When Constraints Go Both Ways: Dynamic Hierarchical Multi-label Classification Networks

Gelsomina Di Palma
DIBRIS
University of Genova
Genova, Italy
gelsomina.dipalma99@gmail.com

Zhiming Wang
Electrical & Electronic Engineering
Imperial College London
London, United Kingdom
zhimingwang2002@outlook.com

Eleonora Giunchiglia
Electrical & Electronic Engineering
Imperial College London
London, United Kingdom
e.giunchiglia@imperial.ac.uk

***Abstract*—Hierarchical Multi-label Classification (HMC) problem assigns multiple labels to each instance under the requirement that predictions satisfy a set of hierarchical subclass constraints among labels. The current state-of-the-art models tend to enforce coherence by exploiting the *sufficient direction* of each constraint by propagating positive evidence from subclasses to superclasses. In this paper, we extend that idea by additionally exploiting the *necessary direction* by suppressing a class when its superclass is not supported. The resulting model, Dynamic Hierarchical Network (DHN), is thus able to leverage both sufficient and necessary conditions while preserving coherence by construction. An extensive evaluation across 30 real-world benchmark datasets shows that DHN consistently improves over competitive baselines.**

***Index Terms*—Hierarchical Multi-label Classification, Neural Networks, Hierarchical Constraints**

I. Introduction

Multi-label Classification (MC) is a standard machine learning problem where a set of classes is assigned to each data point [1]–[3]. Hierarchical Multi-label Classification (HMC) problems are more complex, as they are MC problems in which classes have to comply with a set of hierarchy constraints of the form: $A \rightarrow B$, meaning that class A is a subclass of class B, i.e., every data point that is assigned label A must also be assigned label B. In HMC problems, all predictions must satisfy a set of hierarchy constraints. HMC problems arise in many fields, including image classification [4], functional genomics [5] and text categorization [6]. The challenges associated with HMC are primarily driven by two factors, as noted in [4], [7]:

1) Significant class imbalance, where classes at deeper levels of the hierarchy tend to have much fewer positive examples available for training.
2) The necessity for predictions to maintain coherence, i.e., satisfy the hierarchy constraints.

Many models have been developed during the years to deal with HMC problems and can be grouped into those that directly produce an output coherent with the hierarchy constraints (see, e.g., [8], [9]) and those that allow the output to be incoherent, requiring additional steps at inference time to ensure the constraints' satisfaction (see, e.g., [10], [11]). Most of the state-of-the-art HMC models based on neural networks belong to the second group [10], [12], [13], and different post-processing techniques can be applied to guarantee that the final output of the model is coherent with respect to the constraints [11]. An exception is given by C-HMCNN [14], where the constraints are encoded in a layer that can be built on top of any neural network. This layer is able to both enforce the constraints by design and improve the performance of the underlying model at the same time. However, it gives a strict operational semantic to each constraint $A \rightarrow B$, by exploiting only its *sufficient direction*, i.e., it uses only the fact that if A is positive then B should be positive as well. This behaviour does not take into account of the fact that $A \rightarrow B$ can be equivalently rewritten as $\neg B \rightarrow \neg A$, and thus the *necessary direction* of the constraint can be exploited as well, i.e., if B is negative then we can also conclude that A should be negative.

While exploiting the sufficient direction of hierarchical constraints is often effective, it implicitly assumes that positive evidence propagates reliably upward in the hierarchy. In practice, this assumption is frequently violated. Many real-world hierarchies exhibit strong asymmetries: subclasses may be poorly observed, noisy, or highly localized, while their superclasses are broader and easier to detect. In such cases, propagating positive evidence upward can amplify false positives, whereas exploiting the necessary direction of the constraint—suppressing a subclass when its superclass is unsupported—can lead to more reliable predictions. Importantly, the choice of which direction to exploit is not universal. As we later illustrate through synthetic and real-world examples, different datasets benefit from different orientations of the same hierarchical constraint. Existing coherent HMC models, however, hard-code a single operational semantics, typically relying only on sufficient-condition propagation. This design choice limits their flexibility and prevents them from adapting to the statistical structure of the data.

To address this limitation, in this paper we introduce the Dynamic Hierarchical Network (DHN), a novel neural architecture for HMC that jointly models both sufficient and necessary directions of hierarchical constraints within a single coherent framework. Rather than committing to one direction a priori, DHN represents each class through complementary positive and negative predictions and dynamically fuses them via a constraint-aware mechanism. This allows the model to decide, during training, how much to rely on

positive delegation from subclasses versus negative delegation from superclasses, while preserving hierarchical coherence by construction. DHN retains the desirable properties of prior coherent HMC models—most notably, guaranteed satisfaction of hierarchy constraints—while significantly extending their expressive power. The resulting architecture is scalable, depth-agnostic, and compatible with arbitrary directed acyclic hierarchies. Empirically, DHN consistently outperforms state-of-the-art HMC methods across a large collection of benchmark datasets, demonstrating that dynamically exploiting both directions of hierarchical constraints leads to more robust and accurate predictions.

Contributions.

- We identify a limitation of existing coherent HMC models due to their reliance on a single, fixed interpretation of hierarchical constraints.
- We propose Dynamic Hierarchical Network (DHN), an architecture that jointly encodes sufficient and necessary conditions while enforcing hierarchical coherence by construction.
- We introduce a dynamically constrained loss that aligns the training objective with the model's hierarchical reasoning.
- We evaluate the proposed approach on 30 real-world benchmark datasets, showing consistent improvements over strong baselines.

