proposed topological semantics alignment module (Fig. 2 *(TSA)*).

The last challenge relates to the lack of human-labeled annotations: how to obtain supervision information from unlabeled graph data. We address this issue by explicitly imitating the soft predictions of heterogeneous teachers (Fig. 2 *(STL)*), as is also done in CNN-based model reusing technique [12].

Therefore, in what follows, we put our emphasis upon



(a) Input Graph #1 (b) Input Graph #2

Figure 3. Illustrations of the proposed slimmable graph convolutional operation, where X and Y denote graph nodes. The neurons in multi-layer perceptrons (MLPs) of GNN are adaptively activated or deactivated based on the feature dimensions of the input graph data.

the slimmable graph convolutional modules and the topological semantics alignment module, both of which are specific to the task of GNN model reusing.

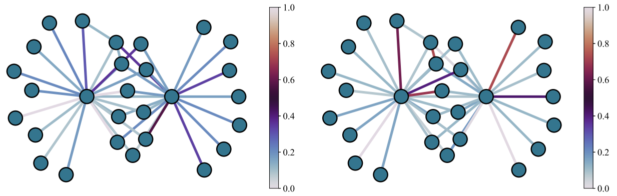
4.2. Slimmable Graph Convolution

On the input side, unlike CNNs that always receive grid-like RGB images with constant channel numbers, GNN models, depending on the handled tasks, vary in the feature dimensions of input nodes. Taking the three popular paper-citation datasets, *Cora*, *Citeseer* and *Pubmed* as examples [31], all of these three datasets contain publications as graph nodes. Nevertheless, they contain distinct channel numbers for each node: 1433 for *Cora*, 3703 for *Citeseer*, and 500 for *Pubmed*. This challenge of diversified feature dimensions makes it infeasible to simply use a naive GNN architecture for the target multi-talented student model.

To solve this dilemma, we devise a dedicated slimmable graph convolutional layer, where the layer channels can be adaptively activated or deactivated depending on different input feature dimensions, as shown in Fig 3. To further illustrate the proposed slimmable graph convolutional layer, we take the task of node classification as an example.

Assume that we have separate input graph nodes X_i and Y_j from different graphs with C_i and C_j feature dimensions ($C_i \neq C_j$) to concurrently account for. Firstly, before training, we set a maximum channel number C_{max} for the proposed slimmable graph convolutions, so as to define the shape of weights in GNN layers. Then, given input nodes X_i with the node feature dimension of C_i, the slimmable graph convolution adaptively deactivates the $|C_{max} - C_i|$ neurons and uses only the C_i-channel filter to deal with X_i. For the processing of the node Y_j with C_j feature channels, a similar scheme is also applied, where the slimmable graph convolution dynamically switches to C_j-channel filter to manage the corresponding input node of Y_j.

In knowledge amalgamation, by replacing the first layer with slimmable graph convolutional layer, the student GNN can simultaneously handle graph samples with varying input feature dimensions; while also equipped with slimmable graph convolutions in the intermediate layers, the student GNN model can also trade off between accuracy and latency at runtime, by switching between models with different numbers of active layer channels, thus making it possible to adapt the learned student model across different devices with limited response time budgets.



(a) TAM of Teacher GNN #1 (b) TAM of Teacher GNN #2

Figure 4. Visualizations of the scaled topological attribution map (TAM) of two teacher GNNs given the same input graph data. As an example, two teachers here are pre-trained multi-label node classification models that handle a different set of classes. Colors encode the importance of each connection for the corresponding task of each teacher.

4.3. Topological Semantics Alignment

Unlike conventional convolutional layers that only receive grid-structured data as input and generate high-level semantic representations, graph convolutional layers are designed to process the graph data, either in the form of grid or non-grid structures. To this end, the intrinsical mechanism of graph convolutions is to generate representations for each node by collectively aggregating its own features and its neighboring nodes' features. As a result, the generated feature maps from graph convolutional layers contain both the topological properties of the input graph and also the high-level node content information. Simply applying prior CNN-based model reusing techniques, regardless of topological connections among different nodes, for GNN-based knowledge amalgamation, will inevitably lead to lossy knowledge transfer [37].

