Supplementary Materials for Learning Hierarchical Graph Neural Networks for Image Clustering

Yifan Xing* Tong He* Tianjun Xiao Yongxin Wang Yuanjun Xiong
Wei Xia David Wipf Zheng Zhang Stefano Soatto

Amazon Web Services
{yifax, htong, tianjux, yongxinw, yuanjx, wxia, daviwipf, zhaz, soattos}@amazon.com

1. Hi-LANDER Clustering Visualization

We visualize in Figure 1 the hierarchical clustering process of the proposed method Hi-LANDER. We show three ground-truth clusters that differ in cluster sizes and embed their features into a 2D plane with t-SNE, and then visualize the points (as shown on the left column). The blue squares are the input nodes at each level of the hierarchy. The colored dots are peak nodes that are grouped from the intermediate clusters (connected-components), and the colors represent the three different ground-truth classes. Note that the peaks at each level then become ordinary input nodes at the next level.

We see that the nodes in the red cluster are grouped efficiently with only one peak node left in level 1, while there are many small clusters for the yellow and green class nodes. In the next hierarchy, as shown in the second row, the distance between each pair of the peak nodes is larger, and the number of peaks reduced rapidly. The red cluster stays unchanged since our base clustering model LANDER stops adding edges, while the green and yellow clusters are further grouped. The last row shows the final level where all three classes converge, and only nodes belonging to the yellow cluster are further grouped.

Besides, on the right column of Figure 1, we demonstrate the actual face images corresponding to the peak nodes at each level of the hierarchical clustering process for all three classes. Compared to level 2 peaks, the images corresponding to level 1 peaks are more "repetitive." If we run a prior GNN based clustering model that only produces a single partition, each "repetitive" level 1 peak will lead to a separate cluster, and this results in low clustering completeness. In level 2, the large number of small clusters corresponding to the yellow class are grouped into 4 larger clusters. As shown in the second row of the right column in Figure 1, the images correspond to the peak nodes of these 4 clusters (with the yellow boundary) become less visually similar, while one can tell that they still represent the same person. Note that the three classes converge at different levels. Nodes of the red class already converge at the first level, the green class nodes converge at level2 while the yellow class requires all three levels to reach convergence. This illustrates the variance of real-world test data where the instance per class can be very different from class to class and it demonstrates Hi-LANDER's capability in dealing with such large variance.

*Indicates equal contribution.

2. Experiment Details

Here we describe additional experiment details including dataset statistics, input feature dimensions, sensitivity tests on Hi-LANDER hyper-parameters, and the runtime hardware and software specifications. Code is included in the supplementary zip file.

Dataset	Images	Entities	Mean Cluster Size
TrillionPairs-Train [1]	669,560	18,084	37.0
Hannah-Test [6]	201,240	251	801.8
IMDB-Test [10]	1,265,173	50,289	25.2
IMDB-Test-SameDist [10]	614,002	18,084	34.0
iNat2018-Train [9]	324,418	5,690	57.0
iNat2018-Train-DifferentDist [9]	51,696	5,690	9.0
iNat2018-Test [9]	135,660	2,452	55.3

Table 1. Statistics of All Datasets

Dataset Statistics Table 1 shows the detailed dataset statistics for all train and test sets used for the experiments.

Hannah	IMDB	iNat2018
128	128	512

Table 2. Input Feature Dimensions For All Datasets.

Input Feature Table 2 lists the input feature dimensions for all datasets. L_2-normalization is applied on the features before network inference.