II. Problem Statement

A. Definition of the HMC problem

An *Hierarchical Multi-label Classification* (HMC) problem is defined as a pair $(\mathcal{P}, \Pi)$, where $\mathcal{P}$ is a multi-label classification problem and Π is a finite set of hierarchy constraints of the form $A \to B$, indicating that label A is a subclass of label B. The constraints in Π induce a directed graph with an edge from A to B for each $A \to B \in \Pi$, which is required to be acyclic.

Let $(\mathcal{P}, \Pi)$ be a HMC problem and let m be a model for $\mathcal{P}$ producing real-valued scores $m_A(x)$ for each label A. For a data point x and a constraint $A \to B \in \Pi$, we say that m commits a *hierarchical violation* if $m_A(x) > m_B(x)$. A *logical violation* occurs if m predicts A but does not predict B on x, i.e., if $m_A(x) \geq \theta$ and $m_B(x) < \theta$, θ being a user-defined threshold. If m does not commit any hierarchical violations, then it does not commit any logical violations and is therefore *coherent* with respect to Π. The converse does not necessarily hold: a model may be coherent while still committing hierarchical violations [13].

III. Method

To introduce our approach, we first consider a basic setting with two classes A and B and a single hierarchical constraint $A \to B$. This simplified case serves to motivate the proposed solution and provide intuition. We then extend the formulation to the general setting.

A. Basic Case

We consider a simplified scenario involving two classes A and B, where A is the sufficient condition for B and B is the necessary condition for A (i.e., B is the *superclass* of A, and A is the *subclass* of B). This relationship is modeled by the hierarchical constraint $A \to B$. The C-HMCNN model captures it through a Constraint Module (CM), whose outputs are defined as:

$$\begin{aligned} \text{CM}_A &= h_A, \\ \text{CM}_B &= \max(h_B, h_A), \end{aligned} \tag{1}$$

where h_A and h_B are the outputs of the bottom module h, before CM is applied. Training is carried out with a constraint-aware loss $\text{CLoss} = \text{CLoss}_A + \text{CLoss}_B$, with:

$$\begin{aligned} \text{CLoss}_A &= -y_A \ln(\text{CM}_A) - (1 - y_A) \ln(1 - \text{CM}_A), \\ \text{CLoss}_B &= -y_B \ln(\max(h_B, y_A h_A)) - (1 - y_B) \ln(1 - \text{CM}_B). \end{aligned} \tag{2}$$

Although this formulation is enough to ensure that the constraint is satisfied by design, it is not fully exploiting all the background knowledge it expresses. Indeed, the constraint $A \to B$ can equivalently be written as: $\neg B \to \neg A$. In this formulation, $\neg B$ acts as the sufficient condition for $\neg A$, which is equivalent to stating that B is the necessary condition for A. Exploiting the reversed constraint therefore amounts to suppressing the prediction of A whenever B is not predicted, that is, predicting $\neg A$ when $\neg B$ holds. Under this interpretation, we can redefine the CM as:

$$\begin{aligned} \text{CM}_A &= \min(h_B, h_A), \\ \text{CM}_B &= h_B. \end{aligned} \tag{3}$$

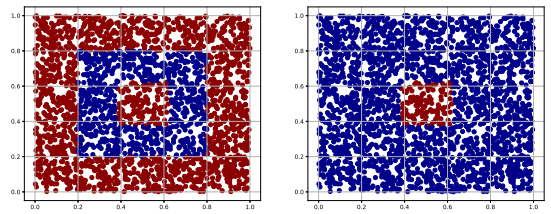
The loss function CLoss would instead be:

$$\begin{aligned} \text{CLoss}_A = &- y_A \ln(\text{CM}_A) \\ &- (1 - y_A) \ln(1 - \min(h_A, (1 - y_B) h_B + y_B)), \\ \text{CLoss}_B = &- y_B \ln(\text{CM}_B) - (1 - y_B) \ln(1 - \text{CM}_B). \end{aligned} \tag{4}$$

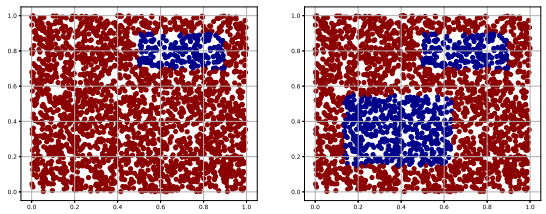
This formulation follows the same underlying intuition as C-HMCNN [14], namely that hierarchical constraints should not only be enforced at the level of predictions, but also respected by the gradient flow during training. In particular, the loss must be designed so that gradients propagate in directions that are consistent with the intended semantics of the constraint, thereby reinforcing coherent behavior rather than correcting violations post hoc.

In this case, the constraint $A \to B$ is interpreted through its necessary-condition semantics (i.e., as $\neg B \to \neg A$): when B is not supported, the model should be discouraged from predicting A. Accordingly, the loss term for class A is modified so that, when $y_A = 0$ and $y_B = 0$, the gradient penalizes high confidence in A proportionally to the confidence assigned to B. This ensures that negative evidence for B propagates downward to suppress A, aligning gradient updates with the contrapositive constraint $\neg B \to \neg A$.

Consider a data point with ground-truth labels $y_A = 0$ and $y_B = 1$, which is consistent with the hierarchy constraint



(a) Overlap dataset (left class A, right class B)



(b) Not overlap dataset (left class A, right class B)

Fig. 1: The two datasets used for the basic case study. The blue points represent samples belonging to the classes reported by the plot captions while the red points represent the samples not belonging to them.

$A \rightarrow B$. Suppose the model assigns scores such that $h_A > h_B$, resulting in a hierarchical violation. In this setting, the model should be penalized for overpredicting A, but this penalty should not depend on the (incorrectly low) prediction of B, as B is known to be present.

