Learning Graphs for Knowledge Transfer with Limited Labels - Supplementary

Pallabi Ghosh Nirat Saini Larry S. Davis Abhinav Shrivastava
University of Maryland, College Park

1. Datasets - Extension

Citation Networks (Cora, Citeseer, Pubmed): The three datasets Cora, Citeseer and Pubmed [16, 19] that we use for semi-supervised learning are citation networks. So the nodes in the graphs are documents and edges are citation connections between them. Table 1 summarizes the characteristics of each of these datasets (also listed in [22]). These datasets are transductive meaning all training and test data samples are present while constructing the input graph, but the class labels for test samples are not used while training.

Table 1: Features of the 3 different datasets - Cora, Citeseer and Pubmed for semi-supervised learning

	Cora	Citeseer	Pubmed
# Nodes	2708	3327	19717
# Edges	5429	4732	44338
# Features/Node	1433	3703	500
# Classes	7	6	3
# Training Nodes	140	120	60
# Validation Nodes	500	500	500
# Test Nodes	1000	1000	1000

Kinetics: Kinetics [9] has 306245 video clips and we use Kinetics400 with 400 action classes. Similar to [5], we use the I3D network trained on Kinetics for pre-training and then finetune the network on the other datasets we use for zero/few-shot learning. We also use the classifier layer weights for the Kinetics classes in the I3D model for training the GCN for zero/few-shot learning. All Kinetics classes are a part of our GCN training class set.

UCF101: UCF101 [20] has 101 action classes with 13320 videos. We use the same test-train data splits as [5]. So there are 23 test classes (3004 videos) and 78 training classes. The only classes in UCF101 that are not in common with Kinetics are a part of the test set. [5] mention some changes to action names such as "front crawl" becoming "front crawl swimming" for an improved input Knowledge Graph (KG). We keep these changes for an accurate comparison with the baseline. While constructing the input KG, it has nodes for the classes in UCF101 as well as nodes from 400 classes in Kinetics dataset. So there are 501 nodes in total.

HMDB51: HMDB51 [12] has 51 action classes with 6849 videos. The test-train split is same as [5] with 12 test classes (1541 videos) and 39 training classes. Again only the classes not in common with Kinetics are kept in the test set for zero/few-shot learning. Also like in UCF101, [5] make some changes to the action class names that make the input KG better such as changing all verbs to continuous tense (eg. "eat" to "eating"). We keep all these changes in our dataset as well. There are Kinetics class nodes in the input KG along with HMDB51 class nodes, same as described for UCF101. So there are a total of 451 nodes in the HMDB51 input KG.

Table 2 shows the classes of UCF101 and HMDB51 that are not in common with Kinetics and form the test sets.

2. Pipeline - Extension

System Overview for zero/few-shot learning Our zero/few-shot learning system is based on [5, 23] and we give an overview of this system in Algorithm1 and pipeline section of the main paper. We describe the baseline GCN system in further detail here. The feature extractor module is same across training and test classes. The only weights that are not available for test classes in zero/few-shot learning is the final classifier layer weights. To learn these weights, the system consists of two phases. Let C^{train} be the number of training classes and C^{test} be the number of test classes. In the training phase, first we finetune a I3D pretrained network for the training classes that predicts an embedding feature for each video of dimension d. It also gives us the final classifier layer weights for the training classes, W^{cls}, of dimension $C^{\text{train}} \times d$. There is a separate input KG based on inter-relationships among classes and this KG is passed though the GCN network which outputs a vector of dimension d per node in the graph. So we take the outputs of the GCN for the training nodes, H^{train}, which will be of dimension $C^{\text{train}} \times d$ and compare it to the final classifier layer weights provided by I3D using MSE loss. This MSE loss is given by $\|H^{\text{train}} - W^{\text{cls}}\|_2$ and it is backpropagated to train the GCN.

Table 2: Test classes from UCF101 and HMDB51 that are not in common with Kinetics Dataset

UCF101 test classes				
apply eye makeup	apply lipstick	balance beam	billiards	field hockey penalty
front crawl	hammering	handstand walking	jumping jack	mixing batter
nunchucks	parallel bars	playing daf	playing dhol	playing sitar
playing tabla	pommel horse	rafting	still rings	table tennis shot
typing	uneven bars	yo yo		

HMDB51 test classes					
chew	climb stairs	fall floor	handstand	pour	shoot gun
sit	smile	stand	talk	turn	wave

In the testing phase we take the output of the GCN for the test class nodes, H^{test} of dimension $C^{\text{test}} \times d$. This gets multiplied with the feature extracted from samples in test classes, f^{test} to give us the predicted class probability for test classes, $P^{\text{test}} = f^{\text{test}}(H^{\text{test}})^T$.

