MODELING LABEL SPACE INTERACTIONS IN MULTI-LABEL CLASSIFICATION USING BOX EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-label classification is a challenging structured prediction task in which a set of output class labels are predicted for each input. Real-world datasets often have natural or latent taxonomic relationships between labels, making it desirable for models to employ label representations capable of capturing such taxonomies. Most existing multi-label classification methods do not do so, resulting in label predictions that are incoherent w.r.t the taxonomic constraints, thus failing to accurately represent the fundamentals of the problem setting. In this work we introduce the multi-label box model (MBM), a multi-label classification method that combines the encoding power of neural networks with the inductive bias and probabilistic semantics of box embeddings (Vilnis et al., 2018), which can be understood as trainable Venn-diagrams based on hyper-rectangles. Representing labels by boxes rather than vectors, MBM is able to capture taxonomic relations among labels. Furthermore, since box embeddings allow these relations to be learned by stochastic gradient descent from data, and to be read as calibrated conditional probabilities, our model provides a high degree of interpretability. This interpretability also facilitates the injection of partial information about label-label relationships into model training, to further improve its consistency. We provide theoretical grounding for our method and show experimentally the model's ability to learn the true latent taxonomic structure from data. Through extensive empirical evaluations on both small and large-scale multi-label classification datasets, we show that MBM can significantly improve taxonomic consistency while preserving or surpassing state-of-the-art predictive performance.

1 INTRODUCTION

Multi-label classification is a machine learning task in which an input is associated with multiple categories. Many real-world multi-label classification datasets in modalities such as text categorization (Lewis et al., 2004), image classification (Lin et al., 2015; Krishna et al., 2016), entity typing (Murty et al., 2018; Onoe et al., 2021), functional genomics (Barutcuoglu et al., 2006; Clare, 2003), and so on, have a rich inter-dependent label structure that can be expressed using a taxonomy graph or a hierarchy. To be useful in practice, a model should produce predictions that are coherent with respect to the label taxonomy. For example, if a book is classified as *drama*, it should also be classified as *fiction* according to the label taxonomy of book genres in the left-hand side of Figure 1. More formally, we are given a label taxonomy in the form of a directed acyclic graph $G = (\mathcal{L}, \mathcal{T})$, where $(a, b) \in \mathcal{T}$ if and only if a is a parent of b in the taxonomy. A model will assign scores $s_\ell(x)$ for each label $\ell \in \mathcal{L}$. We say the scores are *coherent* with respect to the taxonomy if, for all edges $(a, b) \in \mathcal{T}$, $s_a(x) \geq s_b(x)$, and the *model* is consistent if this is the case for all inputs x. In the case of the book genre classification example, this implies that s_{fiction} must be greater than or equal to s_{drama}, regardless of the input.

The problem of producing coherent predictions for multi-label classification has garnered a lot of attention in the machine learning literature (Wehrmann et al., 2018c; Giunchiglia & Lukasiewicz, 2020; Murty et al., 2018; Cerri et al., 2014). Most methods that proposed to improve the coherence in predictions require the complete label taxonomy at inference time, and occasionally at training time as well, making these models hard to scale to large label spaces (Giunchiglia & Lukasiewicz, 2020; Wehrmann et al., 2018c). This brings forth a question: Can we utilize representation learning to model the label-label relationships implicitly in the embedding space?

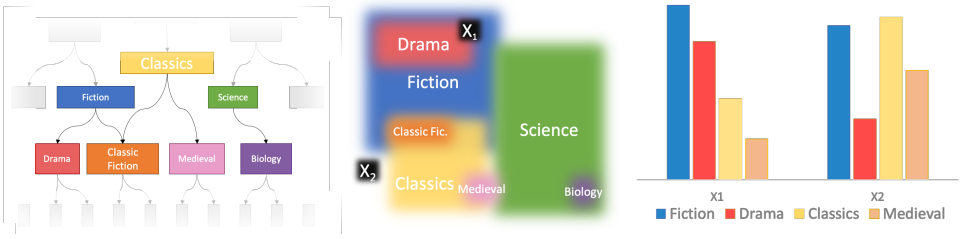


Figure 1: Left: A label taxonomy represented as a DAG. Center: A 2-dimensional box embedding of this taxonomy. Right: Scores assigned by the box embedding model to each label for two inputs.

Vilnis et al. (2018) show that probabilistic box embeddings, which represent concepts as high dimensional hyper-rectangles, can embed DAGs efficiently using the explicit information about the edges. Box embeddings represent edges by box-box containment as shown in Figure 1b. Representing the input and output labels in the same geometric space of boxes allows the multi-label taxonomy to be learned without an explicit taxonomic training signal. Moreover, there exists a large space of possible configurations that represent the same taxonomy, and if the label embeddings in the model gets close to any such configuration, then the model will always produce classifications that are coherent w.r.t the taxonomy, regardless of the input. We show through empirical evidence this is the case, and provide a formal proof for latter.

In this work, we propose the multi-label box model (MBM) that utilizes the geometry and probabilistic semantics of box embeddings to model label-label interactions in multi-label classification. MBM represents labels as boxes using free parameters and uses a deep neural network to embed the inputs in the same space. We perform coherence analysis of the model using two measures and show that MBM not only achieves state-of-the-art predictive performance but it also significantly improves the coherence of predicted scores w.r.t latent label taxonomy. Our analysis further shows that it is possible to retrieve the latent label-label relationships solely by analysing the learnt label representations inside the MBM, endowing the model with high degree of interpretability. Finally, we also present a way to utilize the interpretability of MBM to inject partial information about label-label relationships into the model thereby improving the coherence even further.

2 Related Work

Multi-label classification tasks that exhibit strong label space structure in the form of explicit label taxonomy are termed hierarchical multi-label classification (HMLC) in machine learning literature. Most approaches for such tasks make use of the complete hierarchy at training time. These approaches can be categorized into two buckets (Silla & Freitas, 2010): (1) *Local approaches* that focus on local information for each label or clusters of labels in the hierarchy and classify them independently (Cerri et al., 2014; Huang et al., 2019), and (2) *Global approaches* that treat the problem as a structured classification task and take global interactions into account (Belanger & McCallum, 2016). In the most general setting, however, both local and global interactions between labels exist. The recent advances in deep learning (Wehrmann et al., 2018a) propose a specialized neural network architecture called Hierarchical Multi-Label Classification Network (HMCN-R and HMCN-F) that takes into account both local and global interactions by creating an ensemble of classifiers that can be trained using end-to-end gradient based training. However, HMCN does not try to enforce coherence strongly, focusing solely on predictive performance. In order to improve prediction coherence, recent works employ special loss functions on top of a neural network classifier to enforce coherence w.r.t the label taxonomy (Murty et al., 2018; Giunchiglia & Lukasiewicz, 2020). While effective, these approaches still use the label taxonomy explicitly, making them difficult to scale to very large label spaces.

Recent advances in representation learning provide various methods to embed large graphs and taxonomies parsimoniously in non-euclidean spaces (Nickel & Kiela, 2017; Vilnis et al., 2018; Vilnis & McCallum, 2015; Gu et al., 2019). The most prominent of these embedding methods include hyperbolic embeddings (Nickel & Kiela, 2017; Ganea et al., 2018a;b) and box embedding (Vilnis et al.,

2018; Li et al., 2019; Dasgupta et al., 2020a). The use of representations other than Euclidean vectors for improving the coherence of multi-label classification has been limited to specific domains like text (Chatterjee et al., 2021) or specific tasks like entity typing (Onoe et al., 2021). Moreover, while both hyperbolic and box embedding can model hierarchical relationships, it has been shown that the box embedding can also model more general graphs like DAGs much more efficiently than hyperbolic embeddings (Patel et al., 2020). Hence, we propose a model that uses box embedding to capture general label-label relationships without the explicit use of label taxonomy to improve the coherence of model predictions.

3 Overview of Box Embeddings

Notations: In the problem of multi-label classification, we are given a set of labels $\mathcal{L}$ where $L = |\mathcal{L}|$, and an instance can be labeled with an element $s \in \{0,1\}^L$, where projection to the ith coordinate $\pi_i(s) = 1$ means that the ith label is true. We call the set of all such labelings S, and the associated probability space $(S, \mathcal{P}(S), P_S)$. We use $\mathbb{I}$ to denote the set of all finite closed intervals $[\mu^-, \mu^+]$ in $\Omega \subset \mathbb{R}$ plus the empty set, i.e. $\mathbb{I} := \{[\mu^-, \mu^+] \subset \Omega \,|\, \mu^+ \geq \mu^-\} \cup \emptyset$. We denote the smallest σ-algebra containing $\mathbb{I}$ as $\sigma(\mathbb{I})$ and, given a valid finite measure ν, we consider the measure space $(\Omega, \sigma(\mathbb{I}), \nu)$. As a high dimensional generalization, $\mathbb{I}^d$ will denote a d-dimensional Cartesian product of $\mathbb{I}$.

Definition 1 (Box Embedding (Vilnis et al., 2018))**.** Let $B : \mathbb{I}^d \to S$ be a measurable function such that $B^{-1} \circ \pi_i^{-1}(1) = \prod_i^d [\mu_i^-, \mu_i^+] \in \mathbb{I}^d$. A *box embedding* is defined as the function $\mathrm{Box} : \mathcal{L} \to \mathbb{I}^d$ which maps a label $\ell \in \mathcal{L}$ to $B^{-1} \circ \pi_\ell^{-1}(\{1\}) \in \mathbb{I}^d$.