Towards addressing this challenge, the key issue to be considered is: how to derive a structure-aware graphical representation, tailored for aligning the concealed topological information between teachers and the student. One possible solution to this issue could be using the pairwise feature distance between every two connected nodes as the potential structure-aware representation to perform alignment between the student and teacher, as is done in [43]. This solution might be feasible for topology-aware knowledge transfer from a *single teacher*.

However, this possible graphical representation does not fit our case of amalgamating multiple streams of knowledge from heterogeneous teachers. Take the amalgamation of multi-label node classification models as an example, where each teacher handles a separate set of classes. The goal of the student GNN is to concurrently deal with all the classes covered by the teachers. In this case, given the same graph as input, different teachers would have distinct aforementioned possible representations, whereas the student would derive only one single representation. As a result, simultaneously aligning these multiple distinct representations of teachers with a single student GNN will make the learning of different knowledge compete with each other, which will be validated in the experiments. This competitive situation is contradictory to our goal, where we expect that the learning of different teachers' knowledge could potentially benefit and cooperate with each other for improved performance.

Motivated by this observation, we propose a novel topological representation, termed as *topological attribution map* (TAM), for the structural semantics alignment in knowledge amalgamation from heterogeneous teacher GNNs. Specifically, the proposed TAM is derived by computing the gradients of the given GNN's output class scores with respect to the adjacency matrix, as shown in Fig. 4. As a result, the obtained TAM contains the structural saliency in propagating information from neighbor nodes, indicating the importance of each individual connection on the final GNN predictions. Compared with the aforementioned possible representation, the design of the proposed TAM offers two benefits in knowledge amalgamation:

- The proposed TAM can be readily applied to heterogeneous GNN architectures, including the models with distinct aggregating mechanisms like graph convolutional network (GCN) [16] and graph attention network (GAT) [35], and also those with different layer numbers and channels.
- The proposed TAM can be extracted in a teacher-aware manner, meaning that a student GNN can derive multiple TAMs, which correspond to different teachers that handle separate classes. Specifically, this is achieved by using the specific subset of class scores, corresponding to the task of each teacher, to compute the teacher-specific TAMs. This manner alleviates the aforementioned competitive dilemma in amalgamating multiple teacher GNNs.

The workflow of computing the proposed TAM is given as follows. Consider a graph represented by a tuple $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of unordered vertices and $\mathcal{E}$ represents the set of edges connecting different vertices $v \in \mathcal{V}$. Let $\mathcal{A} \in \mathbb{R}^{n \times n}$ denote the adjacency matrix, where n is the number of graph nodes. Given an input graph $\mathcal{G}_0$ and a GNN model g, the proposed TAM representation $\mathcal{F}$ can be generally computed as:

$$\mathcal{F} = \left.\frac{\partial \mathcal{P}}{\partial \mathcal{A}}\right|_{\mathcal{G}_0} \in \mathbb{R}^{n \times n}, \quad \mathcal{P} = g(\mathcal{G}_0), \tag{1}$$

where $\mathcal{P}$ is the predicted class scores with the input $\mathcal{G}_0$.

Based on Eq. 1, given a set of pre-trained teacher GNNs $\{\mathcal{T}\}$, we propose a topological semantics alignment loss for knowledge amalgamation:

$$\mathcal{L}^{\mathcal{T}_i}_{topology} = \| \frac{\partial \mathcal{P}^{\mathcal{S}}_{d_{\mathcal{S}} \cap d_{\mathcal{T}_i}}}{\partial \mathcal{A}} - \frac{\partial \mathcal{P}^{\mathcal{T}_i}_{d_{\mathcal{T}_i}}}{\partial \mathcal{A}} \|, \tag{2}$$

where $d_{\mathcal{S}}$ and $d_{\mathcal{T}_i}$ represent the set of classes handled by the student $\mathcal{S}$ and the i-th teacher $\mathcal{T}_i$, respectively. $\mathcal{P}^{\mathcal{S}}_{d_{\mathcal{S}} \cap d_{\mathcal{T}_i}}$ represents a subset of the student's predicted class scores corresponding to those of the teacher $\mathcal{T}_i$, thus leading to a teacher-aware topological representation for knowledge amalgamation. The total topological alignment loss can then be computed as the sum of Eq. 2 over multiple teachers: $\mathcal{L}_{topology} = \sum_i \mathcal{L}^{\mathcal{T}_i}_{topology}$.