The masking term $(1 - y_B)h_B + y_B$ ensures this behavior. When $y_B = 1$, the expression evaluates to 1, and the loss term for class A reduces to a standard Binary Cross-Entropy (BCE) penalty based solely on h_A. As a result, gradients correctly suppress h_A without coupling the update to h_B. Without this masking, A would be penalized as a function of h_B, producing misleading gradients that attribute the violation to the superclass rather than to the overconfident subclass. The proposed formulation therefore ensures that gradient updates remain aligned with both the hierarchy semantics and the available supervision.

Depending on the structure of the data hierarchy, either the original constraint $A \rightarrow B$ or its contrapositive form $\neg B \rightarrow \neg A$ may be more suitable for enforcing hierarchical consistency. To investigate this hypothesis, we introduce two synthetic datasets illustrated in Fig. 1. In both cases, a sample space of dimension $N = 2$ was assumed.

Datasets: Both datasets consist of $5,000$ samples drawn uniformly from $[0, 1]^2$ and split evenly into training and test sets. The *overlap dataset* (Fig. 1a) is defined over a two-dimensional domain where class B occupies almost the entire space except for a central region, while class A forms a ring surrounding this region. In this setting, false positive predictions of A inside the region where B is absent constitute the main source of error. Enforcing the contrapositive constraint $\neg B \rightarrow \neg A$ is therefore more suitable, as it suppresses predictions of A in areas where B is confidently inactive. The *not-overlap dataset* (Fig. 1b) is defined over a two-dimensional domain where class A occupies a localized region that is fully contained within the support of class B. In this setting, the main source of error arises in modeling non-contiguous shapes for class B. Enforcing the forward implication $A \rightarrow B$ is therefore more effective, as it propagates positive evidence from A to B within the shared support.

These hypotheses were tested using three models on both datasets to compare the two constraint formulations. The networks used a single hidden layer with 6 neurons, *tanh* activation and a *sigmoid* output nonlinearity. All models were trained for 20,000 epochs using the Adam optimizer [15].

The first model serves as a baseline. It consists of a simple feedforward neural network h with one output per class, trained using the standard BCE loss:

$$\mathcal{L} = -\sum_{c \in \{A,B\}} \left[y_c \ln(h_c) + (1 - y_c) \ln(1 - h_c) \right], \quad (5)$$

where h_A and h_B are the outputs of h for classes A and B, respectively. Since no post-processing is applied, h does not explicitly enforce the hierarchy and may produce constraint-inconsistent predictions.

For the second approach, we implement a model denoted as h_{pos}, which adopts the forward implication ($A \rightarrow B$, consistent with C-HMCNN). The CM outputs and the associated loss function are defined in Eqs. (1) and (2). No post-processing is required, and coherence is ensured by construction [7], [14].

For the last approach, we implement h_{neg}, which encodes the contrapositive constraint $\neg B \rightarrow \neg A$. The corresponding module outputs and loss function are given in Eqs. (3) and (4). As with the second approach, coherence is enforced within the architecture.

As shown in Fig. 2 and Fig. 3, for both h_{pos} and h_{neg}, the first column shows the decision before constraint enforcement, and the second shows the final constrained output. Darker blue regions represent higher confidence in the positive class, while red denotes low-confidence or suppressed areas. In the overlap dataset, h_{neg} effectively suppresses false positives for class A within the central hole by applying the contrapositive constraint $\neg B \rightarrow \neg A$, sharpening the ring boundary. In the not-overlap dataset, h_{pos} improves performance by delegating the small top-right region of class B to class A, and the forward implication $A \rightarrow B$ helps recover the full decision region through constraint composition.

A natural desideratum is a single model that can benefit from both constraint orientations, achieving strong performance on both datasets without changing how the constraint is encoded. Accordingly, this work extends the single-path setting with a formulation that captures the advantages of both h_{pos} and h_{neg} within one architecture.

To achieve this, each original class is represented by two derived labels that explicitly encode its predicted and non-predicted states. In the basic case, A is replaced by A^+ (positive A) and A^- (negative A), and similarly for B. Under

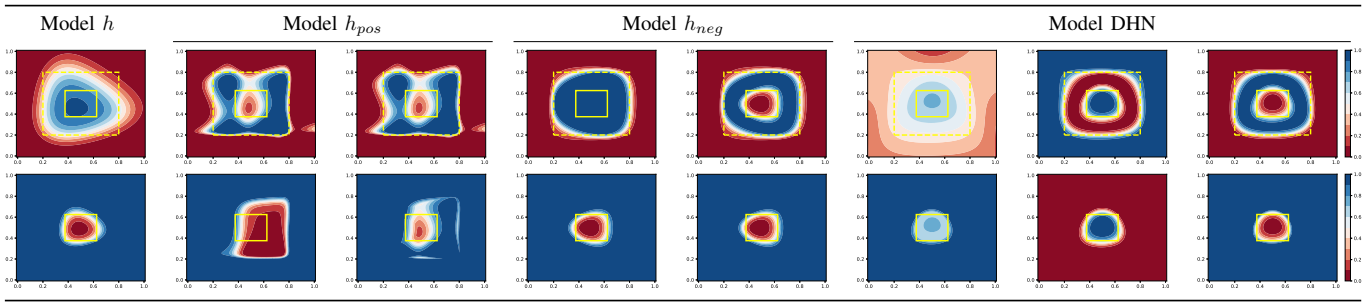


Fig. 2: Decision boundaries on the overlap dataset, where blue (red) regions represent positive (negative) predictions. Columns (left to right): baseline model h; h_{pos} before/after CM; h_{neg} before/after CM; DHN outputs from the positive path, negative path, and FCM. Rows correspond to class A (top) and class B (bottom)