Sensitivity Analysis over Hyper-parameters Figure 2 shows the sensitivity of Hi-LANDER to the various hyper-

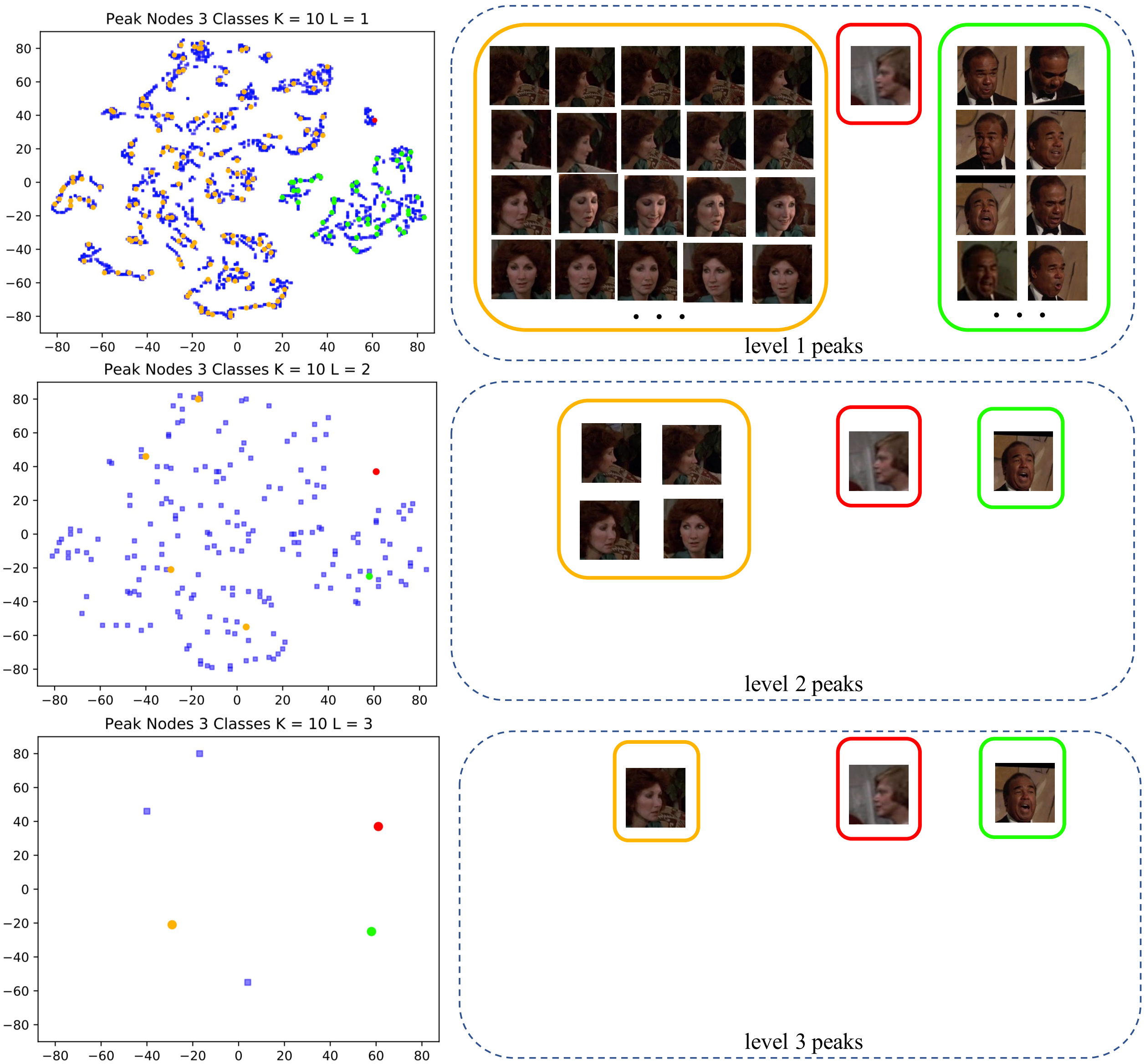

Figure 1. Hi-LANDER clustering process visualization on Hannah with multiple image classes. The yellow, red and green color represent three different classes that vary in cluster size. The left column shows the t-SNE [8] embedded nodes and peaks from level 1 to level 3 of Hi-LANDER's hierarchy. At each level, blue squares represent the input nodes and colored dots refer to the peak nodes which are grouped from the intermediate clusters (connected-components). Note that the peaks at each level then become the input nodes at the next level. The right column shows the images corresponding to the peaks at each level of the three classes. The three classes converge at different levels: nodes of the red class already converges at the first level, the green class converge at level2 while the yellow class converge at level3. Best viewed in color.