Algorithm 1 GNN-based Knowledge Amalgamation from Heterogeneous Teachers

```
Input: $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^{\mathcal{M}}$: $\mathcal{M}$ trained teacher GNNs; $\mathcal{G} = \{\mathcal{G}_k\}_{k=1}^{\mathcal{K}}$: unlabeled graph samples.
Output: $\mathcal{S}$: Target versatile and lightweight student GNN.
Set $C_{max}$ as the maximum feature dimension in $\mathcal{G}$;
Initilize student model $\mathcal{S}$;
for $m = 1$ to $\mathcal{M}$ do
    // Obtain topological representation and soft labels from Teacher $\mathcal{T}_m$
    Feed $\mathcal{G}$ with matched input dimensions into $\mathcal{T}_m$;
    Compute topological representation $\mathcal{F}^{\mathcal{T}_m}$ by Eq. 1;
    Compute the soft labels $\mathcal{P}^{\mathcal{T}_m}$ from the output layer of teacher $\mathcal{T}_m$;
    // Obtain topological representation and output predictions from Student $\mathcal{S}$
    Feed the same $\mathcal{G}$ into $\mathcal{S}$ and process $\mathcal{G}$ with slimmable graph convolutions in $\mathcal{S}$;
    Compute topological representation $\mathcal{F}^{\mathcal{S}}$ by Eq. 1;
    Compute soft labels $\mathcal{P}^{\mathcal{S}}$ from the output layer of $\mathcal{S}$;
    // Compute two losses
    Compute $\mathcal{L}^{\mathcal{T}_m}_{topology}$ from $\mathcal{F}^{\mathcal{T}_m}$ and $\mathcal{F}^{\mathcal{S}}$ by Eq. 2;
    Compute $\mathcal{L}^{\mathcal{T}_m}_{soft}$ from $\mathcal{P}^{\mathcal{T}_m}$ and $\mathcal{P}^{\mathcal{S}}$;
end for
Compute total loss over $\{\mathcal{T}_i\}_{i=1}^{\mathcal{M}}$ by Eq. 3;
Optimize $\mathcal{S}$ with Adam for epochs.
```

For implementations, there are two specific issues to be considered when using the naive computation method in Eq. 1 to obtain TAM. Firstly, there is a lack of a unified approach for computing the derivative of network outputs with respect to the adjacency matrix for heterogeneous GNN architectures like GCN and GAT. Different types of GNNs have different ways to incorporate the adjacency matrix in information aggregations, leading to inconsistent ways to obtain TAM.

Thus, we devise here a unified implementation method to compute TAM across various GNN architectures. Our idea is to first construct unary edges within the network based on the adjacency matrix, where the corresponding edge features are all equal to 1. The constructed unary edges

Table 1. Results of amalgamating knowledge from multi-label node classifications GAT models, in terms of micro-averaged F_1 score. The obtained student achieves competitive performance compared with the teachers, yet with a moderately compact size.

Methods	**Model Size**	**PPI_Set1**	**PPI_Set2**
Teacher 1	11.61M	98.73	N/A
Teacher 2	11.56M	N/A	98.62
Student_{MTL+AT} [48]	14.57M	97.03	96.99
Student_{MTL+LSP} [43]	14.57M	97.27	97.22
Student_Ours (w/o TSA)	14.57M	97.95	97.98
Student_Ours (w/ TSA)	**14.57M**	**98.44**	**98.42**

are then involved in the graph computations by multiplying with the node features in aggregating features from neighbors. In this way, the proposed TAM in Eq. 1 can be equally obtained by directly computing the edge gradients of the constructed unary edges, of which the computation flow is shown as the red arrows in Fig. 2.