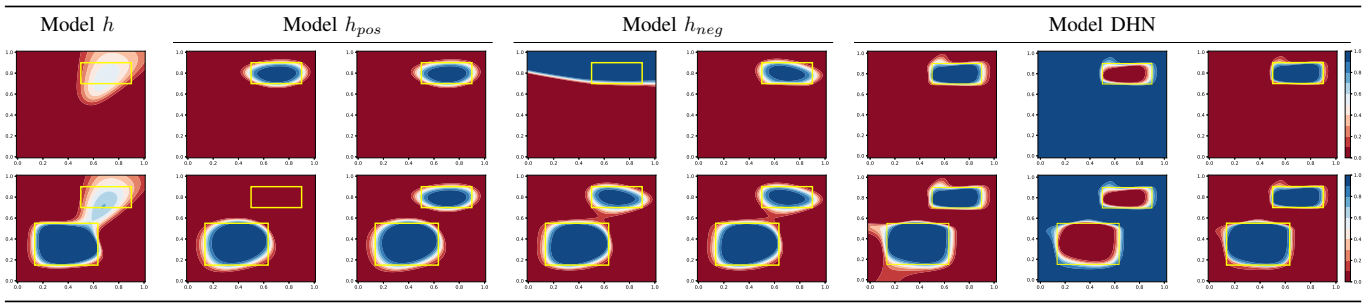


Fig. 3: Decision boundaries on the not-overlap dataset. Layout follows the same format as in Fig. 2

this representation, the $A \to B$ constraint is enforced via the following constraints:

$$A^+ \to B^+, \qquad B^- \to A^-, \tag{6}$$

where the first equation captures the sufficient-condition direction and the second one captures the necessary-condition information via the contrapositive.

We then introduce our new model Dynamic Hierarchical Network (DHN), which retains the two-stage structure of [14]: a bottom module h followed by a CM. The bottom module outputs four scores in $[0, 1]$, corresponding to A^+, B^+, A^-, and B^-. The CM is organized into two sequential sub-modules, an Intermediate Constraint Module (ICM) and a Final Constraint Module (FCM). The ICM takes the outputs of h and produces intermediate scores of the same dimensionality while enforcing Eq. (6). The ICM outputs are defined as:

$$\begin{aligned} \text{CM}_{A^+} &= h_{A^+}, \quad \text{CM}_{A^-} = \max(h_{A^-}, h_{B^-}), \\ \text{CM}_{B^+} &= \max(h_{A^+}, h_{B^+}), \quad \text{CM}_{B^-} = h_{B^-}, \end{aligned} \tag{7}$$

followed by the FCM combines the positive and negative path outputs from the ICM using a mixing weight $\alpha \in [0, 1]$:

$$\begin{aligned} \text{CM}_A &= \alpha \text{CM}_{A^+} + (1-\alpha)(1 - \text{CM}_{A^-}), \\ \text{CM}_B &= \alpha \text{CM}_{B^+} + (1-\alpha)(1 - \text{CM}_{B^-}). \end{aligned} \tag{8}$$

Since the ICM ensures $\text{CM}_{B^+} \geq \text{CM}_{A^+}$ and $\text{CM}_{A^-} \geq \text{CM}_{B^-}$, the final outputs satisfy $\text{CM}_B \geq \text{CM}_A$, thus preserving the hierarchy constraint by construction.

We also introduce a dynamically constrained loss defined as $\text{DCLoss} = \text{DCLoss}_A + \text{DCLoss}_B$, where:

$$\begin{aligned} \text{DCLoss}A = &- y_A(\alpha \ln(h_{A^+}) + (1-\alpha)\ln(1 - \text{CM}_{A^-})) \\ &- (1 - y_A)(\alpha \ln(1 - h_{A^+}) \\ &+ (1-\alpha)\ln(\max(h_{A^-}, (1-y_B)h_{B^-})))), \\ \text{DCLoss}B = &- y_B(\alpha \ln(\max(y_A h_{A^+}, h_{B^+})) \\ &+ (1-\alpha)\ln(1 - h_{B^-})) \\ &- (1 - y_B)(\alpha \ln(1 - \text{CM}_{B^+}) \\ &+ (1-\alpha)\ln(h_{B^-})). \end{aligned} \tag{9}$$

Here, y_A and y_B are the ground truth values for the original labels A and B, and α is the same mixing weight used for the FCM and was set to 0.5 during this phase.

Considering the results of DHN on both datasets (Figs. 2–3), the first column reports the output of the positive path (CM^+), the second column corresponds to the negative path (CM^-), and the third column shows the final prediction after fusion by the FCM. Since the negative path models the absence of a class, the output predictions are complementary to the ground-truth labels. Compared to the single-constraint models h_{pos} and h_{neg}, both CM^+ and CM^- exhibit more interpretable and consistent decision boundaries, indicating that DHN effectively learns and exploits both constraint directions rather than merely combining them post hoc.

B. General Case

We now consider a general HMC problem $(\mathcal{P}, \Pi)$ [7] with constraints Π. Similarly to [7], [14], given a class A, we call

$\mathcal{D}_A$ the set of subclasses (or descendants) of A according to Π (including A itself). Analogously, we define $\mathcal{F}_A$ as the set of superclasses (or ancestor) of class A according to Π (including A itself).

Let $(\mathcal{P}, \Pi)$ be a HMC problem and let h be a model for $\mathcal{P}$ used as the bottom module of DHN. Let A be a class, $C \in \mathcal{D}_A$ one of its subclasses, and $B \in \mathcal{F}_A$ one of its superclasses. For a data point, DHN is said to *delegate the prediction of A to C* if $\text{CM}_{A^+} = h_{C^+}$ and $h_{C^+} > h_{A^+}$. Analogously, DHN is said to *delegate the prediction of A to B* (equivalently, delegate the prediction of $\neg A$ to $\neg B$) if $\text{CM}_{A^-} = h_{B^-}$ and $h_{B^-} > h_{A^-}$.

