HOT-VAE: Learning High-Order Label Correlation for Multi-Label Classification via Attention-Based Variational Autoencoders

Wenting Zhao, [1] **Shufeng Kong,** [1] **Junwen Bai,** [1] **Daniel Fink,** [2] **Carla Gomes** [1]
[1] Department of Computer Science, Cornell University, USA
[2] Cornell Lab of Ornithology, Ithaca, NY, USA
wzhao@cs.cornell.edu, {sk2299, jb2467, daniel.fink}@cornell.edu, gomes@cs.cornell.edu

Abstract

Understanding how environmental characteristics affect biodiversity patterns, from individual species to communities of species, is critical for mitigating effects of global change. A central goal for conservation planning and monitoring is the ability to accurately predict the occurrence of species communities and how these communities change over space and time. This in turn leads to a challenging and long-standing problem in the field of computer science - how to perform accurate multi-label classification with hundreds of labels? The key challenge of this problem is its exponential-sized output space with regards to the number of labels to be predicted. Therefore, it is essential to facilitate the learning process by exploiting correlations (or dependency) among labels. Previous methods mostly focus on modelling the correlation on label pairs; however, complex relations between real-world objects often go beyond second order. In this paper, we propose a novel framework for multi-label classification, High-order Tie-in Variational Autoencoder (HOT-VAE), which performs adaptive high-order label correlation learning. We experimentally verify that our model outperforms the existing state-of-the-art approaches on a bird distribution dataset on both conventional F1 scores and a variety of ecological metrics. To show our method is general, we also perform empirical analysis on seven other public real-world datasets in several application domains, and Hot-VAE exhibits superior performance to previous methods.

Introduction

The study of multi-label classification (MLC) is an active research area and has been receiving increasing attention in the past few decades; unlike traditional single-output learning, it is a task of predicting the presence and absence of multiple entities simultaneously given a sample with a set of features. It finds applications in a wide range of domains including image recognition, natural language processing, and bioinformatics (Xu et al. 2020).

One important field that is in urgent need for a scalable and accurate MLC approach is ecology. The ability to accurately predict which species assemble into communities based on local environmental features is essential to understand how changes in the environment can be expected to impact biodiversity and to plan for the restoration and recovery of species communities in the face of environmental change (D'Amen et al. 2017). This problem is represented as joint species distribution modelling (JSDM), which predicts species occurrences given environmental features and species interactions. There are two key challenges in JSDM. First, species communities are often comprised of very large numbers of individual species, presenting the challenge of learning complex high-dimensional interactions. For example, bird communities are often comprised hundreds of individual species within a single region. To consider how all subsets of all bird species interact is computationally intractable, thus being selective in how we model these interactions is a necessity. On one hand, accounting for more interactions increases learning capacity but will be more computationally demanding and could have a higher risk of over-fitting. On the other hand, one can focus on less or no interactions; however, many communities of species are known to include complex interactions among large numbers of species, so having oversimplifying assumptions leads to inaccurate predictions. The second challenge for modeling the joint distribution of many species is the fact that the relationships between species change as important features of the environment change over space and time. Thus, it is critical to understand how environment changes like climate change impact species interactions and the resulting structure of species communities (Evans et al. 2016).

Our contributions: We propose High-order Tie-in Variational Autoencoder (HOT-VAE), an attentioned-based VAE that leverages latent embedding learning and neural message passing to perform high-order label correlation learning and produce accurate multi-label predictions. More specifically: (1) We introduce a two-branch VAE-based model with a replaceable, domain-specific encoder (i.e., one can choose an encoder that efficiently extracts feature information given an application domain) and a shared message passing neural network (MPNN) decoder where label correlations are computed. (2) HOT-VAE is able to learn high-order label correlation using multiple-step message passing. It also produces label correlation conditioned on features. In other words, HOT-VAE can not only model the interactions between many species simultaneously, it can also adapt the correlation between species to changing environmental factors. (3) With a graph structure to reason about label corre-

lation, we can easily incorporate prior knowledge, resulting in better empirical results. (4) We perform thorough experimental evaluations on a JSDM dataset and seven other real-world datasets on a variety of metrics, and we show HOT-VAE outperforms (or is comparable to) other state-of-the-art MLC methods. On the JSDM dataset, we further evaluate HOT-VAE on several ecological metrics; the result suggests that HOT-VAE produces a meaningful improvement in the field of ecology.

Related Work

We discuss three groups of MLC methods and how they are related to our approach. The first group is binary relevance (BR) methods which treat a MLC problem as a number of independent binary classification problems (Boutell et al. 2004; Zhang and Zhou 2007). To add label correlation, probabilistic classifier chains (PCCs) stack binary classifiers sequentially and output one label at a time conditioned on all previously predicted labels (Read, Pfahringer, and Holmes 2008; Cheng, Hüllermeier, and Dembczynski 2010). Followup works extend PCCs to recurrent neural networks (Wang et al. 2016; Nam et al. 2017). This group suffers from two issues: the quality of predictions can be highly dependent on label ordering, and the nature of autoregressive models prohibits them from parallel computation.