The other issue in implementations is related to the scale of the computed unary edge gradients. We experimentally observe that for some teacher GNNs, the obtained gradients could be large in magnitude, leading to a relatively large topological semantics alignment loss that would dominate other loss terms at the initial stage of training. As a result, the convergence speed of the student GNN would be slowed down. To address this issue, we propose to perform topological-aware edge gradient normalization before computing the topological semantics alignment loss. Specifically, we firstly compute the mean $\mu_i(\{\mathcal{F}\})$ and the standard deviation $\sigma_i(\{\mathcal{F}\})$ of the unary edge gradients around each center node v_i. The normalized unary edge gradients around v_i can then be obtained by computing $\frac{\{\mathcal{F}\}-\mu_i}{\sigma_i+\epsilon}$, where ϵ is a constant that avoids zero denominator.

4.4. Loss Function and Training Strategy

The total loss function for amalgamating knowledge from heterogeneous teachers can be formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{soft} + \lambda \mathcal{L}_{topology}, \tag{3}$$

where $\mathcal{L}_{soft}$ is the soft target loss computed as the mean squared error among the soft predictions from the student and the heterogeneous teachers, which is shown as the soft target learning (STL) module in Fig. 2. The definition of $\mathcal{L}_{topology}$ can be found in Sect. 4.3.

We also propose a training strategy, tailored for the proposed approach. As a whole, the detailed process of training a student GNN model from multiple heterogeneous teacher GNNs is concluded in Alg. 1. For each iteration, we accumulate the loss from all heterogeneous teachers and jointly optimize the student model, so as to make sure that the student simultaneously learns from all the teachers.

Table 2. Results of amalgamating teachers with heterogeneous GNN architectures, in terms of micro-averaged F_1 score.

Type	Teacher 1 (GAT)	Teacher 2 (GAT)	Student (GAT)
Task	{PPI_1}	{PPI_2}	{PPI_1, PPI_2}
F_1 Score	98.73	98.62	**98.44 / 98.42**
Type	Teacher 1 (GCN)	Teacher 2 (GAT)	Student (GAT)
Task	{PPI_1}	{PPI_2}	{PPI_1, PPI_2}
F_1 Score	69.48	98.62	**70.01 / 98.01**
Type	Teacher 1 (GAT)	Teacher 2 (GCN)	Student (GAT)
Task	{PPI_1}	{PPI_2}	{PPI_1, PPI_2}
F_1 Score	98.73	63.62	**98.05 / 62.96**
Type	Teacher 1 (GCN)	Teacher 2 (GCN)	Student (GAT)
Task	{PPI_1}	{PPI_2}	{PPI_1, PPI_2}
F_1 Score	69.48	63.62	**69.64 / 62.51**

5. Experiments

To evaluate the performance of the proposed approach, we conduct experiments on seven publicly available benchmarks across various tasks, including node classifications, point cloud classifications and part segmentation. Here, we clarify that in the experiments, our goal is *not* to achieve the state-of-the-art performance on each benchmark, but rather transferring as much as knowledge from heterogeneous teachers.

5.1. Experimental Settings

Datasets and Implementation Details. We evaluate the proposed knowledge amalgamate method on seven datasets across various tasks. Specifically, for multi-label node classification, we use protein-protein interaction (PPI) dataset [56], containing biological graphs with nodes labeled with various protein functions. Each node can concurrently have several labels. We further divide PPI into two subsets, termed as PPI_Set1 and PPI_Set2 with 60 and 61 biological labels, respectively, which are used to train two corresponding teachers. The student GNN aims to amalgamate the knowledge from the two teachers, capable of predicting all 121 labels.

For the amalgamation of single-label node classification models, we adopt Amazon Computers (10 classes)and Amazon Photo (8 classes) datasets [24], where the nodes represent various goods, labeled by the corresponding product categories. We randomly split the dataset with a ratio of 2:2:6 for training, validation and testing, respectively. We also use three citation network datasets for single-label node classification, *i.e*., Cora (7 classes), Citeseer (6 classes) and Pubmed (3 classes) [31]. The papers involved in these three datasets are all scientific publications, but with different subjects. We adjust the training/validation/testing split for the training of teachers in the supervised scenario, as is also done in [2].

