
A Note on Diversifying Attention for Hierarchical and Explicit Label Modelling with Graph Learning for Multi-Label Image Classification (HELM)

Filip Josifljević¹ Dubravko Ćulibrk^{2,3}

¹RD Institute for Artificial Intelligence of Serbia

²Jožef Stefan Institute, Slovenia

³University of Novi Sad, Serbia

Abstract The HELM approach to hierarchical category modeling in multi-label image classification relies on hierarchy-specific tokens within a Vision Transformer, a GraphSAGE operator, and a self-supervised Build Your Own Latent (BYOL) branch to model category structure and leverage unlabeled data. Here, we examine in more detail a recently published study on attention allocation within the model, and the introduction of attention diversity regularization proposed there in order to improve its performance and insure that hierarchy-specific tokens learn distinct representations. In addition, we investigate whether these tokens develop spatially distinct attention patterns over the image. Despite the fact that the addition of the regularization is able to enhance the diversity at the representation level, this leads to significantly improved performance on only one of the three datasets used in our evaluation. More importantly, it does not ensure this spatial specialization of the tokens. To support reproducibility, we publish the code and the constructed label hierarchies.

1 Introduction

Classification of diverse scenes in the field of remote sensing often requires assigning more than one class to an image, as the same scene may belong to categories at different levels of abstraction. Traditional multi-label classification approaches treat all categories as equally important and mutually independent, overlooking the hierarchical structure that connects specific and general categories.

The development of approaches able to incorporate domain knowledge about label relations is impeded by the scarcity of labeled data. Manually annotating images across multiple hierarchical levels simultaneously, requires considerable time and domain expertise. This makes fully supervised learning difficult to apply in practice and motivates the development of semi-supervised approaches that can leverage a small number of labeled examples alongside a large amount of unlabeled data.

HELM (Hierarchical and Explicit Label Modeling with Graph Learning) [10] is an architecture that explicitly models the hierarchical structure of categories through three interconnected components: a classification branch based on hierarchy-specific CLS tokens within a Vision Transformer [4]; a graph learning branch that propagates information along the category hierarchy using the GraphSAGE operator [3]; and a self-supervised learning branch based on the BYOL method [2], which enables the utilization of unlabeled data.

The original HELM paper states that hierarchical-specific tokens “evolve into semantically meaningful representations that represent specific categories”, but it does not explicitly evaluate whether these tokens indeed develop different values and whether they also develop spatially distinct attention patterns over the image. A recent study focused on reproducibility of HELM [7] indicated that, without special care when it comes to initialization and the addition of diversity-enhancing loss the hierarchy-specific tokens in do not learn distinct representations. Also they do not develop spatially distinct patterns of attention over the image.

Here we conduct a thorough experimental evaluation of effects reported in [7], publishing the code and label hierarchy for the different benchmark datasets used (at <https://github.com/FilipJosifljevic/hierarchical-remote-sensing>), which was a limiting factor of the original HELM paper by Stoimchev *et al.* [10]. Our experiments show that the addition of a regularizer, aimed at enhancing diversity, to the loss function does not consistently improve performance and that the networks is able to learn distinct representation for the category-specific tokens, contrary to what was found in [7]. However, neither the extended nor the original approach ensure that the category-specific tokens learn to attend to different regions in the input images. Further research into the methods to achieving this is needed.

2 Method

Unlike the conventional approach, in which the category tree is manually encoded based on the dataset classification, the hierarchical structure was derived directly from the binary category matrix. For each pair of categories, we examined whether the set of images in which category X is active constitutes a strict subset of the set of images in which category Y is active; if so, Y was considered an ancestor of X in the hierarchy. The immediate parent of each category was then determined as the ancestor with the smallest number of positive samples (i.e., the closest corresponding superset). Using this procedure, the hierarchy was automatically reconstructed to match the structure reported in Stoimchev *et al.*: four categories at the highest level, nine intermediate categories, and seventeen leaf categories, resulting in a total of 30 nodes. An approach based exclusively on subset relationships proved insufficient in two cases. First, sibling categories that consistently co-occur in a small dataset (e.g., "sand" and "sea" in beach scenes) may falsely appear as parent-child pairs. This was addressed by restricting the set of candidate ancestors exclusively to known categorical nodes. Second, certain nodes cannot be distinguished based solely on the data when they correspond to identical sets of positive samples in the dataset. This

was addressed by explicitly adopting a part of the hierarchy from the CORINE [8] nomenclature, according to which the data was originally categorized in [10].