To show performance on GCN zero-shot we use the `A-KG` network described in [5] and our main paper, along with the hyperparameters for GCN training as used by [5]. We use 1 layer in GCN_1 and 5 layers in GCN_2 (both GCNs described in main paper). The intermediate dimensions of the 5 layers are $512 \rightarrow 1024 \rightarrow 1024 \rightarrow 1024 \rightarrow 1024$. The learning rate used is 0.001 with a weight decay of 0.0001. The learning rate scheduler is a stepwise scheduler which drops to 0.999 of the previous value at every 100 steps. To calculate loss we use the weighted summation of MSE loss based on the specific dataset nodes (HMDB51 and UCF101) and Kinetics nodes (baseline GCN training loss) and also triplet loss as described in the main paper.

For few-shot learning we use either `V-KG` or a combination of `V-KG` with `A-KG` and `VN-KG` as explained in [5] and our main paper. The GCN network as well as the optimization parameters remain the same as zero-shot learning (ZSL), except for UCF101, where the learning rate becomes 0.00005 and there is no weight decay. The last layer of the 5 layers belonging to GCN_2 is the fusion GCN layer for systems based on multiple KGs as used by [5]. For merging the output of different KGs, we concatenate the output of first 4 layers of GCN_2 from the different KGs along the channel dimension and finally pass it through the fusion layer which is a single layer GCN.

Additional Parameters The β parameter in the main paper defines the weight factor for combination of triplet loss with MSE loss. For HMDB51 this β factor is 0.1 whereas for UCF101 we do not use β at all and just sum up the losses. For Cora, Citesser and Pubmed, β is 0.8, 0.05 and 0.1 respectively.

Stopping criterion For both semi-supervised learning and zero/few-shot learning we train for a fixed number of epochs and choose the model at the best validation performance for estimating results on test set.

Table 3: Impact of using triplet loss and updating adjacency matrix.

triplet loss	update A	mean accuracy Cora	Citeseer	Pubmed
		80.0	72.0	77.8
✓		81.9	72.0	77.9
	✓	83.3	74.3	79.5
✓	✓	83.6	74.3	79.8

Table 4: Different # of clusters/class for the soft-triple loss.

Clusters per class	2	4	6	8	10
Cora	83.6	82.3	82.3	82.1	82.1
Citeseer	71.5	73.2	71.6	72.9	74.3
Pubmed	79.8	79.7	79.7	79.8	79.2

3. Results-Extension

We have results from ablation experiments in Table 4 of the main paper showing results after decoupling adjacency matrix update and triplet loss for UCF101 V-KG based few-shot learning. We also provide these ablation experiments on Cora, Citeseer and Pubmed test datasets in Table 3 here. For Citeseer, triplet loss does not help, but to build a generic network architecture that work for different kinds of tasks, use of triplet loss can be demonstrated through these results. We conducted analysis for the robustness of the semi-supervised learning experiments to number of clusters per class for the soft-triple loss on semi-supervised learning test datasets and the results are in Table 4.

We run an ablation experiment on `A-KG` for UCF101 validation set, where we use the updated adjacency matrix A^{updated} (defined in main paper) for GCN_1 as well as GCN_2 and our performance drop to 47.85% vs. the original result of 54.41% (in our main paper) for using A^{updated} only for GCN_2.

The different works we compare to in Table 1 and Table 7 from our main paper are SemiEmb[24], DeepWalk[17], ICA[13], Planetoid[25], Chebyshev[2], GCN[10], MoNet[15], GAT[22], GLNN[4], GCN+GDC[11], H-GCN[7], GLCN[8], ESZSL[18], DEM[26], TS-GCN[3], Ghosh *et al.* [5], Mettes *et al.* [14], UR[27], Action2vec[6]