For knowledge amalgamation from point cloud classifi-

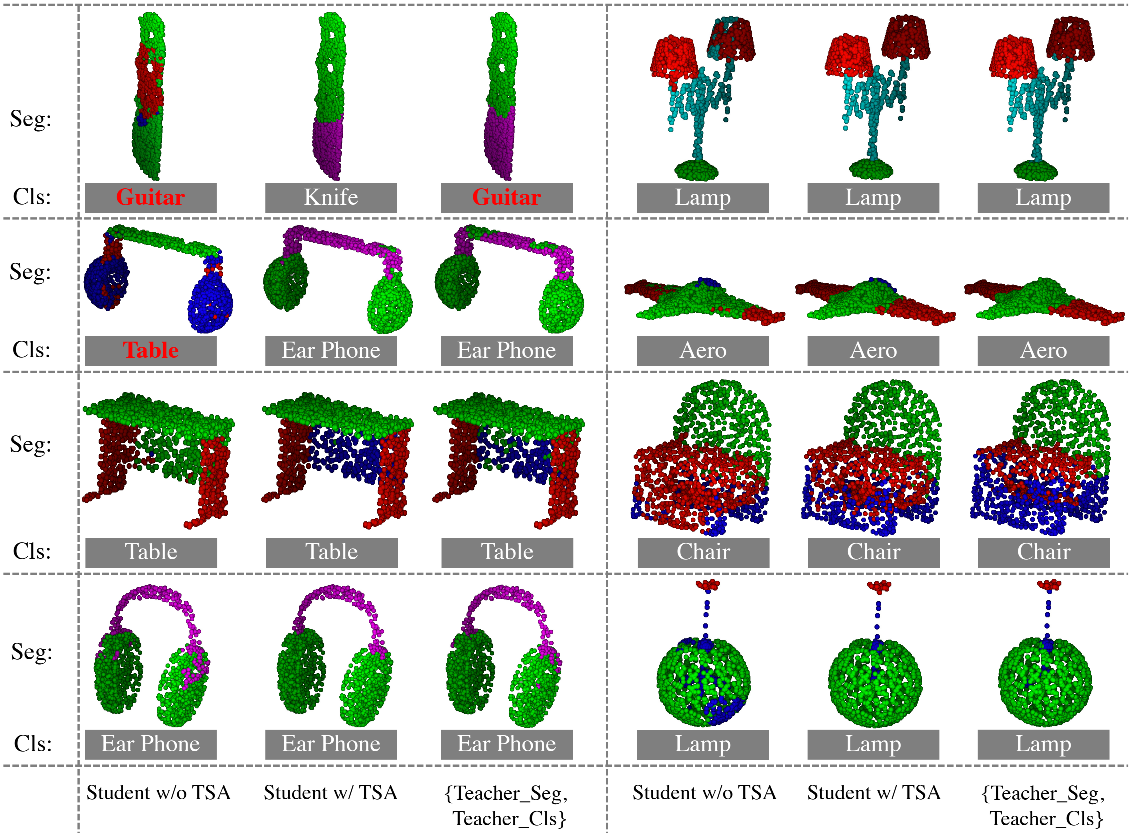


Figure 5. Visualization results of joint part segmentation (Seg) and classification (Cls). From left to right: the results of the learned student GNN without the proposed topological semantics alignment (TSA) module, those of the student with TSA, and the results of the two teacher GNNs. We use red texts to highlight the misclassified outputs. For some cases, our student GNN even achieves results superior to those of the teachers, as shown in the classification result of *Knife* and the segmentation results of *Ear Phone*.

cation and part segmentation models, we use the ShapeNet part dataset [46], containing $16,881$ shapes from 16 categories, annotated with 50 parts in total. The labeled categories and annotated parts are used to pre-train the teacher classification model and segmentation model.