parameters of the method including k for k-NN build, p_τ for edge set decoding, the feature aggregation function choice detailed in Section 3.4 of the main paper, and the encoder layer architecture choice (GAT versus a vanilla GCN layer), mentioned in Section 3.3 of the main paper. The top two plots show the sensitivity of k, p_τ and the feature aggregation mechanism, where solid lines refer to identity feature aggregation and dashed lines represent the concatenation of identity and average feature. The bottom two plots show the sensitivity to the two different types of encoder layer architecture, a GAT (solid lines) and a vanilla GCN layer (dotted lines). Based on the validation set (a part of the meta-training set), with GAT encoding, the optimal hyperparameters over the face clustering task are chosen as k = 10, p_τ = 0.9, aggregation using identity feature only. Thus, for k sensitivity, we vary it from 8 to 12. For p_τ sensitivity, we vary it from 0.85 to 0.95 with interval of 0.025. Metrics of NMI (blue), Fp (yellow), and Fb (red) are shown.

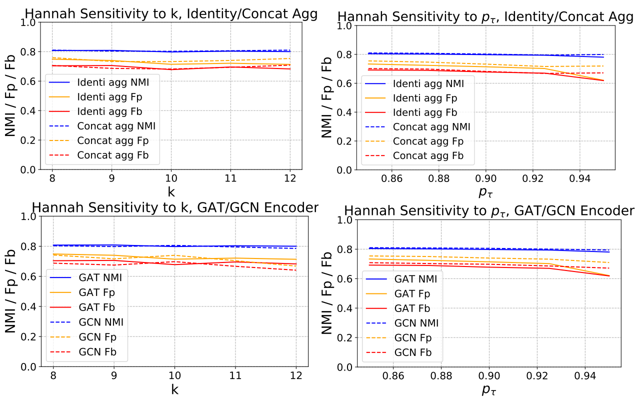

Figure 2. Sensitivity to hyper-parameters on the Hannah face clustering benchmark. The top two plots show sensitivity of Hi-LANDER to the hyper-parameters of k, p_τ and the feature aggregation mechanism, where solid lines show the results of identity feature aggregation and dashed lines show the results from concatenation of identity and average feature. The bottom two plots show sensitivity of Hi-LANDER to different types of encoders, a GAT layer (solid lines) as compared to a vanilla GCN layer (dotted lines), as detailed in Section 3.3 of the main paper. For k sensitivity tests, we vary it around the optimal value of 10 (chosen on the validation sets) from 8 to 12. For p_τ sensitivity tests, we vary it around the optimal value of 0.9, from 0.85 to 0.95 with interval of 0.025. All three clustering metrics of NMI (blue), Fp (yellow) and Fb (red) are shown. Best viewed in color.

The plots show that varying k and p_τ near the optimal value does not result in significant changes in results. The differences in final clustering accuracy between identity-feature-only aggregation and concatenation of both identity and average feature, as well as the variations between using GAT versus a vanilla GCN layer in encoding, are small.

Additional Clustering Benchmark with Unseen Test Data Distribution Besides large-scale face datasets such as IMDB and Hannah, we also test on the smaller IJB-B/C datasets. Table 3 compares against the best-performing prior methods of unsupervised (DBSCAN [3]), hierarchical unsupervised (H-DBSCAN [2]), and supervised (GCN-V+E [12]) baselines. Additionally, we test on another video dataset, MusicVideos [13] with 8 videos and 95k faces from 40 identities. Similar phenomenon as Hannah video testing is observed, where Hi-LANDER (0.472 average F-score) significantly outperforms all baselines (next best from GCN-V+E, 0.410).

