Improving Detection of Rare Nodes in Hierarchical Multi-Label Learning

Anonymous authors
Paper under double-blind review

Abstract

In hierarchical multi-label classification, a persistent challenge is enabling model predictions to reach deeper levels of the hierarchy for more detailed or fine-grained classifications. This difficulty partly arises from the natural rarity of certain classes (or hierarchical nodes) and the hierarchical constraint that ensures child nodes are almost always less frequent than their parents. To address this, we propose a weighted loss objective for neural networks that combines node-wise imbalance weighting with focal weighting components, the latter leveraging contemporary quantification of ensemble uncertainties. By emphasizing rare nodes rather than rare observations (data points), and focusing on uncertain nodes for each model output distribution during training, we observe improvements in recall by up to a factor of five on benchmark datasets, along with statistically significant gains in F1 score. We also provide evidence that the approach can benefit convolutional neural networks in cases where the available vision encoder has not been fully prepared for the downstream task.

1 Introduction

In observing the natural world, contemplating the "relatedness" of objects can aid in understanding their properties. Hierarchies are therefore ubiquitous in our understanding of the world. Furthermore, as machine learning techniques expand to other fields such as the biological sciences, there is growing interest in training hierarchical models capable of annotating in ways useful to experts (Romero et al., 2022; Hunt & Steenstrup Pedersen, 2023; Blondin et al., 2024; Lowe et al., 2025). For this work, we describe broader higher level categories in the hierarchy as "ancestor" or "parent" nodes, while the lower level and more detailed categories stemming from these are referred to as "descendant" or "child" nodes. More generally, hierarchies can be understood as directed acyclic graphs (DAGs), and may therefore be tree-like (where nodes only possess a singular parent) or otherwise (where they can be descended from multiple parents). Finally, there are almost always potential hierarchical associations inherent in data, even if not annotated in this manner.

Another property, the multi-label characteristic of the data, may also appear in datasets. Multi-label refers to the possibility of multiple classifications existing for a single observation. A contemporary challenge for machine learning is the need for big data and the lack of a singular large dataset in fields traditionally accustomed to working with small hand-annotated datasets without standardization (Kuznetsova et al., 2020; Lowe et al., 2025). As researchers work to amalgamate these smaller and potentially multi-label datasets, they may be forced to either exclude these datasets or, if additional processing to separate the annotations is not available, accept the amalgamated dataset as also multi-label. Therefore, it seems reasonable that more multi-label datasets will emerge as the number of amalgamated datasets grows in various academic fields.

Combining hierarchical and multi-label properties transforms the data modelling process into a hierarchical multi-label (HML) task. Within HML, the classification task may be further divided into full depth (HML-FD) and partial depth (HML-PD). Full-depth refers to datasets where each observation is annotated to the maximum depth permitted by the hierarchy. Whereas, partial-depth permits annotations terminating at higher-level categories, prior to reaching the leaf nodes.

Addressing imbalance in the HML context, in order to attain deeper and more comprehensive predictions, remains a relatively under-explored area. Existing works such as Pereira et al. (2018; 2021) apply a resampling

approach rather than a weighted loss methodology. This preference may be a result of the relationship between HML and pure multi-label problems, and the popularity of resampling methods in multi-label imbalance literature (Tarekegn et al., 2021). For HML, we observe that an initial naive approach, adapting the multi-class classification style of weighting, may not be generally viable. Additionally, any observation-based approach in HML, whether resampling-based or weighted, risks the issues of oversampling common nodes or undersampling rare ones, due to the conjoined nature of annotations. Instead, we propose looking at the problem through a node-based weighting process, where the emphasis on nodes is independent of the distribution of individual observations.

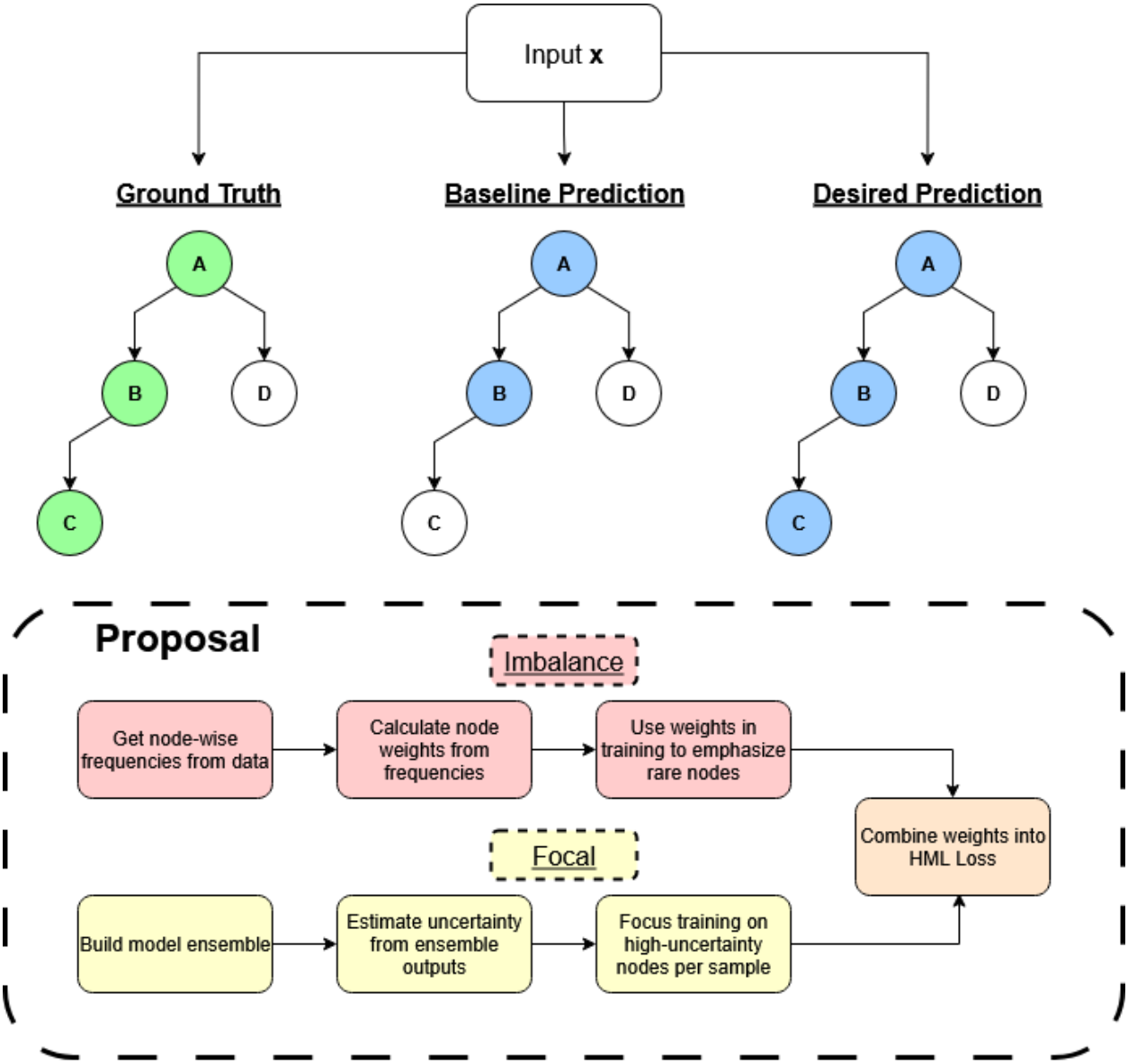


Figure 1: **Overview of HML Rare Node Problem and Proposed Weighted Approach.** The proposed approach towards detecting rare descendant nodes in HML problems consists of two branches. The imbalance branch focuses on a node-wise weighting system independent of sample frequency, while the focal branch employs modern measures of uncertainty in determining challenging nodes for the model ensemble.

Finally, we look to further enhance this effect by utilizing the concept of model uncertainty in a manner inspired by focal loss (Lin et al., 2020). We propose to emphasize nodes characterized by increased uncertainty during training. To this end, we study and apply various ways of reliably quantifying uncertainty in modern literature (Kendall & Gal, 2017; Schweighofer et al., 2023; Gillis et al., 2025b).

Our core problem statement is to improve the detection rates of underrepresented classes in HML datasets by machine learning models. Our contributions include:

1. **Imbalance weighting.** Examination of the different ways imbalance weighting may be defined and implemented to improve hierarchical classification. Evidence is provided for which techniques are effective at fulfilling the objectives in the problem statement.

2. **Focal weighting.** The concept of uncertainty, as defined in contemporary literature, is used to augment a focal term, adapted for the HML context. Evidence is provided that incorporating this uncertainty term is beneficial for improving rare node detection.

An overview of the problem and our proposed approach is available in Figure 1.

2 Background

2.1 Learning with Hierarchical Constraint

One of the first requirements in modelling HML data is satisfying the hierarchical constraint — that is, the probability for the parent's presence in an observation must be greater than or equal to the probability of any descendants. Within the literature, there exists a number of ways to address this matter (Wehrmann et al., 2018; Giunchiglia & Lukasiewicz, 2020). For our work, we apply Coherent Hierarchical Multi-Label Classification Neural Network (C-HMCNN) (Giunchiglia & Lukasiewicz, 2020), which remains a robust and competitive method, as a framework to demonstrate our experiments. The essence of the approach is in translating the hierarchical information into an adjacency matrix of size $n \times n$, where n is the number of nodes in the hierarchy. Each element in this matrix $\mathbf{A}$ is then defined as:

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } j \in S_i, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Where S_i is the set of hierarchical descendants for node i, and by definition, node i is a descendant of itself. Each row i in the matrix $\mathbf{A}$ then acts as a filter for an expanded model output, which turns off all the predictions that are outside of node i's direct descendant line. Finally, a max operation taken along the rows ensures that each prediction for a node is the $\max(p(\text{node}_i),\ p(\text{node}_j))$, thereby satisfying the hierarchical constraint. The loss, referred to in Giunchiglia & Lukasiewicz (2020) as "max constraint loss" ($\mathcal{L}_{\text{MC}}$), involves applying this constraint mechanism in the fashion below, depicted with tensor notation for clarity.