For the unlabeled data sampling for the student GNN, we clarify that for a fair comparison with the pre-trained teacher GNNs, the training of the student in our experiments still uses the same training samples as those of the teachers, but without accessing ground truth labels, as explained in Sect. 3. Sampling more unlabeled graph samples from external datasets for training could further improve the performance of the learned student GNN.

We use heterogeneous architectures for the teachers and students in the task of node classifications, such as GCN [16] and GAT [35]. In particular, all the student GNNs are built with the proposed slimmable graph convolutional layer, so as to support graph inputs of varying feature dimensions. For the task of point cloud classification and part segmentation, we adopt the architecture of PointNet++ [27] for both the teachers and the student. The hyperparameter ϵ is set to 10^{-5}.

Comparison Methods. Since there are few existing knowledge amalgamation methods tailored for GNNs in the literature, we derive two possible solutions based on [48, 43] and the multi-task learning (MTL) scheme for comparisons. Specifically, upon the idea of attention transfer method [48] and MLT scheme, we devise a "Student_{MTL+AT}" method that amalgamates knowledge by matching the attention maps with heterogeneous teacher GNNs. Furthermore, we take the local structure preserving (LSP) module from [43] and develop a "Student_{MTL+LSP}" knowledge amalgamation approach by replacing our topological semantics alignment module with LSP. Specifically, "Student_{MTL+LSP}" uses the pairwise feature distance between every two connected nodes as the structure-aware representation to perform topological alignment, as mentioned as the possible solution in Sect. 4.3.

6. Results

Amalgamating Node Classification Models. Tab. 1 shows the results of amalgamating two pre-trained multi-label node classification model. In particular, to validate the ef-

Table 4. Results of amalgamating knowledge from point cloud classification and part segmentation models. The learned student GNN is even more compact than each of the teacher GNNs, yet competent to simultaneously handle all the tasks of teachers.

Method	**Model Size**	**mAcc** (Cls)	**mIoU** (Seg)	Aero	Bag	Cap	Car	Chair	Ear Phone	Guitar	Knife	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate Board	Table
# shapes	–	–	–	2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Teacher_Cls	17.69M	97.83	N/A	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Teacher_PartSeg	17.01M	N/A	81.72	82.34	81.92	86.12	78.33	90.54	72.18	91.25	86.09	83.57	95.48	70.63	94.98	81.98	55.99	73.73	82.34
Student_{MTL+AT}	6.37M	97.06	77.58	80.40	72.50	81.84	75.58	89.66	64.24	89.92	85.02	82.29	95.39	55.94	93.20	78.20	44.13	70.48	82.52
Student_{MTL+LSP}	6.37M	97.30	77.79	81.04	74.07	79.21	75.97	89.32	59.89	90.15	86.73	82.61	95.40	55.97	93.29	78.80	47.04	72.67	82.52
Student_Ours (w/o TSA)	6.37M	97.23	77.76	80.62	73.08	83.41	76.07	89.54	60.37	90.37	85.19	81.74	95.17	55.32	91.82	79.50	46.94	72.44	82.57
Student_Ours (w/ TSA)	**6.37M**	**97.67**	**78.96**	**81.82**	**76.07**	**81.17**	**76.91**	**89.59**	**70.56**	**90.17**	**85.69**	**82.95**	**94.92**	**57.06**	**94.02**	**79.24**	**48.05**	**72.67**	**82.50**

Table 3. Results of amalgamating single-label node classification models, in terms of average classification accuracies (%).

	Teacher 1	Teacher 2	Teacher 3	Teacher 4	Teacher 5
Type	GCN	GCN	GAT	GAT	GAT
Task	{Computers}	{Photo}	{Cora}	{Citeseer}	{Pubmed}
Model Size	25.84K	25.06K	739.6K	1.901M	259.8K
Accuracy	89.36	92.48	87.90	79.00	85.70
	Student 1		Student 2		
Type	GCN		GAT		
Task	{Computers, Photo}		{Cora, Citeseer, Pubmed}		
Model Size	20.29K		1.450M		
Accuracy	88.81 / 91.79		87.10 / 77.30 / 83.20		

fectiveness of the proposed TSA module, we conduct the ablation study by only using soft target learning for amalgamation, *i.e.*, setting $\lambda = 0$ in Eq. 3, which is termed as the method of "Student_Ours (w/o TSA)" in the table.