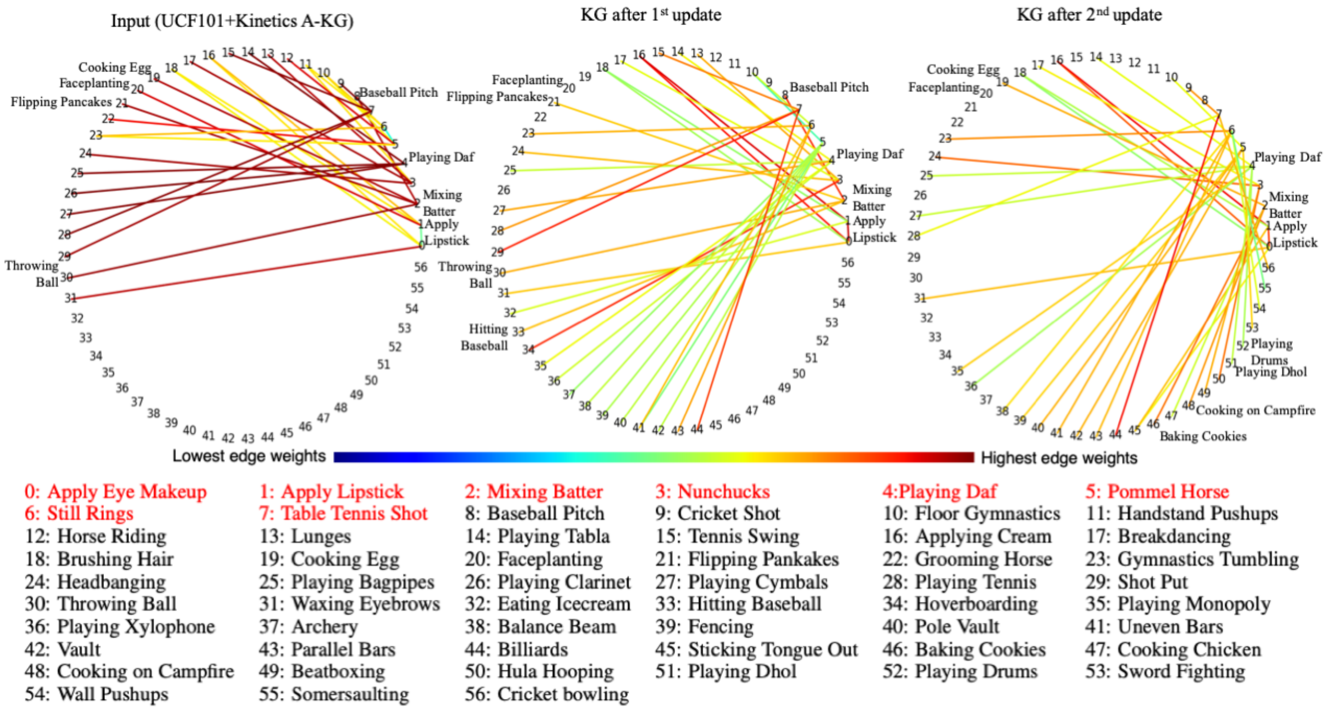

Figure 1: We plot the adjacency matrix connections for UCF101+Kinetics A-KG input and show the following two updates. We plot only a sub-graph due to space complexity, so we chose 8 test classes (shown in red in list of class names) and display all their connections in the KG. The red classes can connect to any other class. The edge colors show the weight of the connection. There are multiple regions where we can see improvements after first and second update. Best viewed in digital.

and TARN[1], and more details about them are as follows:

SemiEmb [24] combines nonlinear embeddings for shallow semi-supervised learning with deep learning where these techniques act as a regularizer on the output layer or at each intermediate layer of a deep network.

DeepWalk [17] extends existing language based and unsupervised learning approaches to graph based multi-label classification tasks.

ICA [13] uses structured logistic regression for a link based model that captures both link information as well as features of the objects connected by these links.

Planetoid [25] trains semi-supervised learning for graph data where the embeddings are jointly encoded for better classification and capturing neighborhood context.

Chebyshev [2] uses spectral graph theory to generalize convolutional neural networks to graph data.

GCN [10] uses a localized first order approximations of spectral graph convolutions for efficient and scalable graph convolution networks.

MoNet [15] develops generalized convolutional neural networks to be applied on graphs and manifolds which are non-euclidean domains.

GAT [22] uses self attention layers on graph convolution networks without any costly matrix inversion operation.

GLNN [4] optimizes the adjacency matrix of a graph with multiple objectives like sparsity constraint, properties of valid adjacency matrix etc.

GCN+GDC [11] replaces message passing in graphs with graph diffusion convolutions that combines advantages of spatial and spectral methods.