The definition of *box embeddings* induces a push-forward measure Q on S such that for any $R \subseteq S$, $Q(S) = \nu \circ B^{-1}(R)$. The complete joint probability distribution over the labels can be modeled using Q as defined above; however, computing $B^{-1}(R)$ requires the use of inclusion-exclusion principle and hence is intractable for a general R.

In order to avoid local identifiability issues in training, Dasgupta et al. (2020b) interpret μ_i^- (resp. μ_i^+) as the location parameters of random variables M_i^- (resp. M_i^+) that are distributed according to GumbelMax (resp. GumbelMin) distributions, leading to a meta-probabilistic generalization of box embedding which they call *Gumbel Box Process*. Since GumbelMax (resp. GumbelMin) is a max (resp. min) stable distribution, it enables the computation of the location parameters of the intersection box as given in the following definition.

Definition 2 (Intersection Box (Dasgupta et al., 2020b))**.** Let $A = \prod_{i=1}^d [a_i^-, a_i^+]$ and $B = \prod_{i=1}^d [b_i^-, b_i^+]$ be two gumbel boxes expressed using their location parameters, then the location parameters of the intersection of these two *gumbel boxes* are given as

$$A \tilde{\cap} B = \prod_{i=1}^{d} \left[\beta \operatorname{lse}\left(\frac{a_i^-}{\beta}, \frac{b_i^-}{\beta}\right), \quad -\beta \operatorname{lse}\left(-\frac{a_i^+}{\beta}, -\frac{b_i^+}{\beta}\right) \right], \tag{1}$$

where $\operatorname{lse}(x, y) = \log(\exp(x) + \exp(y))$.

The expected volume of Gumbel boxes involves the Bessel Function of the Second Kind, however, as shown in Dasgupta et al. (2020b), this integral can be reasonably approximated using softplus function leading to the following definition for *approximate bessel volume*.

Definition 3 (Approximate Bessel Volume (Dasgupta et al., 2020b))**.** For a gumbel box $B = \prod_{i=1}^d [b_i^-, b_i^+]$ we define the approximate Bessel volume $\lambda : \mathbb{I}^d \to \mathbb{R}_+$ as

$$\lambda(B) := \prod_{i=1}^{d} \log\left(1 + \exp\left(\frac{b_i^+ - b_i^-}{\beta} - 2\gamma\right)\right).$$

In the next section, we formally demonstrate the suitability of box embeddings for capturing taxonomic label relationships, and for that we first prove a couple of useful facts regarding the Gumbel intersection and Bessel approximate volume.

Proposition 1. Approximate bessel volume *is monotonic with respect to set containment. That is for two Gumbel boxes* A, B*,*

$$a_i^- \geq b_i^- \quad \textit{and} \quad a_i^+ \leq b_i^+, \quad \forall i \in \{1, \ldots, d\} \iff \lambda(A) \leq \lambda(B). \tag{2}$$

Proof. Follows from the monotonicity of $\log(1 + \exp(.))$. □

Proposition 2. *For any two Gumbel boxes* A, B*,* $\lambda(A \tilde{\cap} B) \leq \lambda(B)$*.*

Proof. The fact that $\max(x, y) \leq \text{lse}(x, y)$, and the statement of proposition 1 together imply the desired result. □

Since λ is neither normalized nor additive, it cannot be used as a probability measure on $(\Omega, \sigma(\mathbb{I}^d))$. However, we can use proposition 1 and 2 to define a conditional probability model as follows.

Corollary 1. *For two gumbel boxes* A, B*, let* $P_{\text{Box}}(A \mid B) = \frac{\lambda(A \tilde{\cap} B)}{\lambda(B)}$*, then*

(i) For any two gumbel boxes A, B*, we have* $0 \leq P_{\text{Box}}(A \mid B) \leq 1$*.*

(ii) $P_{\text{Box}}(A \mid C) \leq P_{\text{Box}}(B \mid C)$ *for any three gumbel boxes* A, B, C*, with* $a_i^- \geq b_i^-, a_i^+ \leq b_i^+$*.*

4 Multi-label Box Model

In order to perform the task of multi-label classification we need to model the conditional probabilities $P(Y|X)$ where $Y \in S$ and X is the input. Using definition 1, we define label *box embeddings* $\text{Box}_\psi : \mathcal{L} \to \mathbb{I}^d$ as

$$\text{Box}_\psi(\ell_i) := \prod_{j=1}^{d} [\psi_{i,j}^-, \ \psi_{i,j}^- + \log(1 + \exp \psi_{i,j}^+)],$$

where $\psi^-, \psi^+ \in \mathbb{R}^{L \times d}$ are trainable parameters. The input instance X is encoded as a fixed-width element of $\mathbb{I}^d$ using a parametric *instance box embedding* $\text{Box}_\theta = I^d \circ \mathcal{F}_\theta : \mathcal{X} \to \mathbb{I}^d$, where $\mathcal{F}_\theta : \mathcal{X} \to \mathbb{R}^d$ is a neural network with parameters θ and $I^d : \mathbb{R}^d \to \mathbb{I}^d$ defined as

$$I^d(x) := \prod_{i=1}^{d} [x_i - \delta, x_i + \delta],$$

where $\delta = 10^{-5}$. The conditional probability for $Y \in S$ given input X is computed using conditional probability under the Gumbel box model as

$$P_{\text{MBM}}(Y|X; \psi, \theta) = \prod_{i=1}^{L} P_{\text{MBM}}(Y_i|X, \psi, \theta) := \prod_{i=1}^{L} P_{\text{Box}}(B^{-1} \circ \pi_i^{-1}(\{Y\}) \mid \text{Box}_\theta(X))$$

Using the definition of P_{Box} as stated through corollary 1, we get the following expression for the conditional probability of Y_i under the model, where the intersection $\tilde{\cap}$ is the *Gumbel Intersection* and measure λ is *Approximate Bessel Volume*.

$$P_{\text{MBM}}(Y_i = 1|X; \psi, \theta) = \frac{\lambda\left(\text{Box}_\psi(\ell_i) \tilde{\cap} \text{Box}_\theta(X)\right)}{\lambda(\text{Box}_\theta(X))}$$

4.1 Modeling label-space interactions

In Section 1, we alluded to the fact that the inductive bias of the MBM allows it to efficiently model partially specified first-order label interactions. Now we make this remark more concrete. If the partial specification of label interaction is defined using a taxonomy that can be represented as a directed acyclic graph (DAG), the following proposition shows that MBM has a strong inductive bias towards maintaining coherence in its scores.

Proposition 3. *Let* $G = (\mathcal{L}, \mathcal{T})$ *denote a DAG defined over the labels where* $\mathcal{L}$ *is the set of all labels and* $\mathcal{T} = \{(\ell_i, \ell_j) \mid \ell_i, \ell_j \in \mathcal{L}, \ P_D(y_i = 1 \mid y_j = 1) = 1\}$ *is the set of edges. Then there exists some* ψ *such that* $P_{(\psi,\theta)}(y_i = 1 \mid x) \geq P_{(\psi,\theta)}(y_j = 1 \mid x)$*, for all* x, θ*.*

Proof. For all $(\ell_i, \ell_j) \in \mathcal{T}$, let ψ be such that $\text{Box}_\psi(\ell_j) \subseteq \text{Box}_\psi(\ell_i)$. Note that such ψ exists since for each $i \in \{1, \ldots, L\}$, $\text{Box}_\psi(\ell_i)$ is defined using only ψ_i. It follows from corollary 1 that $P_{\text{Box}}(\text{Box}_\psi(\ell_i) \mid \text{Box}_\theta(X)) \geq P_{\text{Box}}(\text{Box}_\psi(\ell_j) \mid \text{Box}_\theta(X))$ for any X, θ. □

4.2 LEARNING

The entire MBM is specified using parameters (ψ, θ) where $\psi \in \mathbb{R}^{2d \times L}$ are the label embedding parameters and θ are the parameters of the instance encoder neural network $\mathcal{F}_\theta$. Given data $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$, the model parameters are learnt by minimizing negative log-likelihood loss

$$L_{\text{nll}}(\psi, \theta; D) = -\sum_{i=1}^{D} \sum_{j=1}^{L} \log P(y_j^{(i)} \,|\, x^{(i)}; \psi, \theta), \tag{3}$$

using the ADAM optimizer (Kingma & Ba, 2017). In order to empirically verify the intuition behind proposition 3, we also propose the use of label interaction loss

$$L_{\text{G}}(\psi) = -\sum_{(\ell_i, \ell_j) \in \mathcal{T}} s(\ell_i, \ell_j) + \sum_{(\ell_i, \ell_j) \notin \mathcal{T}} s(\ell_i, \ell_j) \tag{4}$$

that utilizes the geometry of box embeddings to inject partial information about label interactions specified using a label taxonomy $G = (\mathcal{L}, \mathcal{T})$. When label interaction loss is applied (MBM-T), the total loss is $L_T = L_{\text{nll}} + \nu L_G$, where ν is a hyperparameter. For the Box model, label interaction score for a pair of labels is defined as

$$s_{\text{MBM}}(\ell_i, \ell_j) := \log P_{\text{Box}}(\text{Box}_\psi(\ell_i) \,|\, \text{Box}_\psi(\ell_j)). \tag{5}$$

5 BASELINES

Our choice of baselines reflects the focus of this work, i.e., introducing coherence in prediction using suitable representation spaces. To this end, our baselines consist of a high-performing neural network that only uses Euclidean vector representations (MVM), and another that uses hyperbolic representations (MHM). In order to test the importance of the probabilistic semantics used to formulate MBM, we also include as baseline, a non-probabilistic box model as defined in Abboud et al. (2020). The base input encoder architecture $\mathcal{F}_\theta$ in all these baselines is same as the one used in MBM.