The student model learned with the proposed method, as shown in Tab. 1 (Student_Ours (w/ TSA)), achieves gratifying performance on par with that of the two teacher models, and meanwhile maintains a compact model size. Also, the results in the last two lines of Tab. 1 validate the effectiveness of the proposed TAM-based topological semantics alignment module, where Student_Ours (w/ TSA) outperforms Student_Ours (w/o TSA) by about 0.5 in F_1 score. The proposed knowledge amalgamation method also achieves favorable performance compared with the two derived comparison methods.

In Tab. 2, we also show the corresponding multi-label classification results by amalgamating knowledge from various types of GNN models. The notation "{PPI_1}" means that the teacher can only handle the task of PPI_Set1, while "{PPI_1, PPI_2}" indicates the capability of simultaneously handling the two tasks. Despite the heterogeneous types of trained teachers, the obtained student model still achieves encouraging results, sometimes even superior to those of the teacher, as shown in the sixth and the last rows of Tab. 2 for the specific task of PPI_1.

Tab. 3 shows the knowledge amalgamation results from pre-trained single-label node classification teacher GNNs. The first student model, Student 1 in Tab. 3, is obtained by amalgamating two teachers that handle the classification tasks of *Computers* and *Photos*, respectively. With a lightweight architecture which is even smaller than every single teacher, the obtained Student 1 still yields competitive results compared with those of the teachers. We also perform knowledge amalgamation on three teachers that deal with *Cora*, *Citeseer*, and *Pubmed*, respectively. The obtained Student 2 also delivers comparable results with those of teachers, yet maintaining a more compact size.

Amalgamating Point Cloud Classification and Segmentation Models. The results of amalgamating pre-trained classification and part segmentation teacher models are shown in Tab. 4. We also demonstrate in Fig. 5 the corresponding visualization results of the teachers and student. With the proposed TSA module, the learned versatile student gains boost by at least 0.4 in mean class accuracy and 1.2 in mean class IoU. as shown in the last two lines of Tab. 4. Also, as can be observed in Fig. 5, the learned lightweight and multi-talented student can sometimes achieve even superior performance to those of the cumbersome teachers, demonstrating that the knowledge from one teacher can potentially benefit the task of the other.

7. Conclusions

In this paper, we introduce a novel model reusing task tailored for heterogeneous GNNs. Our goal is to learn a versatile and lightweight student GNN that masters the complete set of expertise of multiple heterogeneous teachers, yet without human-labeled annotations. Towards this end, we identified two key challenges, and propose a dedicated slimmable graph convolutional operation as well as a novel topological attribution map (TAM) to solve the dilemma. Experiments on single- and multi-label classification and point cloud segmentation-classification demonstrate that, the obtained student GNN, with a moderately compact size, achieves performances on par with or even superior to those of the individual teachers on their specialized tasks. In the our future work, we will strive to generalize the proposed TAM to other tasks beyond knowledge amalgamation.

Acknowledgements

This work was supported by Australian Research Council Projects FL-170100117, DP-180103424, IH-180100002, and IC-190100031.

References

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008. 2

[2] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. In *ICLR*, 2018. 6

[3] Xiao Chu, Wanli Ouyang, Wei Yang, and Xiaogang Wang. Multi-task recurrent neural network for immediacy prediction. In *ICCV*, 2015. 2

[4] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008. 2

[5] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 2

[6] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020. 2

[7] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *ICML*, 2018. 1, 2

[8] Pinghua Gong, Jiayu Zhou, Wei Fan, and Jieping Ye. Efficient multi-task feature learning with calibration. In *KDD*, 2014. 2

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017. 2

[10] Dan He, David Kuhn, and Laxmi Parida. Novel applications of multitask learning and multiple output regression to multiple genetic trait prediction. *Bioinformatics*, 2016. 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 1, 2, 3