H-GCN [7] aggregates similar nodes into hyper-nodes and then refines each hyper-node to get back the original node embeddings which increases the receptive field of each node and captures global information.

GLCN [8] is an integrated graph learning and graph convolutional network that estimates the optimal graph structure for better results.

ESZSL [18] learns relationships between features, attributes, and classes in ZSL using two simple linear layers. Their system is not graph based and they provide results for zero-shot image classification.

DEM [26] maps the language embedding directly to the image feature space for ZSL and does not use an intermediate space. They are not a graph based system and provide results for zero-shot image classification.

TS-GCN [3] is GCN based, using Conceptnet [21] to construct a KG for both actions and objects. They have a second channel of visual object descriptors and show that they achieve their best results when selecting 2000 most common visible objects in their dataset. So they show their best performance in the transductive mode needing the test data without the labels during training.

[5] uses a GCN based system that has already been described in Section 2 and we use them as the baseline. We use triplet loss and adaptive learning of the adjacency matrix to improve on their results.

[14] uses local and global object awareness for better human object interaction detection. They are not graph based.

UR [27] preserve essential visual and semantic information in shared space to generate a universal representation that can generalize to new datasets for ZSL. They are not graph based.

Action2vec [6] develops a cross-modal embedding space combining language descriptors with spatio-temporal video features. They are also not a graph based system.

TARN [1] uses attention for temporal alignment and the network learns to align using a deep distance metric at the video segment level. They are not a graph based system and give results for zero and few-shot learning.

4. Discussion-Extension

In Figure 1 we point out additional example classes in Figure 4 from the main paper, where the graph connections show how the adjacency matrix update helped. We look at the "Mixing Batter" test class. Due to the presence of the word "batter", the language based KG associates it with "baseball" and "batting". After the first update these "baseball" related classes are still its neighbors, but after the second update they are replaced by "Baking cookies" and "Cooking on campfire". Similarly "Apply Lipstick" is connected to "Faceplanting" class initially which gets removed after the update. Also "Playing Daf" builds connections to "Playing Dhol" and "Playing Drums" after the second update.

References

[1] Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition, 2019. 3, 4

[2] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 2, 3

[3] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8303–8311, 2019. 2, 3

[4] Xiang Gao, Wei Hu, and Zongming Guo. Exploring structure-adaptive graph learning for robust semi-supervised classification. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020. 2, 3

[5] Pallabi Ghosh, Nirat Saini, Larry S. Davis, and Abhinav Shrivastava. All about knowledge graphs for actions, 2020. 1, 2, 4

[6] Meera Hahn, Andrew Silva, and James M Rehg. Action2vec: A crossmodal embedding approach to action learning. *arXiv preprint arXiv:1901.00484*, 2019. 2, 4

[7] Fenyu Hu, Yanqiao Zhu, Shu Wu, Liang Wang, and Tieniu Tan. Hierarchical graph convolutional networks for semi-supervised node classification. *arXiv preprint arXiv:1902.06667*, 2019. 2, 3

[8] Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11313–11320, 2019. 2, 3

[9] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[10] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2, 3

[11] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In *Advances in Neural Information Processing Systems*, pages 13354–13366, 2019. 2, 3

[12] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. 1

[13] Qing Lu and Lise Getoor. Link-based classification. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 496–503, 2003. 2, 3

[14] Pascal Mettes and Cees G. M. Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 4453–4462, 2017. 2, 4

[15] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017. 2, 3

[16] Galileo Namata, Ben London, Lise Getoor, and Bert Huang. Query-driven active surveying for collective classification. In *10th International Workshop on Mining and Learning with Graphs*, volume 8, 2012. 1

[17] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014. 2, 3

[18] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161, 2015. 2, 3

[19] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008. 1

[20] Khurram Soomro, Amir Roshan Zamir, and M Shah. A dataset of 101 human action classes from videos in the wild. *Center for Research in Computer Vision*, 2(11), 2012. 1

[21] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press, 2017. 3

[22] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. 1, 2, 3

[23] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6857–6866, 2018. 1

[24] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural networks: Tricks of the trade*, pages 639–655. Springer, 2012. 2, 3

[25] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016. 2, 3

[26] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2021–2030, 2017. 2, 3

[27] Yi Zhu, Yang Long, Yu Guan, Shawn Newsam, and Ling Shao. Towards universal representation for unseen action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9436–9445, 2018. 2, 4