Multi-label Vector Model (MVM) An input encoder neural network $\mathcal{F}_\theta : \mathcal{X} \to \mathbb{R}^d$ is used to encode the inputs and a label embedding matrix ψ is used to represent the labels. The conditional probability of labels given the input and label interaction scores are given as

$$P_{\text{MVM}}(y_l = 1 \,|\, x; \psi, \theta) := \sigma(\mathcal{F}_\theta(X)^T M_\theta \, \psi_l), \quad \text{and} \quad s_{\text{MVM}}(l_i, l_j) := \sigma(\psi_{l_i}^T M_\theta \, \psi_{l_j}),$$

respectively, where σ is the logistic sigmoid function, and $G_\theta \in \mathbb{R}^{d \times d}$ is a matrix of trainable parameters. The parameters (θ, ψ) are learnt through L_T (Eq. 3,4). Note that, when label interaction is not being explicitly modeled i.e., $\nu = 0$, M is fixed as the identity matrix and MVM reduces to a special case of multi-layer perceptron.

Multi-label Hyperbolic Model (MHM) As discussed in Chatterjee et al. (2021), the isometry between the Lorentz model and Poincare disk model for hyperbolic geometry can be used produce the retraction formula used to project d-dimensional euclidean vectors to d-dimensional Poincaré ball $\mathcal{B}^d = \left\{x \in \mathbb{R}^d \mid \|x\| < 1\right\}$. Here the projection operator $\Pi : \mathbb{R}^d \to \mathcal{B}^d$ and the distance $d : \mathcal{B}^d \times \mathcal{B}^d \to \mathbb{R}_+$ in the hyperbolic space are given as:

$$\Pi(x) := \frac{x}{1 + \sqrt{1 + \|x\|_2^2}} \quad , \qquad d(u, v) := \operatorname{arcosh}\left(1 + 2\frac{\|u - v\|}{(1 - \|u\|_2^2)(1 - \|v\|_2^2)}\right).$$

Specifically, the input is first encoded using $\mathcal{F}_\theta$ and then projected into $\mathcal{B}^d$. The unnormalized score for each label is computed as the negative of the distance between the hyperbolic projections of encoded input and label representation. Since the hyperbolic distance function consists of arcosh, the negative distance is interpreted as log-probability score. Hence, the conditional log-probability of the labels and the label interaction score is given, respectively, by the following expressions, which are then used to learn the parameters (ψ, θ) in the same way as MVM and MBM.

$$\log P_{\text{MHM}}(y_l = 1 \,|\, x) = -d(\Pi(\mathcal{F}_\theta(x)), \Pi(\psi_l)) \quad \text{and} \quad s_{\text{MHM}}(l_i, l_j) := -d(\Pi(\psi_{l_i}), \Pi(\psi_{l_j})),$$

Non-probabilistic box model (BoxE): In order to test the importance of the probabilistic semantics used to formulate the MBM, we also include as baseline, a box model that does not use probabilistic semantics. For this, we replace the P_{MBM} with the non-probabilistic score defined in Abboud et al. (2020) and replace the log-likelihood loss (L_{nll}) with a margin based loss L_{margin}. [1]

6 Evaluation and Results

We evaluate the performance of MBM using 12 real-world multi-label classification datasets (Dimitrovski et al., 2011; Clare, 2003) spanning across three domains: text (Enron), images (Imclef07a, Imclef07d, Diatoms), and functional genomics (Expr, Cellcycle, Derisi, Spo) . These datasets are the ideal test bed as they provide explicit label taxonomies with different connectivity: trees, forests, and more general DAGs. Moreover, all the labels of all training and test instances respect the label taxonomy. THe datasets include both categorical and continuous input features. We convert the categorical features into one-hot feature vectors and standardize all continuous features. The input encoder $\mathcal{F}_\theta$ uses a common architecture for all models consisting of an MLP with a maximum of 3 layers. We perform a grid search over activations, hidden dimensions, dropout, learning rate and use the best parameters for each model.[1] The Mean Average Precision (MAP), that is the mean over the average precision values across instances in the test set, is used to evaluate the predictive performance of the models. Table 1 reports average metric values across 5 runs with different random seeds (the standard error intervals are small and are omitted for the sake of readability).

As seen in Table 1, the predictive performance (MAP) of MBM is better than all other methods on at least 9 out of 12 datasets. To check the statistical significance of our results, following Demšar (2006), we first perform the Friedman test and obtain a p-value of 2×10^{-9}. The critical diagram of the post-hoc Nemenyi test (Figure 2a) performed after the Friedman test allows us to conclude that difference in predictive performance between MBM and all other models except MVM is statistically significant. Through the more powerful pairwise Wilcoxon test, we find that the performance of MBM and MVM is comparable (p-value of 0.1). Furthermore, from these results, one can conclude with confidence that MBM with its probabilistic formulation preforms significantly better than BoxE, which is the non-probabilistic variant of box embeddings. In the following section, we discuss another important aspect of performance, that is, the coherence of predictions.

6.1 Coherence analysis

MAP is a good metric for predictive performance, however it does not take into account the inconsistencies in the predicted scores w.r.t. the label taxonomy. For instance, recalling the earlier example in Figure 1, a consistent model would always assign higher score to *fiction* when compared to *drama*, since a book classified as *drama* should also be classified as *fiction*. Since MAP is incapable of capturing such coherence conditions, we perform further analysis to check the coherence of the predicted scores. This is done by considering two opposite perspectives towards coherence evaluation. The first one is a punitive approach, where we count the number of instances that are given inconsistent scores w.r.t the taxonomy. In the second one, a more constructive approach, we try to check the amenability of the model, by applying post-hoc corrections to the predicted scores to enforce coherence. The results of these two approaches are presented by using Constraint Violation (CV) and Mean Average Precision post Coherence correction (CMAP), which are described below. Note that we do not recommend the use of either CV or CMAP as the sole metrics to evaluate the model, we present these as measures to analyse the coherence of the model predictions.

Constraint violation is a punitive measure that quantifies the extent to which the label scores generated by the model violate the partial ordering of the latent label taxonomy regardless of true labels for the instances. Hence, lower value of CV implies higher taxonomic coherence in the predictions.

$$\text{CV}(s) = \frac{1}{|D||\mathcal{T}|} \sum_{k=1}^{|D|} \sum_{(l_i, l_j) \in \mathcal{T}} \mathbb{1}\left(s_i^{(k)} - s_j^{(k)} < 0\right). \tag{6}$$

Mean Average Precision post Coherence correction (CMAP): Given a complete or partial taxonomy $G = (\mathcal{L}, \mathcal{T})$ for the labels, coherence can be imposed post-hoc by applying a modification

[1]Please see the *Reproducibility Statement* after Section 8, for further details on the implementation of the baselines and the experiments.

Table 1: Performance comparison of MBM models with the baselines for the 12 feature based multi-label classification datasets. The left section compares the models that do not require explicit taxonomy, i.e., BoxE, MVM, MHM and MBM. In the left section, the models with the best performance w.r.t. MAP are highlighted. The right section shows the performance when we include taxonomy information in training through L_G(MVM-T, MHM-T and MBM-T), where the highlighted cells indicate an improvement in performance (MAP and CV) w.r.t. the respective non-T model. All metrics reported are averaged across five runs with different seeds.

