1 Introduction

The purpose of this repo is to provide code and a description of the methods employed by the Dal team in our efforts while participating in the FathomNet 2025 Kaggle competition (eor123 et al., 2025). Although we experimented with a number of different approaches, there are two methods that we credit with the majority of our progress: distance-weighted cross-entropy loss and ensemble-based self-training.

2 Installation

To install the pre-requisite libraries, run:

```
pip install -r requirements.txt
```

The `main.py` file acts as the script to run training models. The only argument for `main.py` is the training configuration file, which we supply examples of in the `cfg` directory. Lastly, we include a sample shell script `train_model.sh`, which may be of use to those running multiple jobs on computing clusters with SLURM scheduling.

3 Methods

3.1 Distance-Weighted Cross-Entropy

Let D be a distance matrix of dimension $N \times N$, where each entry d_{ij} represents the distance between class i and class j. In the case of FathomNet, these classes correspond to leaf nodes. Furthermore, since FathomNet is a full-depth hierarchical classification problem (meaning that appropriate annotations must be leaf nodes, and cannot terminate prior to reaching the full depth of the hierarchy), D is then related to the hierarchical distance, that also evaluates model predictions based on their prediction's "closeness" to ground truth. Therefore, we can also imagine employing D as a means of more heavily penalizing distance predictions. However there exists a number of issues. Firstly, we cannot minimize for distance directly or even scaling with the model probability outputs, as this tends to be quite unstable, and tends to result in poor performance (we verify this emperically as well). Therefore, we are motivated to mix distance into cross-entropy loss. Standard cross-entropy (CE) exists as:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{1}$$

where y_i is the ground truth label (one-hot encoded), $\hat{y}_i$ is the predicted probability for class i, and N is the total number of classes. Since y_i being negative (zero) means the term is essentially ignored in the sum, cross-entropy does not "care" about which incorrect class gets which predicted probability — it is "error agnostic". As an example with three classes, if the ground truth is class one, and class zero gets probability p, while class two gets probability q, cross-entropy does not distinguish this scenario from if class zero received a prediction of q, and class two p instead. However, since errors in the classes for our hierarchy are not symmetric, we need to modify the loss such that it is no longer error agnostic. We introduce a distance-weighted cross-entropy (DW-CE) loss:

$$\mathcal{L}_{\text{DW-CE}} = -\sum_{i=1}^{N}\sum_{j=1}^{N} \tilde{d}_{ij} y_i \log(\hat{\delta}_{ij}) \tag{2}$$

$$\hat{\delta_{ij}} = \begin{cases} \hat{y} & \text{if } j = i, \\ 1 - \hat{y} & \text{else.} \end{cases} \tag{3}$$

Here, we introduce the double summation, looping over the number of classes twice. In this manner, for $y_i = 1$, we consider the model's log probability predicted for each class j based on its hierarchical distance from i. Since we wish for the loss to be at a minimum when the predicted probability for $j \neq i$ is small, and when the probability for when $j = i$ is big, we introduce the $\hat{\delta}$ in Equation 3.

One last thing to mention is the reformulation of D. During testing, we experimented with using a distance matrix consisting of the hierarchical distances directly, and adding one to all the values (to ensure the diagonal is computed in our loss as well), and to normalize the distances. We reasoned that while having a diagonal of zeros should still enable learning, however, empirically, we discovered that adding one to the hierarchical distances and normalizing them proved to facilitate more successful and stable training. This adjustment is why we represent hierarchical distance d as $\hat{d}$ in Equation 2.

3.2 Ensemble-Based Uncertainties for Self-Training

Since the FathomNet test set is out-of-distribution (OOD), we reasoned that it would be extremely valuable to be able to exploit the information present. Naturally, we did not have access to ground truth. However, we could bootstrap off of our existing models trained on the training dataset to make predictions to act as pseudo-labels for the next iteration of models. Naturally, this methodology is known in the literature as "self-training" (Amini et al., 2025).

To facilitate successful self-training, we required some means of capturing model confidence, as we believed that $p(y|x;\theta)$ was notoriously overconfident as a result of it reflecting aleatoric (data) uncertainty, rather than epistemic (model) Schweighofer et al. (2025). For our purposes, we employed a gated marginal uncertainty (GMU), developed by Dr. Martin Gillis of HAL Lab [1] defined as:

$$\mu = p(y|x;\Theta), \quad \sigma^2 = \mathrm{Var}[p(y|x;\Theta)] \tag{4}$$

$$\mathrm{GMU} = \frac{\mu_{\max} - \mu_{\max^{(2)}}}{\sigma_{\max} + \sigma_{\max^{(2)}}}; \text{ and} \tag{5}$$

$$U_{\mathrm{GMU}} = \mu_{\max}(1 - e^{-GMU}). \tag{6}$$

Any attempt at capturing variance between plausible models in a prior distribution of models naturally requires more than one model. To this end, we train multiple models on the same learning objective in Equation 2. For determining appropriate pseudo-labels on the OOD data, we employ the mode among our ensemble with uncertainty U_{GMU} serving as the tie-breaker. Depending on how exhaustively we wish to produce pseudo-labels in each iteration, we can set a threshold for our mode, where if an insufficient number of models agree upon a prediction, we do not add it to our training set for the time being.