Following the HELM approach, a pretrained vision transformer [1] was extended by adding M learnable CLS tokens, where M corresponds to the total number of nodes in the hierarchy ($M = 30$). These tokens, together with the image patch tokens, pass through the standard self-attention blocks, allowing each token to evolve into a representation specific to its corresponding category. After average pooling, these tokens are directly fed into the classification branch through a single fully connected layer.

The hierarchy-specific tokens were, in the original approach by Stoimchev *et al.*, initialized by copying the pretrained CLS token of the base model M times, together with the corresponding positional embedding. Josifljević observed that this initialization, when applied identically to all tokens (i.e., using the same values and the same positional embedding), results in persistent mathematical symmetry: the self-attention mechanism applies identical weights to all positions, and identical input tokens consequently produce identical outputs at every layer of the network, regardless of training. To address this problem he proposed adding a small, independent Gaussian noise ($\sigma = 0.02$) to the initial values of each token and its positional embedding. This successfully breaks the symmetry while preserving the benefits of a warm start from the pretrained weights and we have, therefore, employed this approach in all the experiments conducted in the present study.

The same representation of the hierarchy-specific tokens also serves as the initial node features for the graph derived from the hierarchy, on which a GraphSAGE operator is applied. A semi-supervised learning approach is used. The branch processes the entire batch, including both labeled and unlabeled samples, while the loss function is computed only over the labeled samples. In parallel, the BYOL branch uses patch tokens from two differently augmented views of the same image, enabling learning from fully unlabeled examples as well.

The diversity loss introduced by Josifljević is detailed in the Supplemental Material (Section B).

3 Experiments and Results

The evaluation was conducted on three datasets: UC Merced Land Use dataset extended with hierarchical multi-label categories (UCM-HMLC) [11] ; Aerial Image Dataset (AID) [6] ; and the DFC15 dataset [5] , which is a multi-label remote sensing image classification benchmark derived from the 2015 IEEE GRSS Data Fusion Contest [9] dataset. The details of the datasets used and the construction of the corresponding label hierarchies are provided in Section A.

Rather than following the approach of Stoimchev *et al.* who evaluate the model by varying the proportion of labeled data we limit our experiments to 10% labeled data in the training set, with the remaining samples treated as unlabeled and utilized exclusively through the graph and self-supervised branches.

A standard, supervised ImageNet-pretrained ViT-Base model was used as the encoder backbone and further trained for 100 epochs using the AdamW optimizer with a cosine annealing learning rate schedule (initial learning rate 1×10^{-4}) and a batch size of 16. Training was performed using an NVIDIA V100 GPU.

Following the metrics reported by Stoimchev *et al.*, the results were evaluated using AUPRC (area under the precision–recall curve), macro-averaged across all 30 categories, and the ranking loss metric. Both metrics were computed on the fully labeled test set.

Contrary to the findings in [7], the baseline no longer overfits, improving smoothly across all 100 epochs, and seems to outperform the diversity-loss variant on the AID and DFC15 datasets as shown in Table 1. However, t-tests conducted on the sequences of results obtained for these two datasets indicate that the differences between the two approaches are not statistically significant. However, on UCM-HMLC the diversity loss does result in significantly increased performance.

Direct re-measurement of attention allocation of the trained models shows that the routing loophole previously identified by Josifljević (hierarchy tokens directing the majority of their



Figure 1: Spatial attention patterns for the “cars” and “trees” category tokens.