Dataset	Metric	BoxE	MVM	MHM	MBM	MVM-T	MHM-T	MBM-T	C-HMCNN
ExprFUN	MAP↑	48.41	49.11	39.44	48.35	41.70	38.23	47.88	46.29
	CMAP↑	48.96	46.47	37.54	48.60	38.16	38.47	48.13	46.29
	CV↓	5.07	2.75	0.72	2.21	4.12.	1.61	1.52	0
CellcycleFUN	MAP↑	43.41	42.80	38.68	44.77	38.43	38.54	44.91	42.8
	CMAP↑	43.65	42.55	36.94	44.94	37.15	36.78	45.07	42.8
	CV↓	2.73	1.73	0.79	1.57	2.68	0.96	0.4	0
DerisiFUN	MAP↑	40.23	40.09	37.96	40.81	36.50	37.44	40.67	40.02
	CMAP↑	40.52	39.50	37.25	40.93	32.16	37.43	40.72	40.02
	CV↓	2.54	2.01	0.77	1.34	3.77	1.23	0.05	0
SpoFUN	MAP↑	40.98	40.12	38.21	41.35	35.36	37.24	41.37	33.60
	CMAP↑	41.17	40.07	37.23	41.51	32.59	37.31	41.53	33.60
	CV↓	2.21	1.42	0.86	1.53	4.74	2.68	1.21	0
ExprGO	MAP↑	47.57	58.31	57.05	57.79	52.41	52.08	57.13	56.44
	CMAP↑	48.33	51.64	51.67	57.94	47.09	52.23	57.29	56.44
	CV↓	4.63	4.46	3.94	3.02	4.51	2.06	3.13	0
CellcycleGO	MAP↑	50.48	55.14	52.06	55.93	52.33	52.07	55.14	53.90
	CMAP↑	50.92	49.34	52.24	56.08	47.28	52.26	55.22	53.90
	CV↓	3.57	3.10	1.92	2.93	4.14	2.03	0.62	0
DerisiGO	MAP↑	46.41	53.23	52.23	53.92	51.87	52.22	53.79	52.98
	CMAP↑	47.21	46.77	52.25	54.02	46.79	52.26	53.87	52.98
	CV↓	5.26	2.82	1.93	2.66	7.71	2.06	2.55	0
SpoGO	MAP↑	46.03	53.34	52.18	53.95	51.48	52.25	53.23	52.32
	CMAP↑	45.65	46.82	52.22	54.04	45.71	52.29	53.29	52.32
	CV↓	5.31	3.28	1.94	2.40	8.56	1.99	0.71	0
Enron	MAP↑	83.43	83.76	79.61	82.08	76.73	79.60	81.32	82.29
	CMAP↑	83.43	75.65	79.63	82.12	59.93	79.63	81.34	82.29
	CV↓	0.46	0.79	0.35	0.12	3.02	0.48	0.04	0
Diatoms	MAP↑	79.38	78.40	47.63	83.47	52.26	47.71	82.75	62.1
	CMAP↑	79.72	71.21	47.65	83.55	45.63	47.73	82.66	62.1
	CV↓	6.40	7.23	2.03	3.48	4.59	1.93	0.19	0
Imclef07a	MAP↑	72.48	73.97	65.16	74.3	67.51	65.65	74.94	74.64
	CMAP↑	72.44	73.87	64.78	74.52	63.46	64.56	75.14	74.64
	CV↓	3.17	2.98	2.27	2.92	12.62	2.49	2.83	0
Imclef07d	MAP↑	83.70	84.04	71.73	85.33	70.16	71.71	84.89	84.69
	CMAP↑	84.15	77.56	71.88	85.58	57.39	71.86	85.25	84.69
	CV↓	8.24	9.39	4.47	5.48	8.95	5.44	5.24	0
Avg. Rank ↓	MAP	4.8	2.9	6.2	**1.8**	6.8	6.7	2.4	4.4
	CMAP	4.2	5.4	6.0	**1.6**	7.6	5.5	2.0	3.7
	CV	6.9	6.2	2.9	4.6	7.5	4.0	2.7	**1.0**

function $\delta : \mathbb{R}^L \to \mathbb{R}^L$ to the label-scores produced by the model such that $\delta(s_i) - \delta(s_j) < 0$ for all $(l_i, l_j) \in \mathcal{T}$. This can be achieved by modifying the score s_i for each label l_i to be either the maximum (δ_G^M) of the scores of any of its descendants or the minimum of the scores of its ancestors (δ_G^m) in the taxonomy G. Concretely, given the scores $s \in \mathbb{R}^L$ produced by a model, for an input, the two modification functions are given as

$$\delta_G^m(s)_i = \min_{l_j \in \text{Anc}_G(l_i) \cup \{l_i\}} s_j, \qquad \delta_G^M(s)_i = \max_{l_j \in \text{Des}_G(l_i) \cup \{l_i\}} s_j, \tag{7}$$

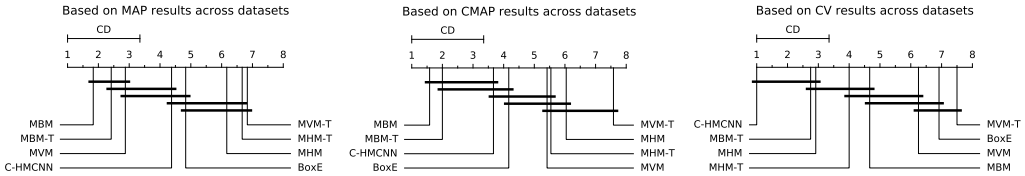


Figure 2: Critical diagrams of the post-hoc Nemenyi test across all 12 datasets.

where $\text{Anc}_G(l)$ and $\text{Des}_G(l)$ is the set of ancestors and descendants, respectively, of l in the graph G. In practice, one would select whichever post-hoc modification performs better, and we refer to the maximum MAP obtained after applying one of the modification functions δ_G^m, δ_G^M to the scores as CMAP.[2] Intuitively, a CMAP value close to MAP value implies that the model is more perceptive of the latent hierarchy in the label space. As seen from the critical diagram of the Nemenyi test (Figure 2b), when coherence is considered along with the predictive performance (CMAP), MBM outperforms MVM and MHM (with statistical significance) indicating that MBM is much more perceptive of the latent taxonomy compared to MVM. In conclusion, while on one extreme there is MVM, which exhibits good predictive performance but fails to maintain coherence w.r.t the taxonomy, on the other extreme we have MHM that exhibits lowest constraint violations but gives inadequate predictive performance. MBM, however, demonstrates good characteristics on both fronts–predictive performance as well as coherence.

Comparison with the state-of-the-art: C-HMCNN (Giunchiglia & Lukasiewicz, 2020), is the current state-of-the-art method for hierarchical MLC, and includes the modification function δ_G^M (called MCM in their work) in their model directly, applying it to scores produced by a multi-layer perceptron at training as well as inference time.[3] This is done by cleverly modifying BCE loss to MCLoss which works well with the modification function during training. There are two key differences between C-HMCNN and MBM that one needs to keep in mind when comparing performance. First, C-HMCNN does not attempt to use the label representations themselves to increase coherence. Second, unlike MBM, C-HMCNN requires the complete label taxonomy at training and inference time. Due to the use of label taxonomy at inference time, C-HMCNN has 0 constraint violations. However, as seen from the Table 1 and the Figure 2, MBM performs significantly better than C-HMCNN w.r.t MAP as well as CMAP (with p-value of 0.002 in Wilcoxon test).

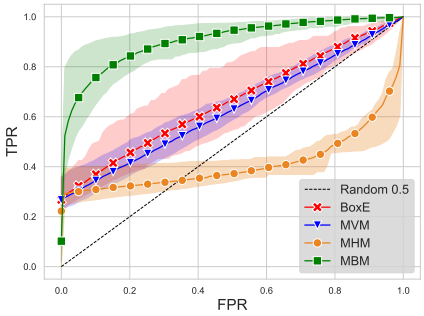
7 Analysis of Learned Label Embeddings

In this section, we analyze the geometry of the learned label embeddings, finding that the simple geometry of box embeddings endows the MBM model with high degree of interpretability. In order to verify that label box embeddings are producing consistent scores by using inclusion in the box space, we inject into the model the taxonomy information through the additional loss term (Eq. 4). As seen from the two right most columns in Table 1, injecting explicit taxonomic information into the label embeddings further reduces the extent of constrain violation for MBM and MHM models. However, it does not aid the MVM model and even drops its predictive performance significantly. This underscores the importance of embedding geometry for inducing taxonomic coherence, and validates our intuition about the arrangement of label embeddings in MBM.

To determine the extent to which the label embeddings capture the latent label taxonomy without it being explicitly provided, we perform ancestor-descendant classification solely using the learned label embeddings. Each pair of labels (l_i, l_j) get a score $\beta(i, j)$ that is determined using their corresponding label embeddings ψ_i, ψ_j. Since MVM and MBM have different geometrical interpretations, we use different scoring for each. Specifically, $\beta_{\text{MVM}}(i, j) = \frac{\psi_i \cdot \psi_j}{\|\psi_j\|}$, $\beta_{\text{MBM}}(i, j) = s_{\text{MBM}}(l_i, l_j)$. These scores are then compared to true ancestor-descendant relations

[2]We find that δ_G^m produces lesser change in the scores predicted by all the models (MBM, MVM and MHM), and hence is used as the modification function for this analysis (see Appendix E for further discussion on this)

[3]Comparision with another competitive baseline HMCN is provided in Appendix D.



Dataset	BoxE	MVM	MHM	MBM
ExprFUN	0.59	0.61	0.40	**0.89**
CellcycleFUN	0.71	0.59	0.40	**0.93**
DerisiFUN	0.64	0.61	0.41	**0.87**
SpoFUN	0.65	0.61	0.39	**0.87**
Enron	0.77	0.66	0.49	**0.92**
Diatoms	0.67	0.68	0.44	**0.96**
Imclef07a	0.58	0.63	0.45	**0.87**

Figure 3: The figure on the left shows the envelope of the ROC curves for the ancestor-descendant relationship classification in the label space for different embedding based models across the datasets and the table on the right is the area under the ROC curves. As seen, MBM implicitly captures the label taxonomy well compared to all other models.

Table 2: Spearman rank correlation between the number of descendants in the true label taxonomy compared with each of the following: embedding magnitude for MVM, negative embedding magnitude for MHM and box embedding volume for MBM.

Model	Expr FUN	Cellcycle FUN	Derisi FUN	Spo FUN	Enron	Diatoms	Imclef07a
MHM	-0.37	0.38	0.40	0.38	0.19	0.31	0.32
MVM	-0.06	-0.11	-0.01	0.06	-0.11	0.04	-0.02
MBM	0.47	0.49	0.50	0.48	0.47	0.23	0.43

in the taxonomy to obtain respective ROC curves as shown in Figure 3. As seen in Table 3, MBM captures the true label taxonomy the best ($AUC \geq 0.87$) for all datasets.

Furthermore, for hyperberbolic space, it is suggested that the magnitude of embeddings relate to the level of generality in taxonomy (Nickel & Kiela, 2017). We show that the same observation holds for box embeddings, with the vector embedding magnitude replaced by box embedding volume. To see this, we compute the Spearman rank correlation between the number of descendants of a node in the true taxonomy and the embedding magnitude, negative embedding magnitude, and embedding volume for MVM, MHM, and MBM, respectively. The correlation values reported in Table 2 confirm our intuition regarding box embeddings stated above.[4] It is interesting to note that as shown by good correlation between the vector magnitude and level of the label node as well as low constraint violations, the MHM model does capture the depth of the label nodes in the taxonomy. However, as shown by the ROC curves in Figure 3, unlike MBM, it fails to capture the exact connectivity of the nodes at different depths in the taxonomy.