The second group deals with latent embedding, in which they learn one shared latent space representing both input features and output labels (Bhatia et al. 2015a; Yeh et al. 2017; Tang et al. 2018; Chen et al. 2019a). Most recently, Bai, Kong, and Gomes (2020) propose MPVAE: it learns VAE-based probabilistic latent spaces for both labels and features and aligns the latent representations using the Kullback–Leibler divergence. These methods impose label-aware structure on the feature latent space, which is empirically showed to produce better predictive performance. However, they only consider up to second-order label correlation: MPVAE's decoder is an multivariate probit model (MVP) (Chen, Xue, and Gomes 2018) which employs a covariance matrix on labels to capture pairwise relations, and Bhatia et al. (2015a) find k-nearest neighbors for label embeddings. Another limitation is that these label correlations are global: when features change, the label interactions remain the same. It is also not clear how they can incorporate prior knowledge on label structures.

The third group models label dependencies using graphical model representations (Lafferty, McCallum, and Pereira 2001; Chen et al. 2019b). Methods within this group often build a label graph, in which a node corresponds to a label, and edges represent how two labels interact with each other. Most recently, Lanchantin, Sekhon, and Qi (2019) propose LaMP, where they apply the attention mechanism from Transformer (Vaswani et al. 2017) to learn how other labels contribute to the presence/absence of a label. Further, they use an MPNN, a generalization of graph neural networks (GNN) (Scarselli et al. 2008), to pass messages among label nodes weighted by attention, thus modelling a conditional joint representation of output labels. Compared to PCC methods, LaMP provides a scalable and flexible module to model label correlations that requires no label ordering and allows parallel computation, and its graphical structure making it a natural fit to impose constraints on labels. However, LaMP follows a encoder-to-decoder architecture, and learning feature embeddings within this architecture has not yet been optimized. It does not incorporate any label information in the feature embedding, which has been shown to be beneficial for making accurate predictions (Yeh et al. 2017).

This work leverages both the state-of-the-art latent embedding learning and powerful attention-based MPNNs to provide an accurate and scalable multi-label classifier, and we further extend the message passing module to model high-order label correlations to improve performance.

Background

Let $\mathcal{D}$ denote the dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^S$ is an input and $\boldsymbol{y}_i \in \{0,1\}^L$ is an output associated with sample i. Input $\boldsymbol{x}_i$ can alternatively be an ordered set of S elements, and output $\boldsymbol{y}_i$ has L binary labels with 1 indicating the presence and 0 indicating the absence of a label.

Aligned Variational Autoencoders

A variational autoencoder (VAE) is a generative model which consists of an encoder, a decoder, and a loss function. The encoder, denoted by $q_\theta(\boldsymbol{z}_i|\boldsymbol{x_i})$, is a neural network that maps features of samples $\boldsymbol{x_i}$ into hidden representations $\boldsymbol{z}$, which have a much lower dimensionality than that of $\boldsymbol{x_i}$. $\boldsymbol{z}_i$ represents a multivariate Gaussian probability density, and by sampling from this distribution we obtain noisy values of $\boldsymbol{z}_i$. The decoder, denoted by $p_\phi(\boldsymbol{x_i}|\boldsymbol{z}_i)$, is another neural network that reconstructs $\boldsymbol{z}$ to $\boldsymbol{x_i}$. The loss function to be minimized is $\mathbb{E}_{\boldsymbol{z}\sim q_\phi}[\log p_\theta(\boldsymbol{x_i}|\boldsymbol{z})] - \mathcal{K}[q_\phi(\boldsymbol{z}|\boldsymbol{x_i})||P(\boldsymbol{z})]$, where $\mathcal{K}$ is the Kullback–Leibler (KL) divergence, and $P(\boldsymbol{z})$ is the prior which is a standard multivariate normal distribution. The first term encourages the reconstruction of $\boldsymbol{x_i}$, and the second term penalizes the KL divergence between the approximated distribution $q_\theta(\boldsymbol{z}|\boldsymbol{x_i})$ and the prior $P(\boldsymbol{z})$ which imposes structure on the latent space.