The ICM implements both forms of delegation by computing:

$$\text{CM}_{A^+} = \max_{D \in \mathcal{D}_A} (h_{D^+}), \quad \text{CM}_{A^-} = \max_{F \in \mathcal{F}_A} (h_{F^-}), \tag{10}$$

while the FCM blends the two paths according to Eq. (8).

For each class A, the number of sequential operations performed by CM_{A^+}, CM_{A^-} and CM_A is independent from its depth in the hierarchy, making DHN a scalable model on GPUs. The constraint modules additionally ensure that the DHN outputs are coherent by construction.

Theorem III.1. *Let $(\mathcal{P}, \Pi)$ be a HMC problem with class set $\mathcal{A} = \{A_1, \ldots, A_n\}$, and let h be a neural network model for $\mathcal{P}$ producing outputs $h_{A_i^+}, h_{A_i^-} \in [0,1]$ for each class $A_i \in \mathcal{A}$. Let $\text{CM}_{A_i^+}$ and $\text{CM}_{A_i^-}$ be defined as in Eq. (10), and let the final outputs CM_{A_i} be obtained as in Eq. (8). Then, DHN does not commit any hierarchical violations.*

Proof. From [14] and Eq. (10), the outputs of the intermediate constraint module satisfy, for every constraint $A \to B \in \Pi$,

$$\text{CM}_{A^+} \leq \text{CM}_{B^+} \qquad \text{and} \qquad \text{CM}_{A^-} \geq \text{CM}_{B^-}.$$

The final constraint module computes, for each class $A \in \mathcal{A}$,

$$\text{CM}_A = \alpha \text{CM}_{A^+} + (1-\alpha)(1 - \text{CM}_{A^-}), \quad \alpha \in [0,1].$$

Then, for every $A \to B \in \Pi$,

$$\begin{aligned} \text{CM}_B - \text{CM}_A &= \alpha(\text{CM}_{B^+} - \text{CM}_{A^+}) \\ &\quad + (1-\alpha)(\text{CM}_{A^-} - \text{CM}_{B^-}) \\ &\geq 0, \end{aligned}$$

where the inequality follows from the ICM constraints and $\alpha \in [0,1]$. Hence, $\text{CM}_A \leq \text{CM}_B$ for all $A \to B \in \Pi$, and thus DHN does not commit hierarchical violations. □

Corollary III.1.1. *Under the above definitions, DHN is coherent with respect to Π and commits no logical violations.*

We finally define $\text{DCLoss} = \sum_{A \in \mathcal{A}} \text{DCLoss}_A$, where DCLoss_A for a class $A \in \mathcal{A}$ is defined as:

$$\begin{aligned} \text{DCLoss}_A = &- y_A(\alpha \ln(\max_{D \in \mathcal{D}_A} (y_{D^+} h_{D^+})) \\ &+ (1-\alpha)\ln(1 - \text{CM}_{A^-})) \\ &- (1-y_A)(\alpha \ln(1 - \text{CM}_{A^+}) \\ &+ (1-\alpha)\ln(\max_{F \in \mathcal{F}_A}((1-y_F)h_{F^-}))). \end{aligned} \tag{11}$$

TABLE I: Summary of the 20 real-world datasets most commonly used for HMC problems (set D_1). *N* represents the number of features, *n* the number of classes, and *Train*, *Val*, *Test* the number of data-points for each dataset split.

Taxonomy	*Dataset*	*N*	*n*	*Train/Val/Test*
FUNCAT (FUN)	Cellcycle	77	499	1625/848/1281
	Derisi	63	499	1605/842/1272
	Eisen	79	461	1055/529/835
	Expr	551	499	1636/849/1288
	Gasch1	173	499	1631/846/1281
	Gasch2	52	499	1636/849/1288
	Seq	478	499	1692/876/1332
	Spo	80	499	1597/837/1263
Gene Ontology (GO)	Cellcycle	77	4122	1625/848/1281
	Derisi	63	4116	1605/842/1272
	Eisen	79	3570	1055/529/835
	Expr	551	4128	1636/849/1288
	Gasch1	173	4122	1631/846/1281
	Gasch2	52	4128	1636/849/1288
	Seq	478	4130	1692/876/1332
	Spo	80	4166	1597/837/1263
Tree	Diatoms	371	398	1085/464/1054
	Enron	1000	56	692/296/660
	Imclef07a	80	96	7000/3000/1006
	Imcllef07d	80	46	7000/3000/1006

TABLE II: Summary of the ontology datasets in Set D_2. *N* represents the number of features, *n* the number of classes, *data* the total number of data points, and *depth* the depth of the DAG.

Taxonomy	*Dataset*	*N*	*n*	*data*	*depth*
DBPedia (DAG)	Comedy	384	395	8333	6
	Culture	384	1429	46495	5
	Energy	384	879	25649	8
	Engineering	384	587	22735	7
	Information	384	754	24781	7
	Law	384	958	61974	10
	Main	384	147	15310	7
	Mathematics	384	407	17163	9
	People	384	177	7983	7
	Philosophy	384	305	10930	7

IV. Experimental Analysis

A. Datasets

We consider two collections of benchmarks. The first set D_1 comprises 20 real-world datasets that are widely adopted for the evaluation of HMC methods [5], [7], [9], [13], [14], [16]. This includes 16 functional genomics datasets [17], two medical imaging datasets [4], one micro-algae imaging dataset [18], and one text categorization dataset [19]. Table I summarizes the characteristics of D_1. These benchmarks are particularly demanding due to the limited number of training instances and they are subject to a large variation both in the number of features and classes.