We define the following variables:

θ	the model;
$\mathbf{R}$	expanded (to match batch size) adjacency tensor representing hierarchy;
f_{CM}	a function that enforces the hierarchical constraint, given probabilities and $\mathbf{R}$;
$\mathbf{X}$	a tensor of input features for a batch;
$\mathbf{Y}$	a tensor containing ground truth annotations for a batch; and
BCE	the computation for binary cross-entropy.

$$\widetilde{\mathbf{Y}}_{\mathrm{A}} = f_{\mathrm{CM}}\big(\theta(\mathbf{X}), \mathbf{R}\big) \tag{2}$$

$$\widetilde{\mathbf{Y}}_{\mathrm{B}} = f_{\mathrm{CM}}\big(\mathbf{Y}\theta(\mathbf{X}), \mathbf{R}\big) \tag{3}$$

$$\widetilde{\mathbf{Y}} \quad = (1-\mathbf{Y})\widetilde{\mathbf{Y}}_{\mathrm{A}} + \widetilde{\mathbf{Y}}_{\mathrm{B}} \tag{4}$$

$$\mathcal{L}_{\mathrm{MC}} = \mathrm{BCE}(\widetilde{\mathbf{Y}}, \mathbf{Y}) \tag{5}$$

One particularity in this process is the asymmetrical treatment of model predictions for positively (nodes that are labelled as present in an observation) and negatively (nodes that are not present in an observation) annotated nodes. If we interpret $\widetilde{\mathbf{Y}}_{\mathrm{A}}$ as the predicted component for negatively annotated nodes, we see that it is filtered in Equation 4, whereas the positive predicted component, $\widetilde{\mathbf{Y}}_{\mathrm{B}}$, is filtered in Equation 3. This asymmetric treatment accounts for the possibility of model predictions for negatively annotated descendant nodes that happen to have positively annotated parents. In this case, it would be undesirable for the positive parent to have its prediction influenced by that of the negative child, which is why it is filtered prior to applying the constraint function in Equation 3. The recombined predicted probabilities (Equation 4) are then applied to calculate binary cross-entropy (BCE) loss (Equation 5). In Giunchiglia & Lukasiewicz (2020), an explicit example is presented for attempting HML directly with BCE and without this constraint process, showing that gradients can point in an incorrect direction for model predictions which violate hierarchy. One last point to note is that $\widetilde{\mathbf{Y}}$ is used only for training. For inference, $\widetilde{\mathbf{Y}}_{\mathrm{A}}$ from Equation 2 is used directly to form predictions.

3 Methods

Our proposal to better detect rare nodes in HML observations involves augmenting the hierarchical loss with two components: a class imbalance weighting and a focal weighting, inspired by focal loss (Lin et al., 2020). We define:

f	vector of occurrence frequencies for each node within a dataset;
$\widetilde{\mathbf{W}}_0$	minimum allowed weight (information gate) for imbalance weighting;
$\widetilde{\mathbf{W}}(\mathbf{Y}, f)$	set of weights accounting for HML imbalance;
$\mathbf{U}_0$	minimum allowed information for uncertainty gating;
$\mathbf{U}(\Theta(\mathbf{X}))$	set of weights attained from ensemble of models (Θ) based uncertainties; and
k	exponential factor to emphasize nodes with extreme uncertainties.

$$\mathcal{L}_{\mathrm{focal}} = \underbrace{\left(\widetilde{\mathbf{W}}_0 + \widetilde{\mathbf{W}}(\mathbf{Y}, f)\right)}_{\text{imbalance weighting}} \underbrace{\left(\mathbf{U}_0 + \mathbf{U}\big(\Theta(\mathbf{X})\big)^k\right)}_{\text{focal weighting}} \mathcal{L}_{\mathrm{MC}} \tag{6}$$

3.1 Imbalance Weighting

In the HML context, imbalance weighting generated by looking at the multi-label annotations on an observation-by-observation basis risks overemphasizing the common component(s) of an annotation while attempting to address its rare component(s). This problem is exacerbated by the hierarchical nature, where it is typical for parent nodes to be common, while their child nodes are rare. Therefore, we propose conducting imbalance weighting on a node-wise basis, with each node possessing its own weight(s).

The question is, therefore, what should these node weights be? If we approach the problem naively with a standard calculation common in mutually exclusive multi-class classification:

$$w_i = \frac{N_{\text{obs}}}{N_{\text{classes}} \cdot n_i} = (N_{\text{classes}} f_i)^{-1}, \tag{7}$$

where n_i refers to the number of occurrences of a class[1], N_{obs} refers to the number of data points in a dataset, and N_{classes} refers to the number of classes, we find that firstly, there is ambiguity regarding what N_{classes} should be. Since we work with binary representation in HML, there is an argument N_{classes} should be two, reflecting whether or not a node is annotated as being present in an observation. Alternatively, N_{classes} can be equivalent to the number of nodes present in the hierarchy. In our preliminary investigations, we observe that setting N_{classes} to the number of nodes appears to deliver stronger model performance. However, both approaches tend to result in common nodes being assigned extremely low weights, which is adversarial to learning, despite large weights for rare nodes. Consequently, we opt to rescale our weights to a floor of zero, and introduce a minimum term $\tilde{w}_0$ afterwards, to ensure a minimum amount of information is passed from the annotation during training for common nodes:

$$\tilde{w}_i = \tilde{w}_0 + w_i \frac{w_i - w_{min}}{w_{max} - w_{min}}. \tag{8}$$

Another line of inquiry is whether nodes should share a weight for both positive and negative annotations. Since different information is communicated by the presence and absence of a node and since the number of positive annotations is inversely related to the number of negatives, the weight should then depend on the annotations themselves. We observe that regardless of different formulations for the negative weightings, we were unable to achieve strong performance. Therefore, our node weights are defined as:

$$\tilde{w}_i(y_i) = \begin{cases} \tilde{w}_i & \text{if } y_i = 1, \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

Hence, in the abbreviated tensor notation in Equation 6, the imbalance component contains the term $\widetilde{\mathbf{W}}$ as a function of $\mathbf{Y}$ and f.

3.2 Focal Weighting

Focal loss was originally developed for image segmentation, where pixel annotations are often strongly imbalanced between pixels of interest and background. To address these conditions, Lin et al. (2020) introduced a term $(1 - p(y|x;\theta))^k$ weighting the loss, which determines the amount of weight given to an annotation based on the model's own confidence — interpreted as $p(y|x;\theta)$. Over time, it is expected that the model's growing confidence would prompt the training process to ignore the common background pixels and instead focus on the rarer annotations. The degree to which extremes are emphasized or ignored is then modulated by the exponential parameter k.

As in image segmentation, HML tends to possess accurate (in terms of correctness) but rare annotations for nodes deep within a label hierarchy, yielding a problem we believe can benefit from a focal-style weighting. However, we also introduce improvements motivated by developments in model uncertainty literature. We define uncertainty as $(1 - c)$, where c is confidence. Furthermore, we note that $p(y|x;\theta)$ is not necessarily an effective measure of model confidence. There exists in the literature a significant number of works that have studied the unreliability of this naive measure of confidence (Hendrycks et al., 2019; Costa et al., 2024). We first begin with the creation of an ensemble of models, Θ, to attempt to capture the distribution of models, $p(\theta|\mathbf{X}, \mathbf{Y})$, for a classification task, with μ being the Bayesian Model Average (BMA) for this ensemble:

$$\mu = \hat{p}(y|x;\Theta), \quad \sigma^2 = \text{Var}[\hat{p}(y|x;\Theta)]; \tag{10}$$

$$\mu_{\max} = \max(\mu, 1 - \mu); \text{ and} \tag{11}$$

[1]In our work, we set n_i to refer to the number of occurrences of a node and all of its descendants (i.e., the aggregate count).

$$\hat{\mu}_{\max} = 2\left(\mu_{\max} - 0.5\right), \quad U_{\text{bBMA}} = 1 - \hat{\mu}_{\max}. \tag{12}$$

By taking the BMA of Θ, we hope to apply a confidence and uncertainty metric which eliminates some of the individual noise stemming from trained models, thereby making it more robust. An important point is the use of $\hat{\mu}_{\max}$ rather than μ directly, as smaller predicted probabilities in binary representation do not indicate uncertain predictions in classes, but rather certain negative predictions. We refer to this measure as the binary Bayesian Model Average (bBMA), with its counterpart being the uncertainty associated with this confidence as our first focal term candidate.

Our second candidate is a signal-to-noise ratio (SNR) based metric, referred to as the Gated Margin Uncertainty (GMU) (Gillis et al., 2025b), intended to capture variance across model predictions:

$$\mu_{\max^{(2)}} = \min\left(\mu, 1 - \mu\right); \tag{13}$$

$$\text{SNR} = \frac{\mu_{\max} - \mu_{\max^{(2)}}}{\sigma_{\max} + \sigma_{\max^{(2)}}}; \text{ and} \tag{14}$$

$$U_{\text{GMU}} = 1 - \left(\hat{\mu}_{\max}(1 - e^{-\text{SNR}})\right). \tag{15}$$

In Equation 13, we use $\mu_{\max^{(2)}}$ in the more general definition of the metric, where it is the probability of the second largest predicted class, which in the binary case is the minimum. The interpretation of GMU in this context is that it captures a notion of how much the models in Θ agree with each other. Scaling this quantity by $\hat{\mu}_{\max}$ then incorporates a notion of "confidence" to the measure. This mixed quantity behaves in a manner where it is large (close to one), given high certitude and agreement between the models, and is small (close to zero), given low certitude or little agreement between the models.

More generally, in the literature, uncertainty is typically decomposed into two forms: aleatoric (data) and epistemic (model) (Kendall & Gal, 2017; Wimmer et al., 2023; Schweighofer et al., 2023). Aleatoric refers to the uncertainty inherent in the data resulting from the collection method. Consequently, aleatoric uncertainty cannot be reduced with the collection of more data. Therefore, it should, in theory, be pointless to focus on nodes with high aleatoric uncertainty. Instead, epistemic uncertainty — the uncertainty of the true model generating the data phenomenon — should be reducible with the acquisition of more varied data or perhaps a review and focus on rare pieces of existing data.

In a recent work, Schweighofer et al. (2023), this model uncertainty $U_{\text{epistemic}}$ is defined (in our terms) as:

$$U_{\text{epistemic}} = \mathbb{E}_{p(\theta|\mathbf{X})}\left[\mathbb{E}_{p(\tilde{\theta}|\mathbf{X})}\left[D_{\text{KL}}\big(p(y|x;\theta) \,\|\, p(y|x;\tilde{\theta})\big)\right]\right]. \tag{16}$$

For our work, we interpret this as a matrix comparing the Kullback–Leibler (KL) divergence for the probabilistic output of every model in a set, compared to that of every other model in this set, with the expectation taken across the matrix dimensions:

$$U_{\text{epistemic}} = \frac{1}{N(N-1)} \sum_{\theta=1}^{N} \sum_{\tilde{\theta}=1}^{N} D_{\text{KL}}\big(p(y|x;\theta) \,\|\, p(y|x;\tilde{\theta})\big); \tag{17}$$

with N referring to the number of models in our set, or ensemble size.

In addition to this formulation of epistemic uncertainty, we experiment with a variant of it where the KL-divergence is switched for the Jensen-Shannon divergence, as a measure bounded by one can perhaps add interpretability and may perform better when scaling with other quantities in the loss.

As before, we define a minimum weight in conjunction with the uncertainty term due to empirically observed benefits to model learning, leading to the focal component in our loss as defined in Equation 6. Lastly, the focal

weighting is calculated without gradients, as existing works suggest that training based on ensemble-derived quantities with gradients can more easily lead to ensemble collapse (Lee et al., 2015).

3.3 Metrics

Micro-averaged precision (AP) score is commonly used to evaluate HML classification performance. This metric is the rectangular approximation of the area under the precision-recall curve (AUPRC). Its popularity is inherited from multi-label research, in which thresholding model outputs can be challenging (Bi & Kwok, 2011). While it can be informative, it tends to be overly optimistic, especially in sparsely annotated data. Furthermore, the AP score evaluates the dataset as a whole and cannot provide insightful information on the model's performance at a node level. For practical purposes, we also value assessment of the model's actual predictions on data (not just probabilities), which cannot be easily attained without thresholding.