Table 1: AUPRC and Ranking Loss comparison across training configurations (10% labeled data, 100 epochs, cosine schedule, supervised backbone, mean $\pm$ std over 3 seeds unless noted)

Dataset	Configuration	λ	Final AUPRC	Best AUPRC	Final Ranking Loss
UCM-HMLC	Baseline	–	0.9361 ± 0.0130	0.9386 ± 0.0129	0.0212 ± 0.0009
UCM-HMLC	+ Diversity loss	0.1	0.9318 ± 0.0136	0.9359 ± 0.0130	0.0217 ± 0.0018
UCM-HMLC	+ Diversity loss	0.7	0.9403 ± 0.0080	0.9406 ± 0.0079	0.0210 ± 0.0040
AID	Baseline	–	0.8724 ± 0.0158	0.8729 ± 0.0157	0.0205 ± 0.0005
AID	+ Diversity loss	0.1	0.8631 ± 0.0124	0.8641 ± 0.0113	0.0207 ± 0.0010
AID	+ Diversity loss	0.7	0.8464 ± 0.0106	0.8467 ± 0.0104	0.0228 ± 0.0017
DFC-15	Baseline	–	0.9109 ± 0.0090	0.9114 ± 0.0097	0.0249 ± 0.0035
DFC-15	+ Diversity loss	0.1	0.8989 ± 0.0070	0.9045 ± 0.0023	0.0256 ± 0.0024
DFC-15	+ Diversity loss	0.7	0.8934 ± 0.0106	0.9035 ± 0.0055	0.0265 ± 0.0041

attention toward other hierarchy tokens rather than the input image) no longer occurs. All measured tokens now allocate the substantial majority of their attention to image patches. However, qualitative and quantitative inspection of the resulting attention maps reveals that this does not correspond to the genuine spatial specialization the loss was designed to induce. Tokens corresponding to semantically distinct categories continue to converge on an identical spatial region of the input image (as shown in Figure 1, with pairwise cosine similarity between per-token attention distributions at its theoretical maximum).

4 Conclusion

We conducted a detailed study of the effect of introducing explicit diversity loss to combat the problem with attention allocation observed in the Hierarchical and Explicit Label Modeling with Graph Learning for Multi-Label Image Classification approach. Although a recent, more limited study, indicated that such a modification could be beneficial, our results show that it does not affect performance with statistical significance across the board, but is decidedly beneficial on one of the datasets evaluated on. We publish all the code used in the study, as well as the constructed label hierarchies. Together with already available datasets used, these make our results fully reproducible.

References

- [1] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- [2] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020). Bootstrap your own latent—a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284.
- [3] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30:1024–1034.
- [4] Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al. (2022). A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110.
- [5] Hua, Y., Mou, L., and Zhu, X. X. (2019). Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149:188–199.
- [6] Hua, Y., Mou, L., and Zhu, X. X. (2020). Relation network for multi-label aerial image classification. *IEEE Transactions on Geoscience and Remote Sensing*.
- [7] Josifljević, F. (2026). Reproduction and analysis of the HELM architecture for satellite image classification. *Proceedings of the Faculty of Technical Sciences*, 2(1). Accepted for publication.
- [8] Kosztra, B., Büttner, G., Hazeu, G., and Arnold, S. (2017). Updated CLC illustrated nomenclature guidelines. Service contract no. 3436/r0-copernicus/eea.57441, task 3, d3.1 — part 1, European Topic Centre on Urban, Land and Soil Systems (ETC/ULS), European Environment Agency, Wien, Austria. Revised 10 May 2019.
- [9] Moser, G., Tuia, D., and Shimoni, M. (2015). 2015 ieee grss data fusion contest: Extremely high resolution lidar and optical data [technical committees]. *IEEE Geoscience and Remote Sensing Magazine*, 3(1):40–41.
- [10] Stoimchev, M., Koloski, B., Levatić, J., Kocev, D., and Džeroski, S. (2026). HELM: Hierarchical and explicit label modeling with graph learning for multi-label image classification. Accepted and presented at the REO workshop at EurIPS 2025.
- [11] Yang, Y. and Newsam, S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 270–279, New York, NY, USA. Association for Computing Machinery.

1. For all authors...

- (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? Yes.
- (b) Did you describe the limitations of your work? Yes.
- (c) Did you discuss any potential negative societal impacts of your work? Not applicable.