Set D_2 includes 4 Ontologue datasets presented in [20], along with 6 additional datasets extracted from DBPedia [21] following the authors' approach, which are intended to overcome the issue of class imbalance. In Table II we report the details of D_2.

We followed the work of [7] and applied the same preprocessing to all datasets: 1) all categorical features were transformed with one-hot encoding, 2) the missing values were replaced by their mean (numerical features) or by a vector of zeros (categorical features), and 3) all features were standardized.

B. Method

The bottom module h is implemented as a 3-layer feed-forward network with ReLU activations. Training minimizes the proposed objective using Adam and early stopping with patience 20. For the final test stage, the model is retrained on the union of training and validation data for the same number of epochs.

To reduce search complexity while maintaining consistent training conditions, we fix the batch size to 4, dropout to 0.7, and weight decay to 10^{-5} for all datasets. Hyperparameters are selected on the validation set via Bayesian optimization [22]–[24], focusing on the learning rate, the mixing coefficient α used in the FCM and DCLoss, and the hidden dimension.

For real-world benchmarks in set D_1, the learning rate is sampled uniformly from $[10^{-5}, 10^{-3}]$ and α from $[0, 1]$. The hidden dimension is searched on a discrete grid from 50 to $10,000$ with step size 50. For the Ontologue benchmarks in D_2, we restrict the learning rate to 10^{-5}, sample α from $[0, 1]$, and search the hidden dimension on a discrete grid from $5,000$ to $10,000$ with step size 100. Details of the dataset, model architecture and hyperparameter choices are available in the public repository[1]. All experiments were conducted on an NVIDIA A100 GPU with 80 GB memory.

C. Results

In this section, the performance of DHN is reported and compared with representative state-of-the-art HMC approaches, including C-HMCNN [14], HMC-LMLP [12], Clus-Ens [25], HMCN-R and HMCN-F [13]. Performance is measured using the area under the precision and recall curve $AU(\overline{PRC})$, calculated as the average precision from prediction scores using scikit-learn library [26], [27]. In addition, C-HMCNN-Min was introduced as an ablation that exploits only the necessary direction of hierarchy constraints. This variant corresponds to the h_{neg} branch introduced in Section III-A.

After hyperparameter tuning, each dataset is evaluated over ten independent GPU runs. For D_1, results for all baselines other than DHN are taken from prior work [13]. In particular, C-HMCNN-Min model follows the training protocol of C-HMCNN as reported in [14].

For D_2, we train DHN, C-HMCNN, and C-HMCNN-Min under the same training strategy described in the previous section to ensure a fair comparison. The result of HMC-LMLP was taken from the Surj study [20]. We do not report HMCN-R and HMCN-F on D_2 because the implementations are not publicly available. Clus-Ens results are unavailable for the Culture dataset due to prohibitive computational requirements.

[1]https://github.com/GelsominaDiPalma/Dynamic-Hierarchical-Multi-label-Classification-Networks

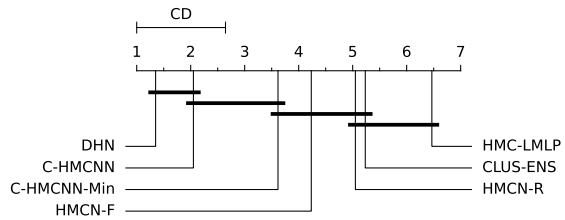


Fig. 4: Critical Difference (CD) diagram resulting from the Nemenyi test.

Table III reports the $AU(\overline{PRC})$ scores of all models across 30 benchmark datasets. DHN achieves the best performance on 19 datasets, ties on 6, and ranks second on 3, significantly outperforming all other state-of-the-art models. It also attains the lowest (best) average rank of 1.35, as shown in the critical difference diagram (Fig. 4). The standard deviation of DHN across datasets lies in the range $[0.5 \times 10^{-3}, 7 \times 10^{-3}]$, indicating strong stability.

A Friedman test across all 30 datasets yields $\chi^2 = 130.52$ with $p = 1.001 \times 10^{-25}$, rejecting the null hypothesis of performance equivalence. The post-hoc Nemenyi test (visualized in Fig. 5) confirms that DHN is statistically superior to all methods except C-HMCNN.

To further validate the impact of the architectural enhancements introduced in DHN, we perform pairwise Wilcoxon signed-rank tests against both C-HMCNN and C-HMCNN-Min. These models serve as ablative baselines isolating the effect of using only positive or negative delegation. The results yield highly significant p-values:

- DHN vs. C-HMCNN: $p = 6.70 \times 10^{-5}$
- DHN vs. C-HMCNN-Min: $p = 2.53 \times 10^{-6}$

A noteworthy pattern in Table III is that C-HMCNN-Min consistently under performs C-HMCNN. Since it only exploits the necessary direction of the constraints via a minimum operator as in Eq. (3), it tends to be more conservative and thus yields lower $AU(\overline{PRC})$ on average. Nonetheless, its occasional proximity to or above with C-HMCNN is consistent with our earlier claim that the most beneficial constraint orientation is data dependent, and it remains a useful ablation isolating purely negative delegation.

These findings collectively demonstrate that DHN not only establishes state-of-the-art performance across diverse HMC tasks but also exhibits robust statistical superiority. The improvements are consistent across all datasets and remain significant even when compared to structurally similar variants, confirming the efficacy of our proposed architectural modifications.