As a result of these motivations, we include the node-wise F1, precision, recall, and binarized average precision score (Bin. AP) as part of the model evaluation. The F1, precision, and recall are calculated for each data point using the true positives, false positives, and false negatives tabulated for each node. The Bin. AP score is simply the AP score evaluated on model predictions thresholded at 0.5, taking ≥ 0.5 as a positive prediction for a node, and < 0.5 as a negative prediction.

4 Related Works

Although our work focuses specifically on addressing HML imbalance, we consider it to be derivative of addressing multi-label imbalance more generally. Broadly speaking, approaches in this setting can be divided into four groups: resampling, classifier adaptation, ensemble approaches, and cost-sensitive algorithms (Tarekegn et al., 2021). Resampling algorithms typically look to determine observations that contain either rare or common annotations and over- or under-sample them, respectively (Charte et al., 2015a;b). Classifier adaptation approaches consist of dedicated algorithms designed to train the model to learn the imbalance prior. These approaches are varied and include examples such as Chen et al. (2006), where multi-label problems are broken down into two-class subproblems and then recombined for inference. Zhang et al. (2022) propose an ensemble approach, in which correlations between class annotations are exploited by converting multi-label binary datasets into a trinary classification task, predicting the presence of one class in conjunction with another. The final output is then based upon the aggregation of the confidences from the binary and multi-class models. Finally, as an example of a cost-sensitive approach, Daniels & Metaxas (2017) employ Hellinger distance (Hellinger, 1909), as an objective for random forests. Among these families of approaches, resampling remains the most popular (Tarekegn et al., 2021).

Reflecting the more general multi-label trend, the most popular existing approaches to addressing HML imbalance appear to be resampling-based. These have focused on either converting HML to a multi-label problem (Pereira et al., 2018) or implementing resampling schemes such as hierarchical random oversampling (HROS-PD) (Pereira et al., 2021). Pereira et al. (2018) convert HML to a multi-label problem by treating each node within a hierarchy as a multi-label class. A multi-label resampling approach is then applied. They compare popular multi-label imbalance methods such as LPROS/US (Charte et al., 2015a), MLROS/US (Charte et al., 2015b), multi-label SMOTE (Charte et al., 2015c), and MLeNN (Liu et al., 2022), concluding that LPROS held the strongest performance.

In Pereira et al. (2021), the team proposes an approach that determines the imbalance of the hierarchical nodes and then resamples them to an acceptable threshold. In their work, it was discovered that the oversampling approach for rare nodes (HROS-PD) outperforms the undersampling approach (HRUS-PD). However, the method is limited in two ways. First, it was designed to handle tree-like hierarchies and therefore would not perform optimally for directed acyclic graph (DAG) hierarchies (as we would expect and demonstrate on our GO datasets). Furthermore, oversampling annotations conjoined with common nodes risks further oversampling already abundant nodes.

In our work, we include the HML to multi-label/LPROS method and HROS-PD as benchmarks. Following the algorithm as described in (Pereira et al., 2021), we implement the approach from scratch and include it with our code. By disconnecting the weighting and confidence for each node from their observations, we

also address the issues of working in both a tree-like or DAG-like hierarchical environment, as well as avoid overemphasizing already abundant nodes.

5 Experiments

Our experiments are presented in four sections. First, we introduce the datasets we will be working with. Next, we examine the impact of applying imbalance weighting alone as a solution on imbalanced gene product data. Then, we study the effects of incorporating focal weighting on these same datasets. Finally, we compare the effects of these approaches on benthic imagery data. When applying focal weighting on gene product data, we employ an ensemble of models to evaluate uncertainty. For imagery data, we use a common encoder and create an ensemble of output heads trained on the same classification task to achieve our distribution of models. This partial ensemble is a common regularization strategy (Lee et al., 2015; Gillis et al., 2025a), although we are specifically applying it to generate quantifiable model uncertainty. The hyperparameters we use for our experiments are presented in Appendix A, and are also available in our code.

5.1 Datasets

We evaluate on 16 gene product datasets, evenly divided into two sets, with each set annotated under two distinct classification frameworks: Functional Catalogue (FUN) (Ruepp et al., 2004) and Gene Ontology (GO)[2] (Ashburner et al., 2000). These two schemes were developed to catalogue gene products and proteins, and in the case of GO, the genes themselves as well. These datasets are popular benchmarks in HML works (Bi & Kwok, 2011; Giunchiglia & Lukasiewicz, 2020; Tarekegn et al., 2021; Pereira et al., 2018; 2021).

There are two key differences between FUN and GO. First, FUN contains hierarchical structures up to a depth of six (we count root node(s) as depth zero), with datasets typically containing 500 nodes, except for Eisen (FUN), which uses 462. GO contains over 4,000 nodes, with Eisen (GO) using 3,574 nodes. Nodes for GO are spread across 14 hierarchical levels. Furthermore, while FUN is tree-like, GO is a DAG.

We also make use of a subset of BenthicNet (Lowe et al., 2025), an image dataset cataloguing benthic habitats. The images we use are of Echinodermata, following the echinoderm branch of the Collaborative and Automated Tools for Analysis of Marine Imagery (CATAMI) biota hierarchical tree (Althaus et al., 2015) — a classification scheme developed for the visual recognition of underwater biota and seafloor characteristics. The hereby dubbed "BenthicNet-E" contains 13 nodes, up to a depth of two, making it a comparatively simpler structure than FUN or GO. The BenthicNet-E dataset contains a total of 5,077 training data points and 1,650 test data points, inherited from the original full BenthicNet partitions, which considered geospatial relations between observations (Lowe et al., 2025). The size and focus of BenthicNet-E is typical for a small locally sourced dataset in the ocean sciences (Brown et al., 2012; Zhou et al., 2023), and is therefore similar in scope to popular problems in the field.

5.2 Imbalance Weighting on Gene Product Data

Our results for the FUN category of datasets are presented in Table 1. Since they show similar trends, we reserve our results for the GO datasets in Appendix C.

These observations suggest that the Bin. AP score, F1, and especially recall scores benefit significantly from node-wise weighting, with recall improving by a factor of five over the control, on certain datasets using $\tilde{w}_0 = 0.25$. However, it appears that in a few of these datasets, the gain in recall comes with a possible trade-off in precision — although the difference is often within two standard deviations (σ). Additionally, in some cases, imbalance weighting may even improve precision (as seems to be the case for Derisi, Gasch-2, and SPO).

In Figure 2, we note the behaviour of precision and recall on the top-20 most frequent nodes in Derisi with varying $\tilde{w}_0$. Firstly, we observe that there appears to be a trend of decreasing precision as we decrease $\tilde{w}_0$. Gains in precision appear to be due to the detection of nodes which were previously being ignored at higher

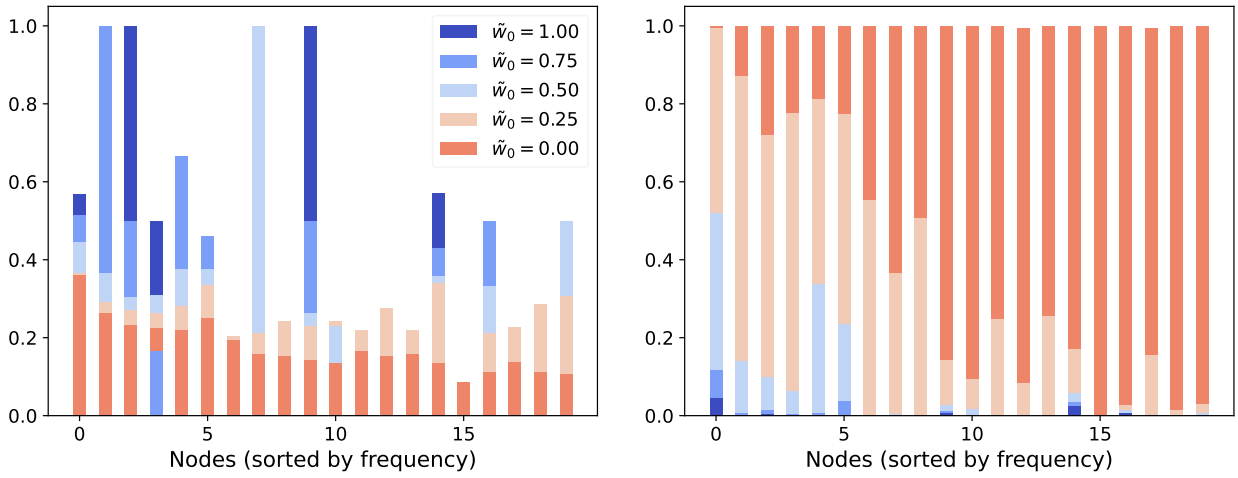
[2]An introduction to GO is available at `https://geneontology.org/`.

Table 1: **HML Method Comparisons (FUN)**. Different strategies in addressing HML imbalance are compared for the FUN datasets: no method used (None), HROS-PD, and node weighting using $\tilde{w}_0$ values of 0.25 and 0.50. Means that are significantly ($> 2\sigma$) greater than all others for a dataset bolded.