3.3 Architectures

We experimented with a variety of architectures, including but not limited to CNNs such as ResNets (He et al., 2015) and transformer-based encoders, such as ViTs (Dosovitskiy et al., 2020). We found that for our loss objective, Wide ResNets (Zagoruyko & Komodakis, 2017) performed powerfully, and were trainable at a much faster rate, enabling more iterations for self-training.

4 Negative Results

During the competition, we tried a number of approaches which did not perform as successfully as we'd hoped. We feel, nonetheless, that it may benefit the academic community to report these negative results. In no particular order, these include:

1. **Employing a multi-modal approach, combining full imagery with RoIs.** We trained a classifier by combining global image context with regional object features. Each input was processed by BioCLIP (ViT-B/16) to extract global embeddings, while bounding box

[1] https://hallab.cs.dal.ca/Main_Page

annotations were used to crop regions of interest (ROIs) that were passed through the same backbone to obtain localized features. These global and regional embeddings were then fused using a multi-head attention module (*AttentionFusion*), and the fused representation was classified by a lightweight head trained with cross-entropy loss. Conceptually, this is similar to the approach employed by Yonsei+SSL, who attained first place in the competition. Unlike their approach, we did not incorporate multi-scale features.

2. **Taxonomy-Aware kNN Classification.** We implemented a nearest-neighbor pipeline using ViT embeddings and FAISS indexing Johnson et al. (2019) (linear and IVF). At inference, test embeddings retrieved k nearest neighbors, and their taxonomic trees were aggregated with three protocols: (*i*) hierarchical majority voting, (*ii*) a strict-majority variant, and (*iii*) a flat most-common label. The most-common label returned valid leaf nodes but did not perform as well as expected, while the other two protocols often produced internal nodes, which were not permitted by the competition.

3. **Hierarchical Binary Classification.** We attempted a modified version of Giunchiglia & Lukasiewicz (2020)'s work for hierarchical multi-label classification. Since the original implementation allowed for multi-label classification, which was not applicable for our RoIs. We added a softmax operation over the leaf nodes to compel single-label outputs.

4. **Using Pre-trained Models.** In general, we were surprised to discover that pre-trained encoders on datasets which appeared more similar to FathomNet in domain, such as Lowe et al. (2024) and Stevens et al. (2024), did not produce stronger results than simply using ImageNet. We hypothesize that perhaps this is a result of the RoI data being out of domain for both of these specialized types of pre-training datasets, and a more general dataset may hold advantages in this case.

References

Massih-Reza Amini, Vasilii Feofanov, Loïc Pauletto, Liès Hadjadj, Émilie Devijver, and Yury Maximov. Self-training: A survey. *Neurocomputing*, 616:128904, February 2025. ISSN 0925-2312. doi: 10.1016/j.neucom.2024.128904. URL http://dx.doi.org/10.1016/j.neucom.2024.128904.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. doi: arXiv:2010.11929. URL https://arxiv.org/abs/2010.11929.

eor123, Kevin Barnard, Laura Chrobak, and picekl. Fathomnet 2025 @ cvpr-fgvc. https://kaggle.com/competitions/fathomnet-2025, 2025. Kaggle.

Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent hierarchical multi-label classification networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9662–9673. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6dd4e10e3296fa63738371ec0d5df818-Paper.pdf.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. doi: arXiv:1512.03385. URL http://arxiv.org/abs/1512.03385.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Scott C. Lowe, Benjamin Misiuk, Isaac Xu, Shakhboz Abdulazizov, Amit R. Baroi, Alex C. Bastos, Merlin Best, Vicki Ferrini, Ariell Friedman, Deborah Hart, Ove Hoegh-Guldberg, Daniel Ierodiaconou, Julia Mackin-McLaughlin, Kathryn Markey, Pedro S. Menandro, Jacquomo Monk, Shreya Nemani, John O'Brien, Elizabeth Oh, Luba Y. Reshitnyk, Katleen Robert, Chris M. Roelfsema, Jessica A. Sameoto, Alexandre C. G. Schimel, Jordan A. Thomson, Brittany R. Wilson, Melisa C.

Wong, Craig J. Brown, and Thomas Trappenberg. Benthicnet: A global compilation of seafloor images for deep learning applications. *arXiv*, 2024. doi: https://doi.org/10.1038/s41597-025-04491-1. URL https://arxiv.org/abs/2405.05241.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. On information-theoretic measures of predictive uncertainty, 2025. URL https://arxiv.org/abs/2410.10786.

Samuel Stevens, Jiaman Wu, Matthew J Thompson, Elizabeth G Campolongo, Chan Hee Song, David Edward Carlyn, Li Dong, Wasila M Dahdul, Charles Stewart, Tanya Berger-Wolf, Wei-Lun Chao, and Yu Su. Bioclip: A vision foundation model for the tree of life, 2024. URL https://arxiv.org/abs/2311.18803.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv*, 2017. doi: arXiv:1605.07146. URL https://arxiv.org/abs/1605.07146.