D. Related Work

Recent advances in HMC move toward structure-aware architectures that encode hierarchical relations directly in the model. For instance, HyperIM [28] embeds documents and labels jointly in hyperbolic space to capture tree-like relations.

TABLE III: Comparison of DHN with other state of the art models. The performance of each system is obtained as the $AU(\overline{PRC})$ obtained in the test set. The best results are showed in bold, and we underline when DHN scores the second best result.

Dataset	DHN	C-HMCNN	C-HMCNN-Min	HMC-LMLP	Clus-Ens	HMCN-R	HMCN-F
Cellcycle_FUN	**0.256**	0.255	0.249	0.207	0.227	0.247	0.252
Derisi_FUN	**0.198**	0.195	0.195	0.182	0.187	0.189	0.193
Eisen_FUN	**0.307**	0.306	0.301	0.245	0.286	0.298	0.298
Expr_FUN	0.301	**0.302**	0.294	0.242	0.271	0.300	0.301
Gasch1_FUN	0.285	**0.286**	0.280	0.235	0.267	0.283	0.284
Gasch2_FUN	**0.260**	0.258	0.251	0.211	0.231	0.249	0.254
Seq_FUN	**0.295**	0.292	0.282	0.236	0.284	0.290	0.291
Spo_FUN	**0.216**	0.215	0.214	0.186	0.211	0.210	0.211
Cellcyle_GO	**0.415**	0.413	0.406	0.361	0.387	0.395	0.400
Derisi_GO	**0.371**	0.370	0.368	0.343	0.361	0.368	0.369
Eisen_GO	**0.456**	0.455	0.450	0.406	0.433	0.435	0.440
Expr_GO	0.448	0.447	0.385	0.373	0.422	0.450	**0.452**
Gasch1_GO	**0.440**	0.436	0.430	0.380	0.415	0.416	0.428
Gasch2_GO	0.416	0.414	0.371	0.408	0.395	0.463	**0.465**
Seq_GO	**0.447**	0.446	0.378	0.370	0.438	0.443	**0.447**
Spo_GO	**0.382**	**0.382**	0.376	0.342	0.371	0.375	0.376
Diatoms	**0.768**	0.758	0.728	-	0.501	0.514	0.530
Enron	**0.764**	0.756	0.752	-	0.696	0.710	0.724
Imclef07a	**0.958**	0.956	0.951	-	0.803	0.904	0.950
Imclef07d	**0.928**	0.927	**0.928**	-	0.881	0.897	0.920
Comedy	**0.904**	**0.904**	0.893	0.627	0.713	-	-
Culture	**0.702**	0.700	0.685	-	-	-	-
Energy	0.858	**0.859**	0.846	-	0.653	-	-
Engineering	**0.839**	0.838	0.825	0.730	0.695	-	-
Information	**0.795**	**0.795**	0.778	-	0.558	-	-
Law	**0.843**	**0.843**	0.832	0.671	0.693	-	-
Main	**0.898**	0.895	0.888	0.663	0.734	-	-
Mathematics	**0.910**	0.909	0.900	-	0.726	-	-
People	**0.916**	0.911	0.908	-	0.792	-	-
Philosophy	**0.742**	0.735	0.722	-	0.577	-	-
AVG RANK	1.35	2.05	3.62	6.47	5.05	4.23	5.23

Surj [29] separates label embedding from instance classification, enabling zero-shot generalization and minimizing violations under a global hierarchy metric.

Building on this, a variety of neural models have incorporated hierarchy more explicitly into their training objectives. HLSTM [30] introduces local and global penalties to reduce constraint violations. SPUR [31] uses dual encoders and self-paced curriculum learning to handle label sparsity and depth imbalance. In text classification, HGBL [32] combines BERT encoders, graph-aware label modeling, and contrastive objectives. More recent formulations, such as HCL-MTC [33] and HMG [34], push hierarchy awareness into contrastive and generative paradigms, the former defines a hierarchical contrastive loss over label graphs, while the latter casts HMC as constrained label sequence generation using level-specific Transformer decoding masks.

V. Conclusion

This paper proposed DHN for HMC problems, introducing a dual-path architecture that enforces hierarchical consistency by design while jointly modeling both sufficient and necessary conditions. Unlike previous constraint-aware approaches that rely solely on positive evidence propagation, DHN integrates negative evidence through a symmetric, dual-branch formulation and fuses the two via a principled constraint module. Methodologically, DHN generalizes coherent HMC frameworks by extending constraint reasoning to paired positive/negative outputs and supports scalable, depth-agnostic inference. The dynamically constrained loss further aligns learning objectives with the model's architecture, ensuring effective optimization across both constraint directions.

Extensive empirical evaluation across 30 benchmark datasets demonstrates that DHN consistently outperforms rep-

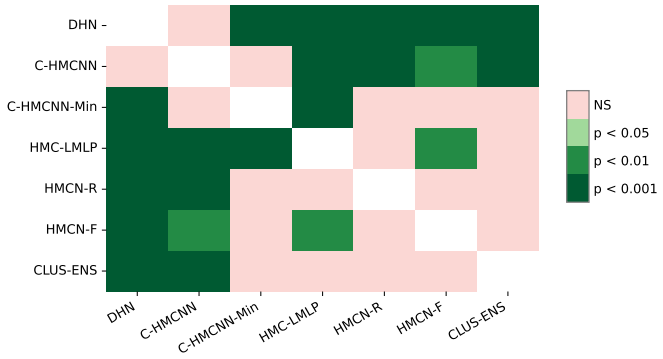


Fig. 5: Heatmap of the p values resulting from the Nemenyi test. The NS value in the legend indicates a non-significant p-value.

resentative state-of-the-art baselines. Statistical analyses further confirm that these gains are significant, including under stringent non-parametric tests and direct comparisons against structurally related ablations. These results establish DHN as a principled and effective solution for HMC. By unifying positive and negative hierarchical reasoning within a single coherent architecture, DHN advances the state of the art in constraint-aware learning and provides a strong foundation for reliable prediction in complex hierarchical domains.