Dataset (FUN)	Method	F1	Precision	Recall	Bin. AP	AP
CELLCYCLE	None	2.61 ± 0.09	9.81 ± 0.70	1.74 ± 0.06	6.26 ± 0.11	25.68 ± 0.13
	LPROS	2.98 ± 0.18	9.70 ± 0.80	2.13 ± 0.19	6.42 ± 0.10	24.81 ± 0.25
	HROS-PD	2.70 ± 0.25	9.57 ± 0.90	1.88 ± 0.22	6.18 ± 0.12	25.57 ± 0.16
	$\tilde{w}_0 = 0.25$	**4.85** ± 0.11	9.13 ± 0.36	**4.59** ± 0.11	**10.21** ± 0.08	25.45 ± 0.10
	$\tilde{w}_0 = 0.50$	3.54 ± 0.08	9.34 ± 0.36	2.68 ± 0.07	8.20 ± 0.08	25.69 ± 0.12
DERISI	None	0.46 ± 0.03	1.75 ± 0.21	0.34 ± 0.03	2.67 ± 0.08	19.58 ± 0.10
	LPROS	0.50 ± 0.02	1.96 ± 0.16	0.35 ± 0.02	2.66 ± 0.08	19.18 ± 0.10
	HROS-PD	0.44 ± 0.02	1.82 ± 0.41	0.33 ± 0.01	2.70 ± 0.09	19.39 ± 0.11
	$\tilde{w}_0 = 0.25$	**1.82** ± 0.05	**2.88** ± 0.20	**2.16** ± 0.07	**7.55** ± 0.07	19.61 ± 0.09
	$\tilde{w}_0 = 0.50$	0.90 ± 0.06	2.31 ± 0.33	0.74 ± 0.04	4.34 ± 0.08	19.66 ± 0.09
EISEN	None	4.82 ± 0.25	14.03 ± 0.62	3.29 ± 0.20	9.36 ± 0.14	30.70 ± 0.07
	LPROS	5.87 ± 0.32	15.15 ± 0.78	4.11 ± 0.30	9.32 ± 0.10	29.69 ± 0.14
	HROS-PD	4.58 ± 0.25	13.81 ± 0.62	3.09 ± 0.27	8.43 ± 0.15	30.19 ± 0.07
	$\tilde{w}_0 = 0.25$	**7.16** ± 0.16	11.27 ± 0.37	**6.69** ± 0.12	**12.64** ± 0.10	30.51 ± 0.11
	$\tilde{w}_0 = 0.50$	5.92 ± 0.21	12.79 ± 0.54	4.55 ± 0.15	10.91 ± 0.17	30.73 ± 0.07
EXPR	None	5.84 ± 0.26	13.49 ± 0.19	4.26 ± 0.24	10.11 ± 0.12	30.13 ± 0.23
	LPROS	7.16 ± 0.26	14.84 ± 0.73	5.40 ± 0.22	9.90 ± 0.17	28.45 ± 0.13
	HROS-PD	6.31 ± 0.40	14.17 ± 0.44	4.71 ± 0.44	10.18 ± 0.15	30.03 ± 0.22
	$\tilde{w}_0 = 0.25$	**8.50** ± 0.30	11.92 ± 0.35	**8.10** ± 0.42	**12.50** ± 0.12	30.30 ± 0.09
	$\tilde{w}_0 = 0.50$	6.95 ± 0.33	12.86 ± 0.46	5.59 ± 0.34	11.47 ± 0.07	30.40 ± 0.09
GASCH-1	None	4.98 ± 0.52	12.56 ± 1.57	3.64 ± 0.37	8.27 ± 0.16	28.44 ± 0.24
	LPROS	5.64 ± 0.24	13.10 ± 0.83	4.16 ± 0.16	8.15 ± 0.27	26.71 ± 0.16
	HROS-PD	5.63 ± 0.16	13.36 ± 0.48	4.13 ± 0.10	8.27 ± 0.17	28.28 ± 0.13
	$\tilde{w}_0 = 0.25$	**7.42** ± 0.36	11.08 ± 0.50	**6.94** ± 0.36	**11.64** ± 0.20	28.57 ± 0.22
	$\tilde{w}_0 = 0.50$	6.03 ± 0.36	12.19 ± 1.02	4.77 ± 0.29	10.19 ± 0.21	28.58 ± 0.22
GASCH-2	None	2.61 ± 0.12	8.72 ± 0.61	1.80 ± 0.08	5.47 ± 0.13	25.83 ± 0.06
	LPROS	2.86 ± 0.10	9.92 ± 0.50	1.98 ± 0.09	5.49 ± 0.15	24.97 ± 0.10
	HROS-PD	2.58 ± 0.10	8.55 ± 0.22	1.79 ± 0.06	5.38 ± 0.13	25.54 ± 0.14
	$\tilde{w}_0 = 0.25$	**4.92** ± 0.16	9.19 ± 0.33	**4.62** ± 0.09	**10.24** ± 0.08	25.51 ± 0.06
	$\tilde{w}_0 = 0.50$	3.47 ± 0.12	9.21 ± 0.59	2.63 ± 0.11	7.66 ± 0.20	25.76 ± 0.06
SEQ	None	4.14 ± 0.19	11.85 ± 0.69	2.91 ± 0.15	9.95 ± 0.14	29.24 ± 0.15
	LPROS	5.38 ± 0.16	**14.31** ± 0.44	3.84 ± 0.12	10.44 ± 0.16	28.56 ± 0.27
	HROS-PD	4.78 ± 0.20	13.29 ± 0.36	3.43 ± 0.18	10.29 ± 0.11	29.40 ± 0.26
	$\tilde{w}_0 = 0.25$	**6.60** ± 0.12	10.57 ± 0.61	**6.29** ± 0.20	**12.13** ± 0.15	28.91 ± 0.20
	$\tilde{w}_0 = 0.50$	5.29 ± 0.11	11.83 ± 0.63	4.18 ± 0.15	11.12 ± 0.20	29.17 ± 0.16
SPO	None	0.61 ± 0.04	3.03 ± 0.17	0.43 ± 0.02	3.39 ± 0.11	21.54 ± 0.07
	LPROS	0.80 ± 0.04	3.20 ± 0.18	0.54 ± 0.03	3.53 ± 0.08	21.17 ± 0.12
	HROS-PD	0.60 ± 0.04	3.09 ± 0.19	0.42 ± 0.02	3.37 ± 0.10	21.38 ± 0.06
	$\tilde{w}_0 = 0.25$	**1.94** ± 0.04	3.21 ± 0.05	**2.17** ± 0.02	**7.82** ± 0.05	21.21 ± 0.09
	$\tilde{w}_0 = 0.50$	1.11 ± 0.04	2.98 ± 0.11	0.87 ± 0.03	4.84 ± 0.07	21.42 ± 0.08

$\tilde{w}_0$ values. Furthermore, we see that recall improves dramatically with almost all presented nodes, being perfectly recalled at $\tilde{w}_0 = 0$. Nonetheless, we find that if we lower $\tilde{w}_0$ to this degree, too much precision is lost, resulting in a decrease in the overall F1 score. Empirically, we find $\tilde{w}_0 = 0.25$ to be a reasonable value in boosting overall F1.

This observed phenomenon is perhaps counterintuitive. It may seem surprising that smaller weights (< 1) on positive instances for majority nodes should improve recall and lower precision. Generally, we would expect that assigning a lower priority to positive annotations for a node typically reduces the model's tendency to predict positively, except in the most certain cases, thereby decreasing recall and increasing precision.



(a) Precision

(b) Recall

Figure 2: **Comparison of the Effects of Varying $\tilde{w}_0$ on Derisi (FUN).** Plotted are node-wise precision and recall scores across the top-20 most frequent nodes in their respective datasets.

We reason that due to the hierarchical constraint, the model may learn to classify majority nodes indirectly through their descendant nodes. Subsequently, because it does not learn these common stem nodes directly, its recall increases at the cost of precision. The more we lower $\tilde{w}_0$, the more we encourage indirect node learning in this fashion.

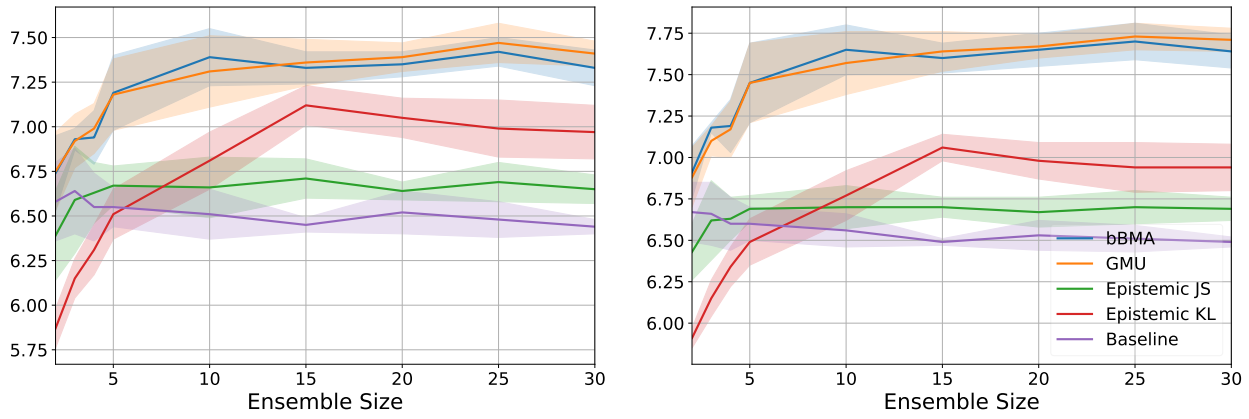
In Appendix D, in order to preserve precision, we attempt to mix a weighted loss objective with an unweighted one by implementing a scheduler that transforms the weighted objective to unweighted over the course of training. Additionally, we explore the possibility of mixing the two objectives through linear interpolation. We find that these approaches were mostly unsuccessful in retaining strong F1 while preserving precision. However, in Appendix E, we combine LPROS with weighted loss, and find significant improvements in preserving precision and thus inducing further boosts to the F1 score on most gene product datasets.

5.3 Focal Weighting on Gene Product Data

We begin by observing the trends of model performance metrics as we increase the number of models in our ensemble. For these experiments, as a baseline, we use the imbalance weighted loss, with $\tilde{w}_0 = 0.25$ and all applications of focal weighting are used in tandem with this imbalance weighting. Additionally, predictions for ensembles are made by thresholding the mean of the outputs.

For F1, we find that GMU and bBMA appear to perform well at augmenting recall (although all focal measures appear to outperform the baseline). Consequently, the higher recall and a relatively unchanged precision boost F1, as can be seen in Figure 3a. As one might expect, we observe that the focal loss candidates perform better as the number of heads increases, suggesting that the ensemble becomes a more reliable representation of the distribution of models as the ensemble size grows. This observation is in contrast to the baseline without focal weighting, which remains relatively independent of ensemble size.

It is nonetheless surprising that epistemic uncertainty did not perform as strongly as anticipated. One possible explanation is that when considering total uncertainty (Wimmer et al., 2023) as the sum total of aleatoric and epistemic components, aleatoric uncertainty is almost always the dominant term with sufficient training. The smaller epistemic uncertainty may explain its limited positive influence. It would be informative to explore whether model calibration (Wimmer et al., 2023; Pavlovic, 2025) enables more robust estimates of uncertainty. Another plausible explanation may be that there exist benefits for models to focus not just on the epistemic component but also the aleatoric, if the model is not learning a generalizable approach, but instead, a memory-reliant means of solving an in-domain classification task.



(a) F1

(b) Recall

Figure 3: **Focal Weighting with Increasing Ensemble Size on Cellcycle (FUN).** For both displayed measures, the bold central line indicates the mean of the uncertainty term used for focal weighting, while the blurred outline indicates a range of one σ.

Hereafter, we implement a static ensemble size of ten to evaluate different kinds of uncertainties for the full catalogue of gene product datasets. We confirm that the same trends we observe in Figure 3, hold true for these other gene product datasets in both FUN and GO. Our results for F1 on FUN are presented in Table 2. A full set of metrics are presented in Appendix B.