[13] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *NeurIPS*, 2018. 2

[14] Yongcheng Jing, Xiao Liu, Yukang Ding, Xinchao Wang, Errui Ding, Mingli Song, and Shilei Wen. Dynamic instance normalization for arbitrary style transfer. In *AAAI*, 2020. 2

[15] Yongcheng Jing, Yang Liu, Yezhou Yang, Zunlei Feng, Yizhou Yu, Dacheng Tao, and Mingli Song. Stroke controllable fast style transfer with adaptive receptive fields. In *ECCV*, 2018. 2

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 2, 3, 5, 7

[17] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 2

[18] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018. 2

[19] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *AAAI*, 2018. 2

[20] Yan Li, Jie Wang, Jieping Ye, and Chandan K Reddy. A multi-task learning formulation for survival analysis. In *KDD*, 2016. 2

[21] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *CVPR*, 2020. 2

[22] Sihui Luo, Wenwen Pan, Xinchao Wang, Dazhou Wang, Haihong Tang, and Mingli Song. Collaboration by competition: Self-coordinated knowledge amalgamation for multi-talent student learning. In *ECCV*, 2020. 2

[23] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In *ICML*, 2019. 2

[24] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, 2015. 6

[25] Hoang NT and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019. 2

[26] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015. 2

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 7

[28] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 2

[29] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *ICLR*, 2020. 2

[30] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 1, 2

[31] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 2008. 3, 4, 6

[32] Chengchao Shen, Xinchao Wang, Jie Song, Li Sun, and Mingli Song. Amalgamating knowledge towards comprehensive classification. In *AAAI*, 2019. 2

[33] Chengchao Shen, Xinchao Wang, Youtan Yin, Jie Song, Sihui Luo, and Mingli Song. Progressive network grafting for few-shot knowledge distillation. *arXiv preprint arXiv:2012.04915*, 2020. 2

[34] Chengchao Shen, Mengqi Xue, Xinchao Wang, Jie Song, Li Sun, and Mingli Song. Customizing student networks from heterogeneous teachers via adaptive knowledge amalgamation. In *ICCV*, 2019. 2

[35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 2, 3, 5, 7

[36] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *AAAI*, 2018. 2

[37] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *TNNLS*, 2020. 2, 4

[38] Zhizheng Wu, Cassia Valentini-Botinhao, Oliver Watts, and Simon King. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *ICASSP*, 2015. 2

[39] Jianpeng Xu, Pang-Ning Tan, Lifeng Luo, and Jiayu Zhou. Gspartan: a geospatio-temporal multi-task learning framework for multi-location prediction. In *SDM*, 2016. 2

[40] Jianpeng Xu, Pang-Ning Tan, Jiayu Zhou, and Lifeng Luo. Online multi-task learning framework for ensemble forecasting. *TKDE*, 2017. 2

[41] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 2

[42] Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. Factorizable graph convolutional networks. In *NeurIPS*, 2020. 2

[43] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *CVPR*, 2020. 1, 4, 6, 7

[44] Yiding Yang, Xinchao Wang, Mingli Song, Junsong Yuan, and Dacheng Tao. Spagan: Shortest path graph attention network. In *IJCAI*, 2019. 2

[45] Jingwen Ye, Yixin Ji, Xinchao Wang, Kairi Ou, Dapeng Tao, and Mingli Song. Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more. In *CVPR*, 2019. 2

[46] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *TOG*, 35(6):1–12, 2016. 7

[47] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 2017. 1

[48] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 1, 2, 6, 7

[49] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 1

[50] Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2018. 2

[51] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014. 2

[52] Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. In *ICLR*, 2020. 2

[53] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018. 2

[54] Ling Zhou, Zhen Cui, Chunyan Xu, Zhenyu Zhang, Chaoqun Wang, Tong Zhang, and Jian Yang. Pattern-structure diffusion for multi-task learning. In *CVPR*, 2020. 2

[55] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, 2018. 1, 2

[56] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 2017. 3, 6