REFERENCES

[1] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[2] J. Bogatinovski, L. Todorovski, S. Džeroski, and D. Kocev, "Comprehensive comparative study of multi-label classification methods," *Expert Systems with Applications*, vol. 203, p. 117215, 2022.

[3] P. Pant, A. Sai Sabitha, T. Choudhury, and P. Dhingra, "Multi-label classification trending challenges and approaches," *Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS 2018*, pp. 433–444, 2018.

[4] I. Dimitrovski, D. Kocev, S. Loskovska, and S. Džeroski, "Hierarchical annotation of medical images," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2436–2449, 2011.

[5] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Machine learning*, vol. 73, pp. 185–214, 2008.

[6] Y. Ma, X. Liu, L. Zhao, Y. Liang, P. Zhang, and B. Jin, "Hybrid embedding-based text representation for hierarchical multi-label text classification," *Expert Systems with Applications*, vol. 187, p. 115905, 2022.

[7] E. Giunchiglia and T. Lukasiewicz, "Multi-label classification neural networks with hard logical constraints," *Journal of Artificial Intelligence Research*, vol. 72, pp. 759–818, 2021.

[8] L. Masera and E. Blanzieri, "Awx: An integrated approach to hierarchical-multilabel classification," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 322–336.

[9] W. Bi and J. T. Kwok, "Multi-label classification on tree-and dag-structured hierarchies," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 17–24.

[10] R. Cerri, R. C. Barros, and A. C. De Carvalho, "Hierarchical multi-label classification using local neural networks," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 39–56, 2014.

[11] G. Obozinski, G. Lanckriet, C. Grant, M. I. Jordan, and W. S. Noble, "Consistent probabilistic outputs for protein function prediction," *Genome Biology*, vol. 9, no. 1, pp. 1–19, 2008.

[12] R. Cerri, R. C. Barros, A. C. PLF de Carvalho, and Y. Jin, "Reduction strategies for hierarchical multi-label classification in protein function prediction," *BMC bioinformatics*, vol. 17, no. 1, pp. 1–24, 2016.

[13] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *International conference on machine learning*. PMLR, 2018, pp. 5075–5084.

[14] E. Giunchiglia and T. Lukasiewicz, "Coherent hierarchical multi-label classification networks," *Advances in neural information processing systems*, vol. 33, pp. 9662–9673, 2020.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] F. K. Nakano, M. Lietaert, and C. Vens, "Machine learning for discovering missing or wrong protein function annotations: a comparison using updated benchmark datasets," *BMC bioinformatics*, vol. 20, no. 1, p. 485, 2019.

[17] A. Clare and R. D. King, "Predicting gene function in saccharomyces cerevisiae," *Bioinformatics-Oxford*, vol. 19, no. 2, pp. 42–49, 2003.

[18] I. Dimitrovski, D. Kocev, S. Loskovska, and S. Džeroski, "Hierarchical classification of diatom images using ensembles of predictive clustering trees," *Ecological Informatics*, vol. 7, no. 1, pp. 19–29, 2012.

[19] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European conference on machine learning*. Springer, 2004, pp. 217–226.

[20] S. Yang, B. Herman, and B. Howe, "Ontologue: Declarative benchmark construction for ontological multi-label classification," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 463–22 476, 2022.

[21] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *international semantic web conference*. Springer, 2007, pp. 722–735.

[22] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019.

[23] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: a python library for model selection and hyperparameter optimization," *Computational Science & Discovery*, vol. 8, no. 1, p. 014008, 2015.

[24] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[25] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev, and S. Džeroski, "Predicting gene function using hierarchical multi-label decision tree ensembles," *BMC bioinformatics*, vol. 11, pp. 1–14, 2010.

[26] E. Zhang and Y. Zhang, *Average Precision*. Boston, MA: Springer US, 2009, pp. 192–193. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_482

[27] M. Zhu, "Recall, precision and average precision," *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, vol. 2, no. 30, p. 6, 2004.

[28] B. Chen, X. Huang, L. Xiao, Z. Cai, and L. Jing, "Hyperbolic interaction model for hierarchical multi-label classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 7496–7503.

[29] S. T. Yang and B. Howe, "Surj: Ontological learning for fast, accurate, and robust hierarchical multi-label classification," in *Companion Proceedings of the Web Conference 2022*, 2022, pp. 1106–1114.

[30] W. Qi and C. Chelmis, "Hybrid loss for hierarchical multi–label classification network," in *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 2023, pp. 819–828.

[31] Z. Yuan, H. Liu, H. Zhou, D. Zhang, X. Zhang, H. Wang, and H. Xiong, "Self-paced unified representation learning for hierarchical multi-label classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 15, 2024, pp. 16 623–16 632.

[32] C. Zhang, L. Dai, C. Liu, and L. Zhang, "Hgbl: A fine granular hierarchical multi-label text classification model," *Neural Processing Letters*, vol. 57, no. 1, p. 1, 2024.

[33] W. Zhang, Y. Jiang, Y. Fang, and S. Pan, "Hierarchical contrastive learning for multi-label text classification," *Scientific Reports*, vol. 15, no. 1, p. 14101, 2025.

[34] L. Chen, W. Wang, W. Wu, and H. Zhong, "Hierarchical multi-label generation with probabilistic level-constraint," *arXiv preprint arXiv:2505.03775*, 2025.