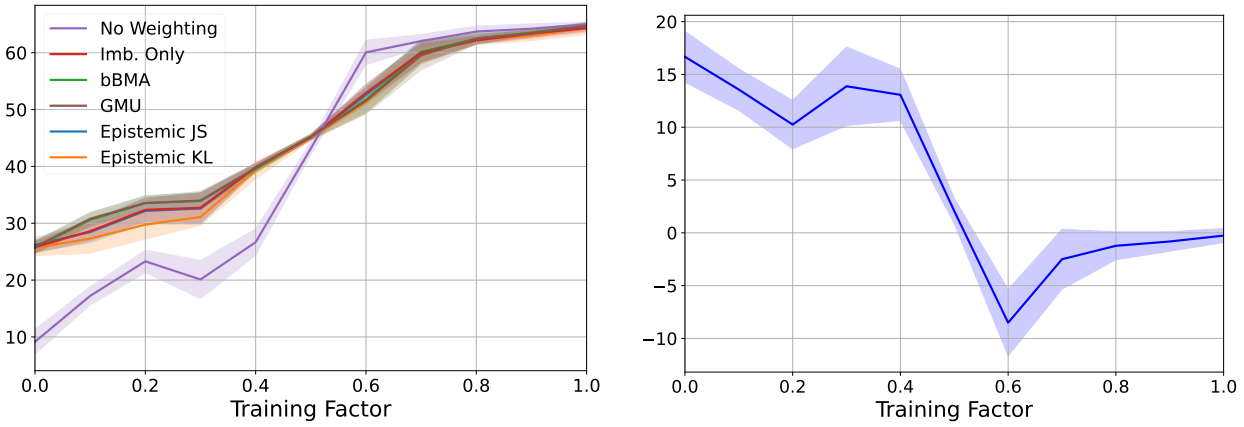
Table 2: **Focal Weighting F1 Score Comparisons (FUN)**. Candidates for uncertainty in focal weighting are compared on the FUN datasets: only imbalance weighting (Baseline), bBMA, GMU, and Epistemic Uncertainty evaluated using JS and KL. Means significantly ($> 2\sigma$) greater than baseline are bolded.

Method	CELLCYCLE	DERISI	EISEN	EXPR	GASCH-1	GASCH-2	SEQ	SPO
Baseline	6.51 ± 0.14	2.27 ± 0.04	9.31 ± 0.11	10.27 ± 0.24	9.80 ± 0.06	6.47 ± 0.19	8.84 ± 0.07	2.48 ± 0.03
bBMA	**7.39** ± 0.16	**2.54** ± 0.03	**9.79** ± 0.23	10.53 ± 0.35	10.24 ± 0.30	**7.23** ± 0.08	8.67 ± 0.12	**2.91** ± 0.06
GMU	**7.31** ± 0.20	**2.54** ± 0.03	**9.78** ± 0.22	10.53 ± 0.30	10.35 ± 0.35	**7.20** ± 0.07	8.65 ± 0.13	**3.03** ± 0.02
Epistemic JS	6.66 ± 0.17	2.29 ± 0.03	9.52 ± 0.13	10.34 ± 0.24	9.81 ± 0.06	6.52 ± 0.18	8.75 ± 0.20	2.50 ± 0.03
Epistemic KL	6.81 ± 0.16	2.28 ± 0.02	9.39 ± 0.12	10.58 ± 0.28	9.82 ± 0.07	6.35 ± 0.11	8.62 ± 0.24	2.48 ± 0.04

Notably, predictions show improvements over even the strongest performing approaches in Table 1. Part of the improvement is due to the additional nine models added within the ensemble. The degree to which improvements result from ensembling is represented via the baseline, which uses only imbalance weighting. There is strong evidence that both bBMA and GMU are practical candidates for focal weighting in improving F1 performance.

5.4 Echinoderm Vision Modelling

For our vision models, we experiment with a representative convolutional neural network — ResNet-50 (He et al., 2016). Our goal here was to determine if our weighting approach carries over benefits to HML for image classification on BenthicNet-E. We found that while the approach provided the same benefits in recall and thereby F1 for frozen untrained encoders, using a fully trained ImageNet-V2 encoder from Torchvision (Marcel & Rodriguez, 2010) showed that any benefits had shrunk to insignificance.



(a) F1

(b) Difference in F1

Figure 4: **Evaluation of F1 over Training Completion on Echinoderms.** The training factor acts as a parameter shifting from 0.0 (randomly initialized encoder) to 1.0 (fully trained ImageNet encoder). In Figure 4a, the weighting methods and an unweighted baseline are compared to each other. In Figure 4b, the difference between GMU and the unweighted baseline is shown, illustrating the dynamical advantage of the weighting method as the encoder becomes better trained.

To demonstrate this phenomenon, in Figure 4a, we show the effect on F1 for classifying echinoderms as we gradually transition from an untrained encoder to a trained one. In our results, we introduce a "training factor" mixing the untrained and pre-trained encoders. We find that as we emphasize the pre-training, the advantages for weighting gradually drop. We also find that bBMA and GMU focal-based approaches, when combined with imbalance weighting, produce the greatest advantages, as they did for the gene product datasets.

We suspect that vision encoders are already highly specialised for visual modalities, and so may avoid facing issues stemming from imbalanced data, at least for datasets with simpler hierarchies. Nonetheless, we believe that our results do point towards the possibility that a weighted approach may benefit encoders which are applied towards more challenging or potentially out-of-domain HML downstream tasks.

6 Conclusion

Our goal was to enable consistent detection of rare nodes deeper in a hierarchy, which are often ignored during model inference due to naturally arising sources of imbalance in datasets — a persistent problem in HML modelling. We attempted to achieve this objective through the exploration of node-wise imbalance weighting. We observe significant benefits to the recall of rare nodes in gene product datasets, with increases of up to five times that of existing methods, with models demonstrating the capability of predicting hierarchical nodes which were previously ignored. We also find that focal weighting with a sufficient ensemble size tends to improve recall, especially when focal loss incorporates the bBMA or the GMU as the uncertainty term.

References

Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018. doi: arXiv:1803.08375. URL http://arxiv.org/abs/1803.08375.

Franziska Althaus, Nicole Hill, Renata Ferrari, Luke Edwards, Rachel Przeslawski, Christine H. L. Schönberg, Rick Stuart-Smith, Neville Barrett, Graham Edgar, Jamie Colquhoun, Maggie Tran, Alan Jordan, Tony Rees, and Karen Gowlett-Holmes. A standardised vocabulary for identifying benthic biota and substrata from underwater imagery: The CATAMI classification scheme. *PLOS ONE*, 10(10):1–18, 10 2015. doi: 10.1371/journal.pone.0141039. URL https://doi.org/10.1371/journal.pone.0141039.

M Ashburner, C A Ball, J A Blake, D Botstein, H Butler, J M Cherry, A P Davis, K Dolinski, S S Dwight, J T Eppig, M A Harris, D P Hill, L Issel-Tarver, A Kasarskis, S Lewis, J C Matese, J E Richardson, M Ringwald, G M Rubin, and G Sherlock. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000. ISSN 1061-4036. doi: 10.1038/75556.

Wei Bi and James T. Kwok. Multi-label classification on tree- and dag-structured hierarchies. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 17–24, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Célia Blondin, Joris Guérin, Kelly Inagaki, Guilherme Longo, and Laure Berti-Équille. Hierarchical classification for automated image annotation of coral reef benthic structures, 2024. URL https://arxiv.org/abs/2412.08228.

Craig J Brown, Jessica A Sameoto, and Stephen J Smith. Multiple methods, maps, and management applications: Purpose made seafloor maps in support of ocean management. *Journal of Sea Research*, 72: 1–13, 2012. doi: 10.1016/j.seares.2012.04.009. URL https://doi.org/10.1016/j.seares.2012.04.009.

Francisco Charte, Antonio J. Rivera, María J. del Jesus, and Francisco Herrera. Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, 163:3–16, 2015a. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2014.08.091. URL https://www.sciencedirect.com/science/article/pii/S0925231215004269. Recent Advancements in Hybrid Artificial Intelligence Systems and its Application to Real-World Problems Progress in Intelligent Systems Mining Humanistic Data.

Francisco Charte, Antonio J. Rivera, María J. del Jesus, and Francisco Herrera. Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, 163:3–16, 2015b. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2014.08.091. URL https://www.sciencedirect.com/science/article/pii/S0925231215004269. Recent Advancements in Hybrid Artificial Intelligence Systems and its Application to Real-World Problems Progress in Intelligent Systems Mining Humanistic Data.

Francisco Charte, Antonio J. Rivera, María J. del Jesus, and Francisco Herrera. Mlsmote: Approaching imbalanced multilabel learning through synthetic instance generation. *Knowledge-Based Systems*, 89: 385–397, 2015c. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2015.07.019. URL https://www.sciencedirect.com/science/article/pii/S0950705115002737.

K. Chen, Bao-Liang Lu, and J.T. Kwok. Efficient classification of multi-label and imbalanced data using min-max modular classifiers. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 1770–1775, 2006. doi: 10.1109/IJCNN.2006.246893.

Joana C. Costa, Tiago Roxo, Hugo Proença, and Pedro Ricardo Morais Inácio. How deep learning sees the world: A survey on adversarial attacks & defenses. *IEEE Access*, 12:61113–61136, 2024. doi: 10.1109/ACCESS.2024.3395118.

Zachary Daniels and Dimitris Metaxas. Addressing imbalance in multi-label classification using structured hellinger forests. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. doi: 10.1609/aaai.v31i1.10908. URL https://ojs.aaai.org/index.php/AAAI/article/view/10908.

H. Martin Gillis, Isaac Xu, Benjamin Misiuk, Craig J. Brown, and Thomas Trappenberg. Last-layer committee machines for uncertainty estimations of benthic imagery. *arXiv*, 2025a. doi: arXiv:2504.16952. URL https://arxiv.org/abs/2504.16952.

H. Martin Gillis, Isaac Xu, and Thomas Trappenberg. Uncertainty estimation using variance-gated distributions. *arXiv*, 2025b. doi: https://doi.org/10.48550/arXiv.2509.08846. URL https://arxiv.org/abs/2509.08846.

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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

E. Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, 1909. doi: doi:10.1515/crll.1909.136.210. URL https://doi.org/10.1515/crll.1909.136.210.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CoRR*, abs/1907.07174, 2019. doi: arXiv:1907.07174. URL http://arxiv.org/abs/1907.07174.

Roberta Eleanor Hunt and Kim Steenstrup Pedersen. Rove-tree-11: The not-so-wild rover. a hierarchically structured image dataset for deep metric learning research. In *Computer Vision – ACCV 2022*, Lecture Notes in Computer Science, pp. 2967–2983, Switzerland, 2023. Springer. 16th Asian Conference on Computer Vision, ACCV 2022 ; Conference date: 04-12-2022 Through 08-12-2022.

Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *arXiv*, October 2017. doi: 10.48550/arXiv.1703.04977. URL http://arxiv.org/abs/1703.04977. arXiv:1703.04977 [cs].

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. ISSN 1573-1405. doi: 10.1007/s11263-020-01316-z. URL https://doi.org/10.1007/s11263-020-01316-z.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks, 2015. URL https://arxiv.org/abs/1511.06314.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. doi: 10.1109/TPAMI.2018.2858826.

Bin Liu, Konstantinos Blekas, and Grigorios Tsoumakas. Multi-label sampling based on local label imbalance. *Pattern Recognition*, 122:108294, 2022. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2021.108294. URL https://www.sciencedirect.com/science/article/pii/S003132032100474X.

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. *Scientific Data*, 12(1):230, 2025. ISSN 2052-4463. doi: 10.1038/s41597-025-04491-1. URL https://doi.org/10.1038/s41597-025-04491-1.

Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pp. 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589336. doi: 10.1145/1873951.1874254. URL https://doi.org/10.1145/1873951.1874254.