8 Conclusion

In this work, we demonstrate that box embeddings with its probabilistic formulation can effectively capture taxonomic relations present between labels in multi-label classification without requiring explicit access to the taxonomy. The multi-label box model achieves a fine balance between predictive performance and coherence. Furthermore, we find that the graph relationships between labels can be easily injected via extra supervision during training, increasing the coherence of the model predictions. Since the model has the same computational complexity as a simple vector neural network model (MVM), and unlike C-HMCNN it does not require the label taxonomy, we believe that the proposed model can be scaled to work with extremely large label spaces. Moreover, due to the flexibility of using any neural network encoder this model can be easily extended to different input modalities, such as raw text, images, etc. We wish to pursue these two directions in our future work.

[4]Refer to Appendix H for scatter plots of box embedding volume vs. the number of children in the label taxonomy.

REPRODUCIBILITY STATEMENT

We have strove make the results reported in this work easily reproducible. In this section, we provide pointers to information necessary to reproduce the results mentioned in this paper. The description of the datasets with various statistics, links to download them, and instructions to pre-process them are provided in Appendix B. The BoxE baseline is further described with expressions for distance and the margin based loss function in Appendix A.1. The hyper-parameter settings for each model-dataset combination (all 96 settings) used to produce the reported results are provided in 3. We also include a discussion on computational complexity of the models in Appendix A.2. Finally, an anonymized version of the code used in this paper is publicly available and can be obtained by following the instructions in A.3.

REFERENCES

Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. BoxE: A box embedding model for knowledge base completion. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in neural information processing systems*, volume 33, pp. 9649–9661. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/6dbbe6abe5f14af882ff977fc3f35501-Paper.pdf.

Zafer Barutcuoglu, Robert E. Schapire, and Olga G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 01 2006. ISSN 1367-4803. doi: 10.1093/bioinformatics/btk048. URL https://doi.org/10.1093/bioinformatics/btk048.

David Belanger and Andrew McCallum. Structured prediction energy networks. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 983–992, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/belanger16.html.

Ricardo Cerri, Rodrigo C. Barros, and AndrÃ© C.P.L.F. de Carvalho. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, 80(1):39 – 56, 2014. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2013.03.007. URL http://www.sciencedirect.com/science/article/pii/S0022000013000718.

Soumya Chatterjee, Ayush Maheshwari, Ganesh Ramakrishnan, and Saketha Nath Jagaralpudi. Joint learning of hyperbolic label embeddings for hierarchical multi-label classification. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2829–2841, Online, April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.eacl-main.247.

Amanda Clare. *Machine learning and data mining for yeast functional genomics*. PhD thesis, Citeseer, 2003.

Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Lorraine Li, and Andrew McCallum. Improving Local Identifiability in Probabilistic Box Embeddings. 2020a. ISSN 23318422. URL https://www.iesl.cs.umass.edu/http://arxiv.org/abs/2010.04831.

Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Lorraine Li, and Andrew McCallum. Improving local identifiability in probabilistic box embeddings. In *Advances in Neural Information Processing Systems*, 2020b. URL https://proceedings.neurips.cc/paper/2020/file/01c9d2c5b3ff5cbba349ec39a570b5e3-Paper.pdf.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Hierarchical annotation of medical images. *Pattern Recognition*, 44(10-11):2436–2449, 2011.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. Technical report, 2018a. URL https://en.wikipedia.org/wiki/.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic Neural Networks. 2018b. doi: arXiv:1805.09112v2. URL http://arxiv.org/abs/1805.09112.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. 2017.

Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent Hierarchical Multi-Label Classification Networks. 2020. URL http://arxiv.org/abs/2010.10151.

Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJxeWnCcF7.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. Hierarchical multi-label text classification: An attention-based recurrent network approach. In *International Conference on Information and Knowledge Management, Proceedings*, pp. 1051–1060. ACM, 2019. ISBN 9781450369763. doi: 10.1145/3357384.3357885. URL https://doi.org/10.1145/3357384.3357885.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. Smoothing the geometry of probabilistic box embeddings. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1xSNiRcF7.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. Hierarchical losses and new resources for fine-grained entity typing and linking. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pp. 97–109, 2018. ISBN 9781948087322. doi: 10.18653/v1/p18-1010. URL https://www.aclweb.org/anthology/P18-1010.

Maximilian Nickel and Douwe Kiela. Poincar\'e Embeddings for Learning Hierarchical Representations. 2017. ISSN 19457871. doi: 10.1109/ICME.2013.6607554. URL http://arxiv.org/abs/1705.08039.

Yasumasa Onoe, Michael Boratko, and Greg Durrett. Modeling fine-grained entity types with box embeddings, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Dhruvesh Patel, Shib Sankar Dasgupta, Michael Boratko, Xiang Li, Luke Vilnis, and Andrew McCallum. Representing joint hierarchies with box embeddings. In *Automated Knowledge Base Construction*, 2020. URL https://openreview.net/forum?id=J246NSqR_l.

C. N. Silla and A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22:31–72, 2010.

Luke Vilnis and Andrew McCallum. Word representations via Gaussian embedding. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. 2018. URL https://people.cs.umass.edu/~luke/box-lattices.pdfhttp://arxiv.org/abs/1805.06627.

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, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018a. PMLR. URL http://proceedings.mlr.press/v80/wehrmann18a.html.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. Hierarchical multi-label classification networks. In *International Conference on Machine Learning*, pp. 5075–5084. PMLR, 2018b.

Jônatas Wehrmann, Ricardo Cerri, and Rodrigo C Barros. Hierarchical Multi-Label Classification Networks. Technical report, 2018c.

A IMPLEMENTATION DETAILS

In this section, we describe the implementation details: training procedure and hyper-parameter search.

Frameworks used: We implement all the models described in this work using PyTorch (Paszke et al., 2019). We also make use of NLP specific abstractions over PyTorch provided by AllenNLP (Gardner et al., 2017).

Data pre-processing: The datasets were pre-processed to remove noisy characters, fix encoding issues, standardize the continuous variables and map the labels to binary one hot encoding label vector.

Training: The learning algorithm used for training is minibatch gradient descent with a fixed batch size of 64. The ADAM (Kingma & Ba, 2017) optimizer was used during training, along with ReduceLROnPlateau learning rate scheduler with early stopping. Since the naive implementation of the label interaction loss described in Eq. 4 is too expensive to compute at each mini-batch step, we approximate it by randomly sampling (without replacement), at each mini-batch step, a subset of edges $\tilde{\mathcal{T}} \sim \mathcal{T}$. The size of the sampled set is a hyper-parameter (last column in Table 3).

Hyper-parameter search: The following hyper-parameters were searched using the performance on the validation set: optimizer learning rate, hidden dimensions of the MLP and activation functions, dropout probabilities, label space dimensions/box space dimensions(half of hidden dimensions), weight and sampling percentage for the labels for computing the label interaction loss(L_G) for the MHM-T, MBM-T and MVM-T models. All the results are reported using the best hyper-parameters found using grid search for each model-dataset combination. Table 3 summarizes the search ranges used and Table 4 presents the final hyper-parameters obtained.

Table 3: Summary of the hyper-parameter search ranges for each dataset and model. The best hyper-parameters for each model and dataset combination were picked using grid search using MAP on the validation set. Note that ν and *label sample percent* are only applicable to the MBM-T, MVM-T and MHM-T models.

Datasets	batch size	lr	hidden dim	layers	activation	dropout	L_G weight (ν)	label sample percent
all	64	1e-5, 1e-4, 1e-2	250, 500, 1000, 1750	1,2,3	sigmoid, relu, tanh	0.0, 0.3, 0.5, 0.7	1e-3, 1e-5, 1e-6, 1e-7, 1e-9	10,15,20,30

Table 4: Best hyper-parameters obtained through grid search for each dataset-model combination.