- (d) Did you read the ethics review guidelines and ensure that your paper conforms to them? (see <https://2022.automl.cc/ethics-accessibility/>) Yes.
2. If you ran experiments...
- (a) Did you use the same evaluation protocol for all methods being compared (e.g., same benchmarks, data (sub)sets, available resources, etc.)? Yes.
 - (b) Did you specify all the necessary details of your evaluation (e.g., data splits, pre-processing, search spaces, hyperparameter tuning details and results, etc.)? Yes.
 - (c) Did you repeat your experiments (e.g., across multiple random seeds or splits) to account for the impact of randomness in your methods or data? Yes.
 - (d) Did you report the uncertainty of your results (e.g., the standard error across random seeds or splits)? Yes.
 - (e) Did you report the statistical significance of your results? No.
 - (f) Did you use enough repetitions, datasets, and/or benchmarks to support your claims? Yes.
 - (g) Did you compare performance over time and describe how you selected the maximum runtime? Yes.
 - (h) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes.
 - (i) Did you run ablation studies to assess the impact of different components of your approach? Yes.
3. With respect to the code used to obtain your results...
- (a) Did you include the code, data, and instructions needed to reproduce the main experimental results, including all dependencies (e.g., `requirements.txt` with explicit versions), random seeds, an instructive README with installation instructions, and execution commands (either in the supplemental material or as a URL)? Yes.
 - (b) Did you include a minimal example to replicate results on a small subset of the experiments or on toy data? No.
 - (c) Did you ensure sufficient code quality and documentation so that someone else can execute and understand your code? Yes.
 - (d) Did you include the raw results of running your experiments with the given code, data, and instructions? Yes.
 - (e) Did you include the code, additional data, and instructions needed to generate the figures and tables in your paper based on the raw results? Yes.
4. If you used existing assets (e.g., code, data, models)...
- (a) Did you cite the creators of used assets? Yes.
 - (b) Did you discuss whether and how consent was obtained from people whose data you're using/curating if the license requires it? Not applicable.
 - (c) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? Not applicable.
5. If you created/released new assets (e.g., code, data, models)...

- (a) Did you mention the license of the new assets (e.g., as part of your code submission)? Yes.
 - (b) Did you include the new assets either in the supplemental material or as a URL (to, e.g., GitHub or Hugging Face)? Yes.
6. If you used crowdsourcing or conducted research with human subjects...
- (a) Did you include the full text of instructions given to participants and screenshots, if applicable? Not applicable.
 - (b) Did you describe any potential participant risks, with links to institutional review board (IRB) approvals, if applicable? Not applicable.
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? Not applicable.
7. If you included theoretical results...
- (a) Did you state the full set of assumptions of all theoretical results? Not applicable.
 - (b) Did you include complete proofs of all theoretical results? Not applicable.

A Datasets and Label Hierarchy Construction

The UCM-HMLC contains 2,100 images distributed across 21 classes. The hierarchical annotation comprises a total of 30 nodes—17 leaf nodes, 9 intermediate categories, and 4 top-level categories—derived from the CORINE land cover classification. The dataset was split into 1,667 training images and 433 test images, following the split reported in the original work.

The AID dataset comprises 3,000 images distributed across 30 scene categories, annotated with a subset of 17 leaf-level labels identical to those used in UCM-HMLC. Given this identical leaf vocabulary, the hierarchical structure derived for UCM-HMLC was reused directly for AID, rather than inferred independently, since the two datasets’ leaf categories and the overall shape of the hierarchy (30 nodes: 17 leaves, 9 intermediate categories, 4 top-level categories) were confirmed to match exactly. The dataset was split into 2,400 training images and 600 test images, same as reported in the original work.

The DFC15 dataset comprises 3,342 images annotated with 8 leaf-level labels (impervious surfaces, water, clutter, vegetation, building, tree, boat, and car), derived from the 2015 IEEE GRSS Data Fusion Contest imagery. Unlike AID, DFC15’s leaf vocabulary differs entirely from UCM-HMLC’s. The original work constructs a three-level hierarchy for this dataset by mapping its categories to the CORINE Land Cover nomenclature, supplemented with LLM-assisted mapping and manual validation where direct CORINE mapping was infeasible, yielding 17 total nodes across three levels (3 top-level, 7 intermediate, 8 leaf categories). Unlike UCM-HMLC, whose full hierarchy (including exact category names and parent-child structure) is disclosed via an explicit diagram in the original paper, and which AID reuses directly, no detailed information is provided for DFC15, and the accompanying configuration files were not yet publicly available at the time of this work. Moreover, even had the CORINE-mapped categories been disclosed, the LLM-assisted mapping step used for categories without a direct CORINE correspondence is not a deterministic process and could not be reliably reproduced by independently repeating it.