Maja Pavlovic. Understanding model calibration – a gentle introduction and visual exploration of calibration and the expected calibration error (ece). *arXiv*, 2025. doi: arXiv:2501.19047. URL https://arxiv.org/abs/2501.19047.

Rodolfo M. Pereira, Yandre M.G. Costa, and Carlos N. Silla Jr. Toward hierarchical classification of imbalanced data using random resampling algorithms. *Information Sciences*, 578:344–363, 2021. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2021.07.033. URL https://www.sciencedirect.com/science/article/pii/S0020025521007234.

Rodolfo Miranda Pereira, Yandre Maldonado e Gomes da Costa, and Carlos Nascimento Silla. Dealing with imbalanceness in hierarchical multi-label datasets using multi-label resampling techniques. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 818–824, 2018. doi: 10.1109/ICTAI.2018.00128.

Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995. doi: 10.1080/09540099550039318. URL https://doi.org/10.1080/09540099550039318.

Miguel Romero, Felipe Kenji Nakano, Jorge Finke, Camilo Rocha, and Celine Vens. Hierarchy exploitation to detect missing annotations on hierarchical multi-label classification. *arXiv*, 2022. doi: arXiv:2207.06237. URL https://arxiv.org/abs/2207.06237.

Andreas Ruepp, Alfred Zollner, Dieter Maier, Kaj Albermann, Jean Hani, Martin Mokrejs, Igor Tetko, Ulrich Güldener, Gertrud Mannhaupt, Martin Münsterkötter, and H Werner Mewes. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32 (18):5539–5545, 2004. ISSN 1362-4962. doi: 10.1093/nar/gkh894.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. Introducing an improved information-theoretic measure of predictive uncertainty, 2023. URL https://arxiv.org/abs/2311.08309.

Daniel L. Silver, Qiang Yang, and Lianghao Li. Lifelong machine learning systems: Beyond learning algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*, 2013. URL https://api.semanticscholar.org/CorpusID:18548239.

Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965, 2021. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2021.107965. URL https://www.sciencedirect.com/science/article/pii/S0031320321001527.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. Hierarchical multi-label classification networks. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5075–5084. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/wehrmann18a.html.

Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In Robin J. Evans and Ilya Shpitser (eds.), *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 2282–2292. PMLR, 31 Jul–04 Aug 2023. URL https://proceedings.mlr.press/v216/wimmer23a.html.

Min-Ling Zhang, Yu-Kun Li, Hao Yang, and Xu-Ying Liu. Towards class-imbalance aware multi-label learning. *IEEE Transactions on Cybernetics*, 52(6):4459–4471, 2022. doi: 10.1109/TCYB.2020.3027509.

Zhinuo Zhou, Ge-Yi Fu, Yi Fang, Ye Yuan, Hong-Bin Shen, Chun-Sheng Wang, Xue-Wei Xu, Peng Zhou, and Xiaoyong Pan. Echoai: A deep-learning based model for classification of echinoderms in global oceans. *Frontiers in Marine Science*, 10, 2023. ISSN 2296-7745. doi: 10.3389/fmars.2023.1147690. URL https://www.frontiersin.org/journals/marine-science/articles/10.3389/fmars.2023.1147690.

A Hyperparameters

In this section, we discuss the hyperparameter settings used for our experiments.

A.1 Gene Product Modelling

Our hyperparameters for all methods applied to the gene product are identical to those retrieved from Giunchiglia & Lukasiewicz (2020), to ensure fair comparisons between the approaches. For all our experiments on gene product datasets, we use a batch size of four and set the number of layers equal to three. In other words, we have one input layer mapping from input size to hidden size, one hidden layer mapping from hidden size to the same hidden size, and finally one output layer mapping from hidden size to output size. Additionally, the layers use a dropout of 0.7, and ReLU (Agarap, 2018) as a non-linear activation.

Table 3: **Hyperparameters for experiments ran on FUN and GO datasets.** Input, Output, and Hidden refer to dimensional size.

Dataset	**Input**	**Output (FUN)**	**Output (GO)**	**Hidden (FUN)**	**Hidden (GO)**	**LR (FUN)**	**LR (GO)**	**Epochs (FUN/GO)**
Cellcycle	77	499	4122	500	1000	1e−4	1e−4	106 / 62
Derisi	63	499	4116	500	500	1e−4	1e−4	67 / 91
Eisen	79	461	3570	500	500	1e−4	1e−4	110 / 123
Expr	561	499	4128	1250	4000	1e−4	1e−4	20 / 70
Gasch1	173	499	4122	1000	500	1e−4	1e−4	42 / 122
Gasch2	52	499	4128	500	500	1e−4	1e−4	123 / 177
Seq	529	499	4130	2000	9000	1e−4	1e−4	13 / 45
Spo	86	499	4116	250	500	1e−4	1e−4	115 / 103

A.2 Echinoderm Modelling

The most drastic difference for our models on echinoderm data is the use of a frozen shared vision encoder. The employed batch size was 128. Given the common encoder, we opt for a multi-layer perceptron classifier component of the same design, consisting of three layers for our ensembles. Dimensional mapping occurs from the embedding dimension of the vision architectures used to the hidden dimension (equal to the embedding dimension), and then to a dimensional size of 13, reflecting the number of nodes in the echinoderm hierarchy. Additionally, we use a learning rate of 1×10^{-3} and a weight decay of 1×10^{-4}, training for 20 epochs.

For our training factor analysis, we apply a λ and $1 - \lambda$ mixing between the weights of two encoders: a randomly initialized one and an ImageNet pre-trained one. As we gradually increase the training factor, we run trials using each of the compared methods in accordance to the described settings.

B Focal Weighted FUN Results

In this section, in Table 4, we present the full results for our experiments with focal weighting on the FUN datasets. We see that while gains in recall and F1 are almost always present for bBMA and "GMU" based uncertainty terms, there are no statistically significant trends for any of the other measures. Notably, we see that even in cases where F1 was not significant, as in "Expr" and "Seq" datasets, the means were still greater. Based on these observations, we believe that including a focal term is a strategy that generates benefits but produces little downside.

C Experiments on GO Hierarchy

In Table 5 and Table 6, we present our results for the GO datasets. The overall trend in performance match those presented for the FUN hierarchy. However, we note the significant drop in HROS-PD performance. This decrease is expected as the method was not designed for DAG hierarchies (Pereira et al., 2021).

Table 4: **Focal Weighting Comparisons (FUN)**. Different uncertainties for implementing focal weighting are compared for the FUN datasets: only imbalance weighting (Baseline), bBMA, GMU, and Epistemic Uncertainty evaluated using JS and KL. Means which are significantly ($> 2\sigma$) greater than the baseline are bolded.

Dataset (FUN)	Method	F1	Precision	Recall	Bin. AP	AP
CELLCYCLE	Baseline	6.51 ± 0.14	9.83 ± 0.31	6.56 ± 0.10	10.53 ± 0.11	25.32 ± 0.03
	bBMA	**7.39** ± 0.16	9.47 ± 0.30	**7.65** ± 0.15	10.46 ± 0.07	24.79 ± 0.06
	GMU	**7.31** ± 0.20	9.36 ± 0.36	**7.57** ± 0.19	10.44 ± 0.10	24.81 ± 0.07
	Epistemic JS	6.66 ± 0.17	9.83 ± 0.28	6.70 ± 0.13	10.58 ± 0.11	**25.43** ± 0.04
	Epistemic KL	6.81 ± 0.16	10.35 ± 0.38	6.77 ± 0.15	10.62 ± 0.05	**25.60** ± 0.02
DERISI	Baseline	2.27 ± 0.04	3.35 ± 0.28	2.61 ± 0.02	7.66 ± 0.04	19.19 ± 0.05
	bBMA	**2.54** ± 0.03	**3.53** ± 0.04	**2.87** ± 0.01	7.70 ± 0.03	19.19 ± 0.05
	GMU	**2.54** ± 0.03	**3.52** ± 0.07	**2.88** ± 0.02	7.70 ± 0.03	19.20 ± 0.05
	Epistemic JS	2.29 ± 0.03	3.51 ± 0.21	2.63 ± 0.02	7.68 ± 0.04	19.20 ± 0.05
	Epistemic KL	2.28 ± 0.02	3.52 ± 0.18	2.61 ± 0.02	7.70 ± 0.03	19.19 ± 0.07
EISEN	Baseline	9.31 ± 0.11	12.87 ± 0.26	8.85 ± 0.10	12.57 ± 0.10	30.22 ± 0.07
	bBMA	**9.79** ± 0.23	12.42 ± 0.36	**9.62** ± 0.18	12.11 ± 0.13	29.60 ± 0.07
	GMU	**9.78** ± 0.22	12.50 ± 0.33	**9.59** ± 0.17	12.16 ± 0.13	29.65 ± 0.07
	Epistemic JS	9.52 ± 0.13	13.32 ± 0.30	8.98 ± 0.10	12.59 ± 0.08	30.34 ± 0.07
	Epistemic KL	9.39 ± 0.12	13.18 ± 0.34	8.89 ± 0.10	12.58 ± 0.06	**30.54** ± 0.06
EXPR	Baseline	10.27 ± 0.24	11.89 ± 0.17	10.68 ± 0.30	12.21 ± 0.09	29.68 ± 0.07
	bBMA	10.53 ± 0.35	11.36 ± 0.21	11.53 ± 0.47	11.91 ± 0.13	29.02 ± 0.11
	GMU	10.53 ± 0.30	11.31 ± 0.23	**11.54** ± 0.39	11.89 ± 0.08	29.03 ± 0.11
	Epistemic JS	10.34 ± 0.24	11.82 ± 0.23	10.79 ± 0.29	12.25 ± 0.13	29.76 ± 0.08
	Epistemic KL	10.58 ± 0.28	11.96 ± 0.28	11.09 ± 0.33	12.26 ± 0.14	29.89 ± 0.12
GASCH-1	Baseline	9.80 ± 0.06	12.73 ± 0.22	9.67 ± 0.08	11.41 ± 0.05	27.94 ± 0.11
	bBMA	10.24 ± 0.30	12.08 ± 0.51	**10.71** ± 0.18	10.92 ± 0.10	27.11 ± 0.12
	GMU	10.35 ± 0.35	12.23 ± 0.54	**10.79** ± 0.26	10.94 ± 0.11	27.12 ± 0.12
	Epistemic JS	9.81 ± 0.06	12.58 ± 0.21	9.72 ± 0.04	11.35 ± 0.03	27.96 ± 0.11
	Epistemic KL	9.82 ± 0.07	12.60 ± 0.09	9.70 ± 0.04	11.34 ± 0.02	28.02 ± 0.10
GASCH-2	Baseline	6.47 ± 0.19	9.70 ± 0.46	6.42 ± 0.16	10.56 ± 0.06	25.54 ± 0.02
	bBMA	**7.23** ± 0.08	9.60 ± 0.27	**7.43** ± 0.11	10.46 ± 0.05	25.30 ± 0.04
	GMU	**7.20** ± 0.07	9.59 ± 0.17	**7.37** ± 0.13	10.45 ± 0.06	25.32 ± 0.03
	Epistemic JS	6.52 ± 0.18	9.78 ± 0.39	6.46 ± 0.16	10.60 ± 0.04	**25.62** ± 0.02
	Epistemic KL	6.35 ± 0.11	9.71 ± 0.45	6.32 ± 0.09	10.61 ± 0.09	**25.76** ± 0.02
SEQ	Baseline	8.84 ± 0.07	13.11 ± 0.28	8.31 ± 0.07	12.41 ± 0.11	29.46 ± 0.10
	bBMA	8.67 ± 0.12	11.70 ± 0.21	8.56 ± 0.14	12.02 ± 0.09	29.23 ± 0.08
	GMU	8.65 ± 0.13	11.61 ± 0.36	8.54 ± 0.12	12.03 ± 0.13	29.25 ± 0.10
	Epistemic JS	8.75 ± 0.20	12.69 ± 0.41	8.32 ± 0.17	12.39 ± 0.10	29.59 ± 0.09
	Epistemic KL	8.62 ± 0.24	12.22 ± 0.49	8.34 ± 0.22	12.33 ± 0.12	**29.78** ± 0.11
SPO	Baseline	2.48 ± 0.03	3.87 ± 0.31	2.79 ± 0.02	8.21 ± 0.06	21.18 ± 0.02
	bBMA	**2.91** ± 0.06	3.78 ± 0.20	**3.20** ± 0.06	8.19 ± 0.04	21.16 ± 0.02
	GMU	**3.03** ± 0.02	**4.08** ± 0.06	**3.27** ± 0.03	8.22 ± 0.03	21.19 ± 0.02
	Epistemic JS	2.50 ± 0.03	3.86 ± 0.19	2.80 ± 0.03	8.21 ± 0.06	21.24 ± 0.03
	Epistemic KL	2.48 ± 0.04	3.51 ± 0.20	2.77 ± 0.04	8.20 ± 0.06	**21.35** ± 0.04