Dataset	Model	learning rate	hidden dim	layers	dropout	% label sample	ν
CellcycleFUN	BoxE	0.001	1000	3	0.5	-	-
CellcycleFUN	C-HMCNN	0.001	1000	3	0.5	100	-
CellcycleFUN	MBM	0.001	1750	3	0	-	-
CellcycleFUN	MBM-T	0.001	1750	3	0	100	1e-6
CellcycleFUN	MHM	0.001	1750	3	0	-	-
CellcycleFUN	MHM-T	0.001	1750	3	0	100	1e-7
CellcycleFUN	MVM	0.001	1000	3	0.5	-	-
CellcycleFUN	MVM-T	0.001	1000	3	0.3	100	1e-3
CellcycleGO	BoxE	0.001	1750	3	0.3	-	-
CellcycleGO	C-HMCNN	0.001	500	2	0	100	-
CellcycleGO	MBM	0.001	1000	3	0	-	-
CellcycleGO	MBM-T	0.001	500	3	0.3	20	1e-6
CellcycleGO	MHM	0.001	500	3	0.3	-	-
CellcycleGO	MHM-T	0.001	500	3	0.3	15	1e-7
CellcycleGO	MVM	0.001	1750	3	0.3	-	-
CellcycleGO	MVM-T	0.001	1750	3	0.3	30	1e-7
DerisiFUN	BoxE	0.001	1000	3	0	-	-
DerisiFUN	C-HMCNN	0.001	1000	3	0	100	-
DerisiFUN	MBM	0.001	500	3	0	-	-
DerisiFUN	MBM-T	0.001	1750	3	0	100	1e-5
DerisiFUN	MHM	0.001	1750	3	0	-	-
DerisiFUN	MHM-T	0.001	1750	3	0	100	1e-7

Table 4 continued from previous page

Dataset	Model	learning rate	hidden dim	layers	dropout	% label sample	
DerisiFUN	MVM	0.001	1000	3	0	-	-
DerisiFUN	MVM-T	0.001	1000	3	0.3	100	1e-3
DerisiGO	BoxE	0.001	1750	2	0.5	-	-
DerisiGO	C-HMCNN	0.001	500	2	0	100	-
DerisiGO	MBM	0.001	1000	3	0	-	-
DerisiGO	MBM-T	0.001	500	3	0	20	1e-9
DerisiGO	MHM	0.001	1750	3	0	-	-
DerisiGO	MHM-T	0.001	500	3	0	15	1e-7
DerisiGO	MVM	0.001	1750	2	0.5	-	-
DerisiGO	MVM-T	0.001	1000	3	0.3	30	1e-3
Diatoms	BoxE	0.001	1750	2	0	-	-
Diatoms	C-HMCNN	0.0005	1000	3	0.3	100	-
Diatoms	MBM	0.001	1750	3	0	-	-
Diatoms	MBM-T	0.001	1750	3	0	100	1e-7
Diatoms	MHM	0.001	500	3	0	-	-
Diatoms	MHM-T	0.001	500	3	0	100	1e-7
Diatoms	MVM	0.001	1750	2	0	-	-
Diatoms	MVM-T	0.001	1000	3	0.3	100	1e-7
Enron	BoxE	0.001	500	3	0	-	-
Enron	C-HMCNN	0.001	500	3	0	100	-
Enron	MBM	0.001	1750	3	0	-	-
Enron	MBM-T	0.001	1750	3	0.5	100	1e-7
Enron	MHM	0.001	1750	3	0.3	-	-
Enron	MHM-T	0.001	1750	3	0.3	100	1e-7
Enron	MVM	0.001	500	3	0	-	-
Enron	MVM-T	0.001	1000	3	0.3	100	1e-3
ExprFUN	BoxE	0.001	1750	3	0.5	-	-
ExprFUN	C-HMCNN	0.001	1750	3	0.5	100	-
ExprFUN	MBM	0.001	1750	3	0.5	-	-
ExprFUN	MBM-T	0.001	500	3	0.3	100	1e-7
ExprFUN	MHM	0.001	1750	3	0.3	-	-
ExprFUN	MHM-T	0.001	1750	3	0.3	100	1e-3
ExprFUN	MVM	0.001	1750	3	0.5	-	-
ExprFUN	MVM-T	0.001	1000	3	0.3	100	1e-7
ExprGO	BoxE	0.001	1750	3	0.3	-	-
ExprGO	C-HMCNN	0.001	500	3	0	100	-
ExprGO	MBM	0.001	1000	3	0	-	-
ExprGO	MBM-T	0.001	500	3	0	20	1e-9
ExprGO	MHM	0.001	1750	3	0.3	-	-
ExprGO	MHM-T	0.001	500	3	0.3	15	1e-7
ExprGO	MVM	0.001	1750	3	0.3	-	-
ExprGO	MVM-T	0.001	1000	3	0.5	30	1e-7
Imclef07a	BoxE	0.0001	1000	3	0.5	-	-
imclef07a	C-HMCNN	0.0001	1000	3	0.5	100	-
Imclef07a	MBM	0.001	500	3	0.7	-	-
Imclef07a	MBM-T	0.001	500	3	0.7	100	1e-7
Imclef07a	MHM	0.001	1000	3	0	-	-
Imclef07a	MHM-T	0.001	1000	3	0	100	1e-7
Imclef07a	MVM	0.0001	1000	3	0.5	-	-
Imclef07a	MVM-T	0.0001	1000	3	0.5	100	1e-3
Imclef07d	BoxE	0.001	500	3	0	-	-
Imclef07d	C-HMCNN	0.001	500	3	0	100	-
Imclef07d	MBM	0.001	1750	3	0	-	-
Imclef07d	MBM-T	0.001	1750	3	0	100	1e-9
Imclef07d	MHM	0.001	1000	3	0.3	-	-
Imclef07d	MHM-T	0.001	1750	3	0.3	100	1e-7
Imclef07d	MVM	0.001	500	3	0	-	-
Imclef07d	MVM-T	0.001	1750	3	0.3	100	1e-3

Table 4 continued from previous page

Dataset	Model	learning rate	hidden dim	layers	dropout	% label sample	
SpoFUN	BoxE	0.001	1750	3	0.5	-	-
SpoFUN	C-HMCNN	0.001	1750	3	0.5	100	-
SpoFUN	MBM	0.001	1000	3	0.3	-	-
SpoFUN	MBM-T	0.001	500	3	0.3	100	1e-7
SpoFUN	MHM	0.001	1750	3	0	-	-
SpoFUN	MHM-T	0.001	1750	3	0	100	1e-7
SpoFUN	MVM	0.001	1750	3	0.5	-	-
SpoFUN	MVM-T	0.001	1000	3	0.3	100	1e-7
SpoGO	BoxE	0.001	1750	2	0.3	-	-
SpoGO	C-HMCNN	0.001	250	3	0.3	100	-
SpoGO	MBM	0.001	1000	3	0	-	-
SpoGO	MBM-T	0.001	500	3	0.3	20	1e-6
SpoGO	MHM	0.001	1750	3	0.5	-	-
SpoGO	MHM-T	0.001	1750	3	0.3	15	1e-7
SpoGO	MVM	0.001	1750	2	0.3	-	-
SpoGO	MVM-T	0.001	1000	3	0.3	30	1e-3

A.1 Further Details for the baselines

BoxE: Let $p \in \mathbb{R}^d$ be a point and $z, Z \in \mathbb{R}^d$ with $z_i < Z_i$ be the lower left and upper right coordinates of a non-probabilistic label box. Then a non-probabilistic compatibility score between the point and the box can be defined as shown in Abboud et al. (2020). Concretely, Abboud et al. (2020) define per-dimension distance as

$$\text{dist}(p_i, z_i, Z_i) := \begin{cases} \frac{|p_i - c_i|}{w_i + 1}, & \text{if } z_i \leq p_i \leq Z_i \\ \frac{|p_i - c_i|}{w_i + 1} - \kappa, & \text{otherwise,} \end{cases} \tag{8}$$

where $c_i = (Z_i - z_i)/2$, $w_i = Z_i - z_i$ are the center and width of the projection of the box (i.e., an interval) in dimension i, and $\kappa = 0.5 * w_i * (w_i + 1 - \frac{1}{w_i+1})$ is the width dependent factor. The score function that measures the compatibility of the point p and box (z, Z) is defined as $s_{\text{BoxE}}(p, z, Z) = -\|v\|_2$, where $v_i = \text{dist}(p_i, z_i, Z_i)$. Given the trainable parameters $\psi^-, \psi^+ \in \mathbb{R}^{L \times d}$ for representing the labels, the lower left and the upper right coordinates of a non-probabilistic box for label l_i are taken to be $z_i = \psi^-_{i,*}$ and $Z_i = \psi^-_{i,*} + \log(1 + \exp \psi^+_{i,*})$, respectively. The input x is encoded as a point $p \in \mathbb{R}^d$ using a multi-layer perceptron $\mathcal{F}_\theta$, just like MVM. However, since the score is negative distance, the parameters are learnt using margin based loss given as:

$$L_{\text{margin}}(\psi, \theta; D) = \sum_{i=1}^{D} \sum_{p \in \mathcal{P}_i, n \in \mathcal{N}_i} \frac{\max\left(0, 1 - (s_{\text{BoxE}}(\mathcal{F}_\theta(x), z_p, Z_p) - s_{\text{BoxE}}(\mathcal{F}_\theta(x), z_n, Z_n)\right)}{|\mathcal{P}_i| + |\mathcal{N}_i|},$$

where $\mathcal{P}_i$ and $\mathcal{N}_i$ are the positive and negative labels for i-th data instance, respectively.

A.2 Computational complexity

The computational complexity of the MBM can be divided into two parts–the computation of the instance box $\text{Box}_\theta(X)$, and the computation of probability score $P(Y \mid X)$ given the encoding of the instance and label box parameters. Computing $\text{Box}_\theta(X)$ amounts to splitting the output of a feed-forward network $\phi(x)$ into min and max parameters $\phi(x)^-, \phi(x)^+$, and thus has the same complexity as that of the MVM. Assuming that the operations $\sum, \prod$, $\log$ and $\exp$ have unit complexity, computing $P(Y \mid X)$ also has equivalent complexity. The computation of conditional probability for a single label y_k using the MVM model involves computing $\sigma(\phi(x) \cdot y_k)$, an $\mathcal{O}(dL)$ operation. Now, with same assumptions for the MBM, computing $\lambda(\text{Box}_\theta(x) \cap \text{Box}_\psi(y_k))$ involves d invocations of $\log(\exp(\cdot) + \exp(\cdot))$ followed by d subtractions and d invocations of $\log(1 + \exp(.))$, resulting in $\mathcal{O}(d)$ operations. The computation of $\lambda(\text{Box}_\theta(x))$ omits the intersection calculation, but otherwise is the same, and thus is also $\mathcal{O}(d)$. Hence, calculating $\frac{\lambda(\text{Box}(x) \cap \text{Box}(y_k))}{\lambda(\text{Box}(x))}$ is $\mathcal{O}(d)$, resulting in overall complexity of $\mathcal{O}(dL)$, which is equal to that of MVM. We note that the complexity of our

model is not dependent on the depth of the hierarchy, as in C-HMCNN, and thus can reasonably scale to arbitrarily deep hierarchies as might be present in extreme multi-label classification. These theoretical statements are supported by Table 5, which provides average epoch duration in seconds for MVM, MBM, MBM-T, and C-HMCNN with the same hidden size. Hence, we can conclude that the computational complexity of MVM $\approx$ MBM $<$ MBM-T.