Method	IJB-B clustering task		IJB-C clustering task		MusicVideos	
	Avg F-score	NMI	Avg F-score	NMI	Avg F-score	NMI
DBSCAN [3]	0.214	0.809	0.271	0.841	0.026	0.500
H-DBSCAN [2]	0.677	0.902	0.703	0.924	0.184	0.540
GCN-V+E [12]	0.759	0.944	0.769	**0.953**	0.410	0.682
Hi-LANDER	**0.820**	**0.945**	**0.820**	0.952	**0.472**	**0.704**

Table 3. We use the largest protocols with all subjects in IJB-B/C; Average F-score between Fp and Fb is reported.

Method	Hannah		IMDB		IJB-B		IJB-C		MusicVideos		iNat2018-Test	
	Avg F	NMI	Avg F	NMI	Avg F	NMI	Avg F	NMI	Avg F	NMI	Avg F	NMI
GAT	0.695	0.797	0.774	0.945	0.820	0.945	0.820	0.952	**0.472**	0.704	0.323	**0.764**
GCN	**0.723**	**0.809**	**0.799**	**0.949**	**0.891**	**0.959**	**0.887**	**0.964**	0.451	**0.709**	**0.345**	0.759

Table 4. Ablation: GAT versus vanilla GCN in graph encoding.

GCN vs GAT Encoding Ablation Table 4 shows the clustering performance ablation about using GAT versus a vanilla GCN layer in graph encoding over tests with unseeen data distribution. Both models have their respective hyper-parameters tuned to optimal over the validation sets. It is observed that the two encoding achieves similar performances. GAT encoding outperforms vanilla GCN over the MusicVideo and iNat2018-Test benchmarks over the average F-score and NMI metrics respectively while GCN outperforms GAT over the rest tests.

Additional Training Details For the base clustering model LANDER, we use 1 layer of GAT as encoder and a 2-layer MLP for joint linkage and density prediction. Both face and nature species models are trained for 250 epochs with batchsize 4096. All models use SGD optimizer with 0.1 base learning rate, 0.9 momentum, and 1e-5 weight decay. The learning rate follows a cosine annealing schedule [5].

Runtime Experiment Hardware and Software We measure the runtime (Section 4.7 of the main paper) with 8-core Intel(R) Xeon(R) E5-2686 v4 CPU and Tesla V100 GPU. Our models use PyTorch[7] v1.5, DGL[11] v0.6 with CUDA v10.1. k-NN building leverages faiss[4].

Runtime Experiment Additional Analysis Details Hi-LANDER can be slower than FINCH/Graclus per hierarchical iteration since the latter has no or lightweight model inference overhead. However, Hi-LANDER runs the fastest on Hannah because 1) Hannah has many largely similar nodes that are easily merged, greatly reducing the number of nodes to cluster for next iterations (16x $\downarrow$ after the 1^{st} iteration) thus decreasing subsequent inference cost. 2) Hi-LANDER runs 4 iterations to converge on Hannah, fewer than 8 in FINCH which converges at fewer nodes. Against GCN-V, Hi-LANDER per iteration is faster due to the smaller graph neighborhood (k) but recurrent iterations make it slower (IMDB/iNat). However, it is faster on Hannah with a cost close to a single iteration as overhead after the 1^{st} iteration is marginal.

References

[1] http://trillionpairs.deepglint.com/overview. 1

[2] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013. 3

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. 3

[4] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. 3

[5] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 3

[6] Alexey Ozerov, Jean-Ronan Vigouroux, Louis Chevallier, and Patrick Pérez. On evaluating face tracks in movies. In *2013 IEEE International Conference on Image Processing*, pages 3003–3007. IEEE, 2013. 1

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019. 3

[8] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 2

[9] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 1

[10] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. *arXiv preprint arXiv:1807.11649*, 2018. 1

[11] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, et al. Deep graph library: Towards efficient and scalable deep learning on graphs. *arXiv preprint arXiv:1909.01315*, 2019. 3

[12] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13369–13378, 2020. 3

[13] Shun Zhang, Yihong Gong, Jia-Bin Huang, Jongwoo Lim, Jinjun Wang, Narendra Ahuja, and Ming-Hsuan Yang. Tracking persons-of-interest via adaptive discriminative features. In *European Conference on Computer Vision*, pages 415–433. Springer, 2016. 3