D Combining Imbalance Weighted Learning Objectives

Given our observations, we are motivated to ask if it is possible to mitigate the decrease in precision by setting $\tilde{w}_0$ to a scheduler. We hypothesized that, as a model learns to recall certain nodes given the weighted learning objective, if we switch to an unweighted learning objective later on in training, once it has already learned to recall certain nodes, precision decline may be mitigated. In testing, we explored both an epoch-based scheduler — where we progressively approach an unweighted learning objective in the later epochs, and a batch-based scheduler — where we approach the unweighted objective every epoch. We find that batch-based

Table 5: **HML Method Comparisons (GO)**. Different strategies in addressing HML imbalance are compared for the FUN datasets: no method used (None), HROS-PD, and node weighting using $\tilde{w}_0$ values of 0.25 and 0.50. Means which are significantly ($> 2\sigma$) greater than all others for a dataset and a measure are bolded.

Dataset (GO)	Method	F1	Precision	Recall	Bin. AP	AP
CELLCYCLE	None	1.16 ± 0.07	3.23 ± 0.19	0.86 ± 0.05	16.31 ± 0.46	41.34 ± 0.10
	LPROS	1.38 ± 0.05	**4.05** ± 0.13	1.01 ± 0.04	16.45 ± 0.44	40.70 ± 0.14
	HROS-PD	0.59 ± 0.06	2.26 ± 0.31	0.43 ± 0.04	11.37 ± 0.60	40.29 ± 0.11
	$\tilde{w}_0 = 0.25$	**1.94** ± 0.07	3.22 ± 0.26	**1.86** ± 0.05	**19.84** ± 0.09	40.93 ± 0.08
	$\tilde{w}_0 = 0.50$	1.47 ± 0.04	3.29 ± 0.14	1.23 ± 0.04	19.49 ± 0.14	41.24 ± 0.12
DERISI	None	0.34 ± 0.01	0.69 ± 0.12	0.30 ± 0.01	12.88 ± 0.36	37.01 ± 0.06
	LPROS	0.33 ± 0.03	0.63 ± 0.07	0.28 ± 0.03	12.77 ± 0.78	37.02 ± 0.07
	HROS-PD	0.19 ± 0.02	0.55 ± 0.05	0.15 ± 0.02	8.79 ± 0.39	36.38 ± 0.04
	$\tilde{w}_0 = 0.25$	**0.77** ± 0.05	0.97 ± 0.22	**0.95** ± 0.04	**17.38** ± 0.08	36.86 ± 0.05
	$\tilde{w}_0 = 0.50$	0.53 ± 0.02	0.78 ± 0.13	0.57 ± 0.02	17.22 ± 0.08	36.95 ± 0.06
EISEN	None	1.90 ± 0.18	4.50 ± 0.32	1.45 ± 0.15	19.57 ± 0.61	45.34 ± 0.15
	LPROS	2.15 ± 0.10	4.90 ± 0.34	1.62 ± 0.07	19.25 ± 0.10	44.70 ± 0.15
	HROS-PD	1.41 ± 0.09	4.17 ± 0.19	1.02 ± 0.07	16.19 ± 0.51	44.66 ± 0.16
	$\tilde{w}_0 = 0.25$	**3.17** ± 0.07	4.78 ± 0.11	**3.04** ± 0.09	**22.47** ± 0.13	44.75 ± 0.14
	$\tilde{w}_0 = 0.50$	2.40 ± 0.22	4.40 ± 0.36	2.04 ± 0.18	22.38 ± 0.16	45.14 ± 0.14
EXPR	None	3.80 ± 0.17	7.41 ± 0.37	2.98 ± 0.20	16.93 ± 0.25	38.13 ± 0.25
	LPROS	3.75 ± 0.05	7.31 ± 0.24	2.93 ± 0.07	16.48 ± 0.25	37.47 ± 0.30
	HROS-PD	2.35 ± 0.14	5.94 ± 0.63	1.69 ± 0.11	15.15 ± 0.70	38.89 ± 0.31
	$\tilde{w}_0 = 0.25$	**4.61** ± 0.10	6.87 ± 0.15	**4.13** ± 0.13	**19.37** ± 0.28	**40.41** ± 0.43
	$\tilde{w}_0 = 0.50$	4.31 ± 0.09	7.42 ± 0.20	3.56 ± 0.12	18.11 ± 0.23	39.21 ± 0.18
GASCH-1	None	1.52 ± 0.03	3.08 ± 0.20	1.19 ± 0.02	17.79 ± 0.25	43.61 ± 0.12
	LPROS	1.97 ± 0.07	4.01 ± 0.17	1.56 ± 0.05	18.04 ± 0.44	43.27 ± 0.06
	HROS-PD	0.86 ± 0.06	2.38 ± 0.22	0.63 ± 0.05	12.96 ± 0.51	42.48 ± 0.19
	$\tilde{w}_0 = 0.25$	**2.66** ± 0.08	3.98 ± 0.25	**2.61** ± 0.05	**21.48** ± 0.09	43.40 ± 0.11
	$\tilde{w}_0 = 0.50$	2.06 ± 0.06	3.53 ± 0.20	1.80 ± 0.04	21.21 ± 0.14	43.57 ± 0.11
GASCH-2	None	1.19 ± 0.02	3.52 ± 0.10	0.86 ± 0.02	16.43 ± 0.49	41.56 ± 0.10
	LPROS	1.30 ± 0.04	**3.80** ± 0.03	0.95 ± 0.04	16.54 ± 0.48	41.45 ± 0.12
	HROS-PD	0.49 ± 0.03	1.57 ± 0.10	0.36 ± 0.02	11.10 ± 0.57	40.34 ± 0.10
	$\tilde{w}_0 = 0.25$	**1.84** ± 0.09	3.21 ± 0.26	**1.84** ± 0.08	20.03 ± 0.12	41.13 ± 0.06
	$\tilde{w}_0 = 0.50$	1.42 ± 0.06	3.42 ± 0.30	1.20 ± 0.05	19.82 ± 0.10	41.40 ± 0.07
SEQ	None	3.94 ± 0.25	8.96 ± 0.40	2.91 ± 0.19	17.77 ± 0.38	38.60 ± 0.28
	LPROS	4.07 ± 0.14	9.34 ± 0.38	3.01 ± 0.10	16.74 ± 0.59	37.30 ± 0.81
	HROS-PD	2.46 ± 0.21	6.84 ± 0.37	1.72 ± 0.17	15.17 ± 0.81	39.63 ± 0.33
	$\tilde{w}_0 = 0.25$	4.46 ± 0.33	7.93 ± 0.66	3.65 ± 0.26	**19.69** ± 0.27	39.90 ± 0.32
	$\tilde{w}_0 = 0.50$	4.29 ± 0.21	8.63 ± 0.23	3.31 ± 0.19	18.95 ± 0.37	39.22 ± 0.44
SPO	None	0.68 ± 0.10	1.85 ± 0.33	0.54 ± 0.07	13.65 ± 0.56	38.19 ± 0.05
	LPROS	0.78 ± 0.09	2.25 ± 0.42	0.59 ± 0.05	13.51 ± 0.27	37.94 ± 0.06
	HROS-PD	0.31 ± 0.05	0.94 ± 0.37	0.25 ± 0.03	9.15 ± 0.32	36.98 ± 0.11
	$\tilde{w}_0 = 0.25$	**1.26** ± 0.02	2.29 ± 0.24	**1.29** ± 0.01	**18.12** ± 0.04	37.88 ± 0.07
	$\tilde{w}_0 = 0.50$	1.01 ± 0.02	2.49 ± 0.11	0.89 ± 0.02	17.95 ± 0.20	38.09 ± 0.05

schedulers perform significantly stronger. While this result isn't surprising from a life-long learning or multi-task learning perspective (Robins, 1995; Silver et al., 2013), it seems to suggest that the loss landscapes for an HML imbalanced weighted objective and an unweighted objective do not easily transition from one to the other and are likely conflicting.

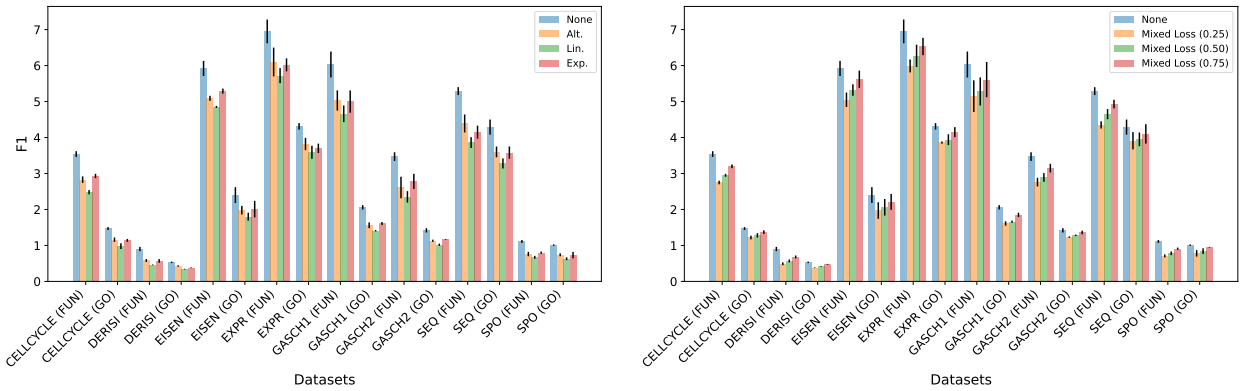
In total, we explore four approaches in combining our learning objectives: linear, exponential, alternating, and mixed loss. Of the four, alternating and mixed loss explore an alternative hypothesis, that the order in which the weighted and unweighted tasks occur do not matter. Alternating refers to alternating between the

Table 6: **Focal Weighting Comparisons (GO)**. Different uncertainties for implementing focal weighting are compared for the FUN datasets: only imbalance weighting (Baseline), bBMA, GMU, and Epistemic Uncertainty evaluated using JS and KL. Means which are significantly ($> 2\sigma$) greater than the baseline are bolded.