In the absence of this information, we constructed DFC15’s hierarchy independently, using the subset-relationship method described in Section 2, applied with a relaxed containment threshold (0.95 rather than a strict 1.0) to account for small amounts of real annotation noise between conceptually related categories. This produced a substantially shallower structure than the one described by Stojimchev *et al.*: 8 nodes across two levels, with four top-level categories (impervious surfaces, water, vegetation, clutter) and four leaf categories nested directly beneath them (building and car under impervious surfaces; boat under water; tree under vegetation), omitting the intermediate level entirely, and consequently reducing the total node count from 17 to 8. As a result, DFC15 results reported in this work are not directly comparable to those reported in the original paper, but are fully reproducible. The dataset was split into 2,673 training images and 669 test images, following the split reported by the dataset’s original authors.

B Diversity Loss

To directly encourage spatial diversity, in [7] an auxiliary loss was introduced that penalizes the similarity between the attention distributions of different tokens over the image patch tokens:

$$L_{\text{div}} = \frac{1}{M(M-1)} \sum_{i \neq j} \frac{\mathbf{a}_i \cdot \mathbf{a}_j}{\|\mathbf{a}_i\| \|\mathbf{a}_j\|}$$

The weight (λ) controls the contribution of this auxiliary loss to the model’s overall loss function, and in the experiments it was set to a relatively small value so as not to compromise the primary objectives of classification and graph learning. The effect of this loss on the spatial distribution of attention is illustrated using the example of the “cars” and “trees” category pair in Figure 1. M represents the number of hierarchy tokens, with i and j ranging over all pairs of distinct tokens ($i \neq j$, so that a token’s trivial self-similarity of 1 is excluded from the sum). For each token, $\mathbf{a}_i$

denotes its attention distribution over the image’s patch tokens, a vector giving the amount of attention allocated to each individual patch. The term $\frac{\mathbf{a}_i \cdot \mathbf{a}_j}{\|\mathbf{a}_i\| \|\mathbf{a}_j\|}$ is the cosine similarity between two such vectors, and the leading factor $\frac{1}{M(M-1)}$ normalizes the sum over all $M(M-1)$ such pairs into a mean. L_{div} thus represents the average pairwise cosine similarity across all hierarchy tokens’ attention distributions.

Table shows the effect of different λ values on the performance of the approach, for the UCM-HMLC data set. The best performance is achieved for a fairly large value (0.7), suggesting that the simple diversity loss contributes more than other, more sophisticated loss terms.

Table 2: Effect of the diversity-loss weight λ on UCM-HMLC, compared against the no-diversity-loss baseline (10% labeled data, 100 epochs, cosine schedule, mean $\pm$ std over 3 seeds)

λ	Final AUPRC	Best AUPRC	Final Ranking Loss
Baseline (no loss)	0.9361 $\pm$ 0.0130	0.9386 $\pm$ 0.0129	0.0212 $\pm$ 0.0009
0.2	0.9320 $\pm$ 0.0065	0.9344 $\pm$ 0.0060	0.0232 $\pm$ 0.0028
0.3	0.9345 $\pm$ 0.0142	0.9358 $\pm$ 0.0142	0.0220 $\pm$ 0.0031
0.4	0.9365 $\pm$ 0.0111	0.9370 $\pm$ 0.0104	0.0229 $\pm$ 0.0014
0.5	0.9291 $\pm$ 0.0136	0.9385 $\pm$ 0.0041	0.0236 $\pm$ 0.0027
0.6	0.9357 $\pm$ 0.0081	0.9372 $\pm$ 0.0068	0.0224 $\pm$ 0.0017
0.7	0.9403 $\pm$ 0.0080	0.9406 $\pm$ 0.0079	0.0210 $\pm$ 0.0040
0.8	0.9348 $\pm$ 0.0105	0.9380 $\pm$ 0.0129	0.0234 $\pm$ 0.0021
0.9	0.9350 $\pm$ 0.0051	0.9387 $\pm$ 0.0070	0.0223 $\pm$ 0.0022