Table 5: Average epoch duration in seconds for MVM, MBM, MBM-T and CHMCNN with the same hidden size

Model	ExprFUN	CellcycleFUN	SpoFUN	DerisiFUN
MVM	17.2	15.4	15.8	15.5
MBM	18.4	16.9	16.3	15.2
MBM-T	20.1	26.7	24.1	27.1
C-HMCNN	17.8	17.0	15.9	15.4
BoxE	18.9	16.6	15.9	16.1

Computational resources used: For datasets with number of labels less than 500, i.e., the 4 FUNCAT datasets, Imclef07a, Imclef07d, Diatoms and Enron, all the models were trained on TitanX GPU (memory=12GB). For the 4 GO datasets that have number of labels greater than 4000, all the models are trained on M40 GPU (memory=24GB).

A.3 Code

Completely anonymized and executable code with detailed instructions is provided using Anonymous Github at `anonymous.4open.science/r/modeling-label-space-interactions-in-multi-label-classification-using-box-embeddings`. The instructions cover all the use cases, i.e., obtaining the pre-processed datasets, training a new model from scratch (MBM or any baseline), evaluating a pre-trained model on test set, directly downloading the pre-trained models for datasets, and reproducing the graphs reported in the analysis section.

B Datasets

The 12 datasets used in this work wary greatly in terms of domain, number of labels, number of instances, and the connectivity of label taxonomy. The characteristics of each dataset w.r.t these properties is summarized in 6. These datasets do not require a licence and are available for public usage. The links to the sources for all the datasets are provided in Table 7 below. The continuous features are standardized by removing mean and scaling to unit variance, and the categorical features are encoded as one-hot vectors.

C Significance Testing

In section 6, we provide the results for Friedman test followed by post-hoc Nemenyi test, and also the result of pairwise Wilcoxon test where the former are not sufficient, i.e. for the case of MVM vs MBM w.r.t MAP. For the sake of completeness, here in table 8, we provide the complete table of pairwise Wilcoxon test from comparing all models with MBM and MBM-T w.r.t all three metrics.

D Comparison to C-HMCNN and HMCN

In our experiments with the official code of C-HMCNN(Giunchiglia & Lukasiewicz, 2020) as well as our re-implementation of the same, we found that C-HMCNN consistently gets lower MAP compared to a well tuned MVM (note that the base MVM results are not reported in the C-HMCNN or HMCN paper). In Giunchiglia & Lukasiewicz (2020), C-HMCNN was shown to outperform HMCN(Wehrmann et al., 2018c). Based on these observations, we considered a well-tuned MVM to be the stronger baseline and only included the results of C-HMCNN (our implementation) in table

Table 6: Summary of the datasets used in experiments. The feature based multi-label datasets span across 3 domains: functional genomics, image and text.

Dataset	Domain	Input/Feature Type	Label Taxonomy	#Labels	#Instances		
					Train	Val	Test
Expr FUN	Genomics	Continuous	Forest	500	1636	849	1288
Cellcycle FUN	Genomics	Continuous	Forest	500	1628	848	1281
Derisi FUN	Genomics	Continuous	Forest	500	1608	842	1275
Spo FUN	Genomics	Continuous	Forest	500	1600	837	1266
Expr GO	Genomics	Continuous	DAG	4132	1636	849	1288
Cellcycle GO	Genomics	Continuous	DAG	4126	1625	848	1278
Derisi GO	Genomics	Continuous	DAG	4120	1605	842	1272
Spo GO	Genomics	Continuous	DAG	4120	1597	837	1263
Diatoms	Image	Continuous	Tree	399	1500	565	1054
Imclef07a	Image	Continuous	Tree	97	7000	3000	1006
Imclef07d	Image	Continuous	Tree	47	7000	3000	1006
Enron	Text	Binary	Tree	57	650	338	600

Table 7: The table provides the links to download the data from original source.

Dataset(s)	Download Links
Imclef07a, Imclef07d, Enron, Diatoms	`http://kt.ijs.si/DragiKocev/PhD/resources/doku.php?id=hmc_classification`
Expr, Spo, Derisi, Cellcycle (FUN/GO)	`https://dtai.cs.kuleuven.be/clus/hmcdatasets/`

Table 8: The table presents the p-values of the two-sided Wilcoxon signed-rank test performed between MBM/MBM-T and the baselines following the Critical diagram presented in Figure 2. The Wilcoxon test is used to test the null hypothesis that two related paired samples come from the same distribution. The test considers the ranking as well as the differences in performance between the models across the 12 datasets.

Metric		BoxE	MVM	MHM	MBM	MVM-T	MHM-T	MBM-T	C-HMCNN
MAP	MBM	0.0048	0.109	0.0004	-	0.0004	0.00048	0.026	0.0024
	MBM-T	0.016	0.469	0.0004	0.026	0.0004	0.00048	-	0.0092
CMAP	MBM	0.0092	0.0004	0.0004	-	0.00048	0.0004	0.0209	0.0024
	MBM-T	0.016	0.0004	0.0004	0.0209	0.0004	0.0004	-	0.0068
CV	MBM	0.0004	0.0014	0.0209	-	0.0004	0.0522	0.0034	0.0004
	MBM-T	0.0004	0.0004	0.30	0.0034	0.0004	0.0639	-	0.0004

1. Table 9 compares our models to the C-HMCNN and HMCN models. Here, the columns with * are taken from their respective papers and these models use the union of train and validation set to train their final model. However, in all of our experiments, we only train on the train set to obtain the final test performance. As seen from the table, except for Imclef07a, MBM performs better than HMCN and C-HMCNN (even with lesser training data).

E MAP AND CMAP

In MCL problems, where predictions consistent with a taxonomy are necessary, the model designer might be ready to sacrifice some absolute predictive performance for increased coherence, where the predictive performance can be measure using Mean Average Precision (MAP). First recall that Average Precision (AP) for an instance is the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight, and Mean Average Precision (MAP) is the mean of AP across instances. Given data D, let $s_i^{(k)} = P(y_i = 1|x^{(k)})$ denote the score generated by the model for label i given the input $x^{(k)}$. Then MAP is computed as

Table 9: The table presents the area under the precision-recall curve (AU(PRC))comparing MVM and MBM to C-HMCNN (Giunchiglia & Lukasiewicz, 2020) and HMCN-F (Wehrmann et al., 2018b). Here, the columns with * are taken from their respective papers, C-HMCNN** is the result of running the code provided with the paper (Giunchiglia & Lukasiewicz, 2020) and C-HMCNN is implementation of C-HMCNN model in our codebase.

Dataset	**C-HMCNN***	**C-HMCNN****	**HMCN-F***	**C-HMCNN**	**MVM**	**MHM**	**MBM**	**BoxE**
ExprFUN	30.2	27.66	30.1	37.58	41.08	29.71	38.88	15.78
CellcycleFUN	25.5	24.02	25.2	35.06	35.2	31.43	36.07	15.39
DerisiFUN	19.5	18.91	19.3	30.48	31.95	30.38	32.6	17.21
SpoFUN	21.5	20.25	21.1	25.85	32.53	30.62	33.45	25.04
ExprGO	30.2	30.23	30.1	50.44	51.67	51.03	51.16	36.98
CellcycleGO	25.5	25.50	25.2	46.73	49.15	42.01	49.51	43.29
DerisiGO	19.5	18.91	19.3	45.39	46.59	45.13	47.28	31.35
SpoGO	21.5	21.53	21.1	45.66	46.98	44.03	47.33	17.24
Enron	75.6	75.29	72.4	79.20	80.49	75.85	78.38	74.90
Diatoms	75.8	66.36	53.0	73.18	74.69	44.69	82.8	22.97
Imclef07a	95.6	95.63	95.0	73.17	73.27	62.32	74.01	66.93
Imclef07d	92.7	92.77	95.0	83.97	82.59	66.54	84.26	69.18

$$\text{MAP}(s) = \frac{1}{|D|} \sum_{k=1}^{|D|} \text{AP}\left(s^{(k)}, y^{(k)}\right).$$

As described in Section 6.1, given the scores $s \in \mathbb{R}^L$ produced by a model, for an input, the two post-hoc modification functions that can be applied to make these score coherent are given as

$$\delta_G^m(s)_i = \min_{l_j \in \text{Anc}_G(l_i) \cup \{l_i\}} s_j, \qquad \delta_G^M(s)_i = \max_{l_j \in \text{Des}_G(l_i) \cup \{l_i\}} s_j, \tag{9}$$

where $\text{Anc}_G(l)$ and $\text{Des}_G(l)$ is the set of ancestors and descendants, respectively, of l in the graph G.