Dataset (GO)	Method	F1	Precision	Recall	Bin. AP	AP
CELLCYCLE	Baseline	3.50 ± 0.04	5.26 ± 0.20	3.24 ± 0.03	19.71 ± 0.06	40.39 ± 0.04
	bBMA	**3.77** ± 0.06	5.03 ± 0.11	**3.67** ± 0.06	18.92 ± 0.10	39.59 ± 0.05
	GMU	**3.79** ± 0.05	5.04 ± 0.07	**3.69** ± 0.04	18.89 ± 0.10	39.56 ± 0.04
	Epistemic JS	3.51 ± 0.03	5.25 ± 0.07	3.26 ± 0.03	19.67 ± 0.04	40.34 ± 0.04
	Epistemic KL	3.53 ± 0.04	5.17 ± 0.19	3.30 ± 0.03	19.64 ± 0.09	40.31 ± 0.07
DERISI	Baseline	1.28 ± 0.04	1.93 ± 0.12	1.37 ± 0.04	17.26 ± 0.01	36.58 ± 0.02
	bBMA	**1.40** ± 0.02	1.97 ± 0.10	**1.51** ± 0.03	17.03 ± 0.04	36.38 ± 0.01
	GMU	**1.40** ± 0.02	1.96 ± 0.10	**1.51** ± 0.01	17.00 ± 0.03	36.38 ± 0.01
	Epistemic JS	1.27 ± 0.02	1.81 ± 0.08	1.37 ± 0.03	17.24 ± 0.03	36.56 ± 0.02
	Epistemic KL	1.28 ± 0.04	1.90 ± 0.15	1.37 ± 0.03	17.25 ± 0.05	36.54 ± 0.01
EISEN	Baseline	5.16 ± 0.11	7.20 ± 0.24	4.93 ± 0.08	22.37 ± 0.06	45.05 ± 0.04
	bBMA	**5.54** ± 0.12	6.93 ± 0.22	**5.59** ± 0.15	21.62 ± 0.11	44.41 ± 0.09
	GMU	**5.56** ± 0.13	6.98 ± 0.17	**5.60** ± 0.17	21.62 ± 0.08	44.41 ± 0.08
	Epistemic JS	5.19 ± 0.12	7.10 ± 0.23	5.01 ± 0.14	22.34 ± 0.09	45.07 ± 0.04
	Epistemic KL	5.18 ± 0.11	7.09 ± 0.17	5.00 ± 0.14	22.34 ± 0.06	45.09 ± 0.07
EXPR	Baseline	5.28 ± 0.12	8.67 ± 0.25	4.50 ± 0.09	16.34 ± 0.09	36.13 ± 0.06
	bBMA	5.20 ± 0.10	8.29 ± 0.14	4.54 ± 0.09	16.42 ± 0.12	35.78 ± 0.08
	GMU	5.26 ± 0.07	8.28 ± 0.11	4.57 ± 0.09	16.51 ± 0.12	35.84 ± 0.07
	Epistemic JS	5.24 ± 0.08	8.37 ± 0.15	4.47 ± 0.05	16.51 ± 0.13	35.95 ± 0.01
	Epistemic KL	5.48 ± 0.11	8.64 ± 0.22	**4.77** ± 0.07	**16.95** ± 0.13	36.27 ± 0.09
GASCH-1	Baseline	4.23 ± 0.07	5.62 ± 0.10	4.32 ± 0.05	21.12 ± 0.03	43.93 ± 0.02
	bBMA	**4.92** ± 0.12	5.90 ± 0.20	**5.33** ± 0.09	20.28 ± 0.10	43.16 ± 0.06
	GMU	**4.94** ± 0.11	5.92 ± 0.21	**5.33** ± 0.12	20.26 ± 0.07	43.13 ± 0.08
	Epistemic JS	4.30 ± 0.06	5.63 ± 0.09	4.43 ± 0.07	21.09 ± 0.05	43.92 ± 0.03
	Epistemic KL	4.29 ± 0.03	5.75 ± 0.08	**4.39** ± 0.03	21.17 ± 0.07	43.98 ± 0.05
GASCH-2	Baseline	3.40 ± 0.04	5.24 ± 0.07	3.37 ± 0.04	20.23 ± 0.06	41.73 ± 0.02
	bBMA	**3.71** ± 0.05	5.11 ± 0.14	**3.74** ± 0.05	19.76 ± 0.07	41.49 ± 0.04
	GMU	**3.75** ± 0.09	5.23 ± 0.12	**3.77** ± 0.08	19.81 ± 0.08	41.47 ± 0.02
	Epistemic JS	3.39 ± 0.09	5.14 ± 0.13	3.38 ± 0.08	20.20 ± 0.08	41.71 ± 0.02
	Epistemic KL	3.35 ± 0.06	5.07 ± 0.14	3.31 ± 0.06	20.23 ± 0.04	41.70 ± 0.03
SEQ	Baseline	7.52 ± 0.08	14.03 ± 0.25	6.04 ± 0.06	17.77 ± 0.08	39.39 ± 0.04
	bBMA	7.31 ± 0.14	13.55 ± 0.33	5.88 ± 0.11	17.30 ± 0.10	38.95 ± 0.08
	GMU	7.43 ± 0.10	13.80 ± 0.33	5.98 ± 0.07	17.28 ± 0.13	38.91 ± 0.08
	Epistemic JS	7.50 ± 0.13	14.08 ± 0.09	6.01 ± 0.12	17.83 ± 0.18	39.19 ± 0.13
	Epistemic KL	7.73 ± 0.12	14.10 ± 0.31	6.24 ± 0.10	**18.11** ± 0.05	39.41 ± 0.10
SPO	Baseline	2.01 ± 0.06	3.36 ± 0.06	1.91 ± 0.05	17.91 ± 0.03	37.57 ± 0.03
	bBMA	**2.28** ± 0.06	3.47 ± 0.06	**2.21** ± 0.05	17.58 ± 0.05	37.49 ± 0.03
	GMU	**2.34** ± 0.05	**3.54** ± 0.07	**2.26** ± 0.04	17.62 ± 0.07	37.49 ± 0.02
	Epistemic JS	2.03 ± 0.07	3.32 ± 0.08	1.93 ± 0.06	17.89 ± 0.05	37.57 ± 0.04
	Epistemic KL	2.04 ± 0.08	3.41 ± 0.15	1.92 ± 0.06	17.96 ± 0.08	**37.64** ± 0.02

weighted and unweighted objectives every batch, and mixed loss imposes a λ and $1 - \lambda$ relationship between the two objectives.

For our linear scheduler, we evenly space out $\tilde{w}_i$ update steps to the number of training steps (N_{steps}) in each epoch. Our exponential scheduler divides exponential steps as $\Delta\tilde{w}_i = \frac{\mathbf{1}-\tilde{w}_i}{(N_{steps}-1)^k}$. Exponential updates are then performed as $\tilde{w}_{i,t+1} = \tilde{w}_{i,t} + t^k \Delta\tilde{w}_i$, where we introduce the second subscript t to describe the batch index in an epoch, while k is the exponential term. For our experiments, we set $k = 3$.



(a) Schedulers

(b) λ-Mixed Loss

Figure 5: **Combining Weighted and Unweighted Objectives.** (a) Scheduler effect on F1 score across all FUN and GO datasets, and (b) mixed loss objective, with varying λ effect.

Overall, despite the variety of methods to combine learning objectives, none surpassed results from working solely with the weighted loss. our results are presented in Figure 5.

E Combining Imbalance Weighting with LPROS

Table 7: **Combined LPROS and Imbalance Weighting F1 Performance.** Results which exceed 2 σ in comparison to the previously best (Prev. Best) performing method for their corresponding dataset are bolded. Previous best values are attained from Table 1 and Table 5.

	LPROS + Imb. Weighting		**Prev. Best**	
Dataset	FUN	GO	FUN	GO
Cellcycle	**5.22** ± 0.16	**2.33** ± 0.18	4.85 ± 0.11	1.94 ± 0.07
Derisi	**2.01** ± 0.08	**0.94** ± 0.05	1.82 ± 0.05	0.77 ± 0.05
Eisen	**8.28** ± 0.29	**3.52** ± 0.12	7.16 ± 0.16	3.17 ± 0.07
Expr	**9.29** ± 0.28	4.40 ± 0.11	8.50 ± 0.30	**4.61** ± 0.10
Gasch1	**8.05** ± 0.16	**3.09** ± 0.04	7.42 ± 0.36	2.66 ± 0.08
Gasch2	**5.39** ± 0.22	**2.09** ± 0.05	4.92 ± 0.16	1.84 ± 0.09
Seq	**7.57** ± 0.35	4.60 ± 0.14	6.60 ± 0.12	4.46 ± 0.33
Spo	**2.24** ± 0.03	1.35 ± 0.08	1.94 ± 0.04	1.26 ± 0.02

Rather than attempting to merge losses to enhance precision in imbalance weighted learning as in Appendix D, we could seek to integrate resampling methods instead. In Table 1 and Table 5, we observe that LPROS often attains the highest precision in comparison to other approaches. Furthermore, LPROS may be applied in conjunction with DAG-like hierarchies as in the case of our GO datasets, in contrast to HROS-PD. We are therefore motivated to ask if this performance could be combined constructively with the effects of imbalance weighting.

Firstly, examining the individual node performances across datasets revealed that LPROS impacts the nodes in a multi-faceted manner. While some nodes traded recall for improved precision, others saw precision gains. Additionally, the LPROS-trained model began predicting previously unpredicted nodes, but also ceased predictions for some previously predicted nodes. These results suggest that LPROS behaves differently from imbalance weighting, and their combination may derive synergistic benefits. For the purposes of this work, and given finite computational budgets, we calculate node weights after resampling. We reason that this ordering is a safer option to provide beneficial results, given the unknown effects of their combination. Our results are presented in Table 7.

As observed, the combined approach exceeds the previous best method (imbalance weighting with $\tilde{w}_0 = 0.25$) across almost all gene product datasets. Although we had expected a jump in F1 score due to increased precision, which was indeed observed, surprisingly, the recall too was boosted as a result of this combination. However, we caution that the combined method did not perform strongly on our echinoderm dataset. It is difficult to ascertain why this unexpected difference in performance occurred, as there may be a myriad of issues pertaining to computer vision, such as hyperparameter specifications for differing architectures, or augmentation-related influences, and so on. Further investigation would be required.