One can compute MAP after applying either one of the two modification functions. If a human were to make consistent score corrections based on the complete hierarchy, they would either pick the score of a label to be the minimum of its ancestors or maximum of its descendants. Once picked, the same approach has to be applied to all the labels. We argue that in terms of correlation with human judgment, both these methods are equivalent. Hence, one can pick the modification function based on empirical performance of the post-hoc correction algorithm. We observe that δ_G^m, i.e., "min of ancestors" approach, produced higher MAP with both MVM and MBM models compared to δ_G^M. as can be seen in table 10. We believe that "min of ancestors" works better because it promotes sparsity in predictions, which is essential for the task of MLC.

Table 10: CMAP represents post-hoc correction of scores by taking the "minimum of ancestors", CMAP' represents post-hoc correction of scores by taking "maximum of descendants". As seen, CMAP produces better results for both MBM and MVM compared to CMAP'.

Dataset	**CMAP**	**CMAP'**	**MAP**
Expr	46.47	26.53	49.11
Cellcycle	42.50	17.58	42.80
Derisi	39.50	17.23	40.09
Spo	40.07	16.75	40.12
Enron	75.65	66.07	83.76
Diatoms	71.21	67.81	78.40
Imclef07a	73.87	43.58	73.97

F PERFORMANCE ON LONG TAIL OF LABELS

Since MVM and MBM both are embedding based models, we expect the performance of the MBM model to wary with the label frequency in the same manner as MVM. This hypothesis is corroborated by the following table which shows the spearman rank correlation between the label frequency and the mean average precision (MAP) for that label. The high positive correlation for all the models suggests that the performance of both MVM and MBM often degrades as the frequency of the labels drop. However, as shown by the table 11, both the models perform reasonably well for extremely low frequency labels. Here, tail_MAP is the mean average precision for labels with frequency 0.001% or lower.

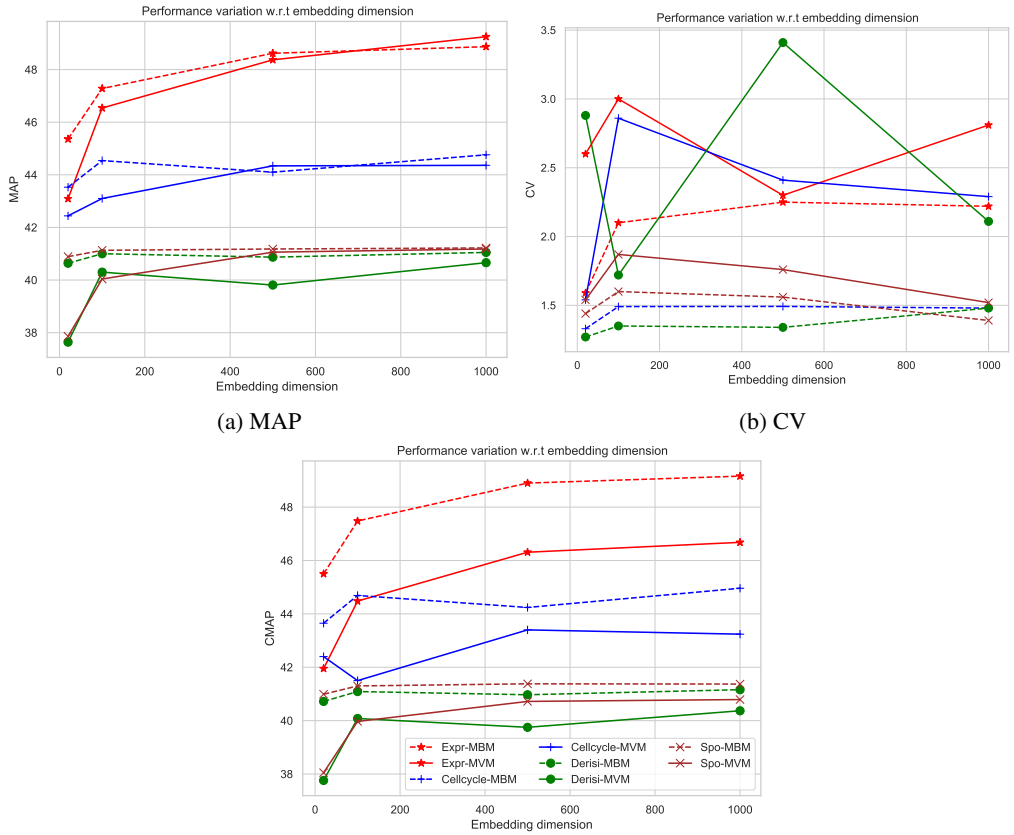
Table 11: Comparing the performance of long tails of labels for MBM, MBM-T and MVM. Here, correlation is the spearman rank correlation between label frequency and MAP of the label and tail_MAP is the MAP for labels with frequency 0.001% or lower.

Dataset	MBM		MBM-T		MVM	
	Correlation	tail_MAP	Correlation	tail_MAP	Correlation	tail_MAP
ExprFUN	0.72	48.91	0.73	48.68	0.75	49.45
CellcycleFUN	0.74	46.11	0.75	46.17	0.82	44.02
DerisiFUN	0.79	41.86	0.82	41.62	0.79	41.78
SpoFUN	0.8	42.67	0.77	42.75	0.79	42.15
Enron	0.84	79.19	0.83	74.97	0.91	67.63
Diatoms	0.29	86.3	0.29	85.46	0.28	82.2
Imclef07a	0.73	59.59	0.74	60.2	0.82	58.78

G EFFECT OF EMBEDDING DIMENSION

To study the effect of embedding dimensions on the performance, we plot the MAP, CMAP and CV vs the hidden dimension size for a subset of the datasets. As shown in figure 4, at extremely low dimensions, the MBM model consistently outperforms the MVM. This shows that the favorable inductive bias of the box space induces parsimonious representations.

H EFFECT OF HIERARCHY ON THE MARGINAL SCORE

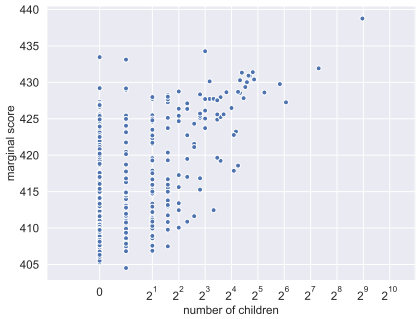


(a) MAP

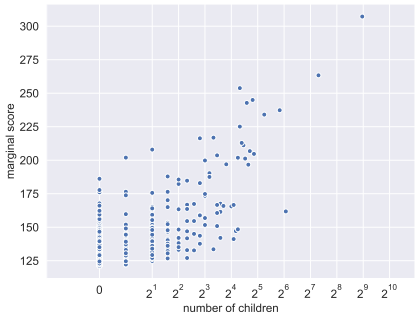
(b) CV

(c) CMAP

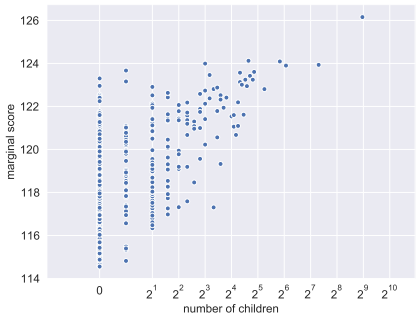
Figure 4: Above figure shows the variation of performance w.r.t the embedding dimensions for model MBM and MVM on FUNCAT datasets.



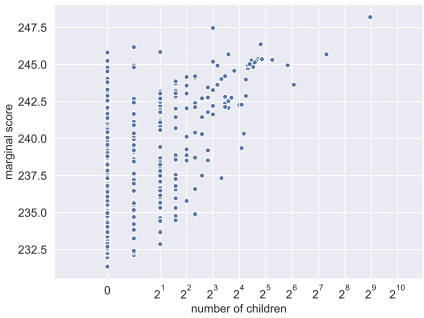
(a) Expr FUN



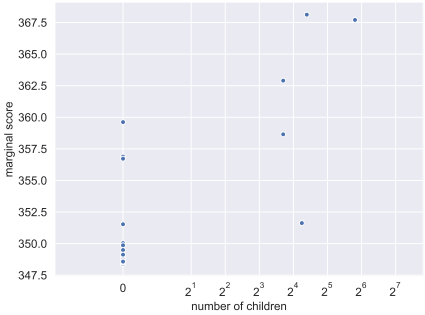
(b) Cellcycle FUN



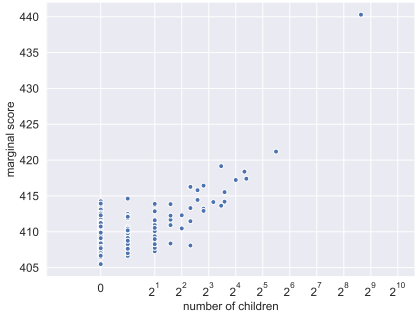
(c) Derisi FUN



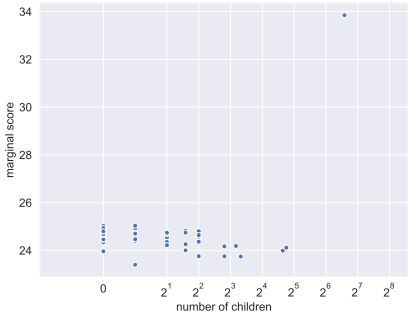
(d) Spo FUN



(e) Enron



(f) Diatoms



(g) Imclef07a

Figure 5: As discussed in Section 7, the scatter plots further show that the magnitude of box embeddings(box volume) relate to the level of generality in taxonomy.In the figure, the marginal score refers to box volume of a label and the number of children in the label space taxonomy refers to the level of generality.