VAE can be used for the task of classification if the decoder is to predict target y_i instead of reconstructing feature x_i. Therefore, we rewrite the decoder and loss function as $p_\phi(\boldsymbol{y_i}|\boldsymbol{z_x})$ and $\mathbb{E}_{\boldsymbol{z_x}\sim q_\phi}[\log p_\theta(\boldsymbol{y_i}|\boldsymbol{z_x})] - \mathcal{K}[q_\phi(\boldsymbol{z_x}|\boldsymbol{x_i})||P(\boldsymbol{z_x})]$, respectively. In this case, it is desired that the prior $P(z_x)$ imposes domain-specific structure on the latent space rather than just imposing the conventional standard multivariate normal structure. Recently, MPVAE was proposed to use another VAE to learn a latent multivariate Gaussian distribution $q_\psi(\boldsymbol{z_y}|\boldsymbol{y}_i)$, and align $q_\psi(\boldsymbol{z_y}|\boldsymbol{y}_i)$ and $q_\phi(\boldsymbol{z_x}|\boldsymbol{x}_i)$ by penalizing their KL divergence (Bai, Kong, and Gomes 2020). The feature and label VAEs are designed to share the same decoder. Thus, the label decoder is $p_\theta(y_i|z_y)$, and the loss function can be revised as:

$$\begin{aligned} KL =& \frac{1}{2}(\mathbb{E}_{\boldsymbol{z_y}\sim q_\phi}[\log p_\theta(\boldsymbol{y}_i|\boldsymbol{z_y})] + \mathbb{E}_{\boldsymbol{z_x}\sim q_\psi}[\log p_\theta(\boldsymbol{y}_i|\boldsymbol{z_x})]) \\ & - \beta\mathcal{K}[q_\psi(\boldsymbol{z_y}|\boldsymbol{y}_i)||q_\phi(\boldsymbol{z_x}|\boldsymbol{x}_i)], \end{aligned}$$

where β is a hyper-parameter to control the similarity between the two latent Gaussian distributions.

Attention Models

Self-attention, sometimes called *intra-attention*, is a mechanism that assigns different importance to different positions of a sequence in order to focus on more important parts. Self-attention has been used successfully in a variety of tasks such as natural language processing (NLP) and MLC. The Transformer and the Generative Pre-trained Transformer 3 (GPT-3) (Brown et al. 2020) are two well-known attention models in NLP achieving state-of-the-art performance. The recent Label Message Passing (LaMP) (Lanchantin, Sekhon, and Qi 2019) can be regarded as an application of the Transformer/GPT-3 on multi-label classification.

LaMP alternates between self-attention and feed forward layers. In the t-th self-attention layer, each label is represented by a node $v_i^t \in \mathbb{R}^d$. The attention weight a_{ij}^t for a node pair $(\boldsymbol{v}_i, \boldsymbol{v}_j)$ is computed as:

$$e_{ij}^t = a(\boldsymbol{v}_i^t, \boldsymbol{v}_j^t) = \frac{(\mathbf{W}^q \boldsymbol{v}_i^t)^\top (\mathbf{W}^u \boldsymbol{v}_j^t)}{\sqrt{d}} \tag{1}$$

$$\alpha_{ij}^t = \text{softmax}_j(e_{ij}^t) = \frac{\exp(e_{ij}^t)}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^t)}, \tag{2}$$

where $a(\cdot)$ is a dot product with node-wise linear transformations $\mathbf{W}^q \in \mathbb{R}^{d \times d}$ on node $\boldsymbol{v}_i^t$ and $\mathbf{W}^u \in \mathbb{R}^{d \times d}$ on node $\boldsymbol{v}_j^t$, scaled by $\sqrt{d}$; e_{ij}^t represents the raw importance of label j to label i and is further normalized by a softmax function to obtain α_{ij}^t. Then, the attention m_i^t message of v_i^t is generated as:

$$M_{\text{atn}}(\boldsymbol{v}_i^t, \boldsymbol{v}_j^t) = \alpha_{ij}^t \mathbf{W}^v \boldsymbol{v}_j^t, \tag{3}$$

$$\boldsymbol{m}_i^t = \boldsymbol{v}_i^t + \sum_{j \in \mathcal{N}(i)} M_{\text{atn}}(\boldsymbol{v}_i^t, \boldsymbol{v}_j^t), \tag{4}$$

where $\mathbf{W}^v \in \mathbb{R}^{d \times d}$ is a node-wise linear transformation.

After going through the t-th feed forward layer U^t, we obtain v_i^{t+1} in the $(t+1)$-th self-attention layer as:

$$\boldsymbol{v}_i^{t+1} = \boldsymbol{m}_i^t + U^t(\boldsymbol{m}_i^t; \boldsymbol{W}). \tag{5}$$

Method: Attention-based VAE for High-order Correlation

We propose HOT-VAE, a novel two-branch variational autoencoder model building on top of attention-based neural message passing networks for MLC, which can learn feature embeddings representing both input features and output labels, perform high-order correlation learning, and flexibly incorporate prior knowledge on label structures. The illustration of the framework is shown in Figure 1. At training, the feature encoder and the label encoder first map features and labels to a set of Gaussian subspaces respectively. There are many possible ways to parameterize the encoders; common choices are multi-layer perceptrons and graph networks such as graph convolutional networks and message passing neural networks. Then, the shared decoder makes a prediction on labels based on the samples from the feature Gaussian subspace and recovers the input labels based on the samples from the label Gaussian subspace. There are two message passing modules. One passes attention from features to labels, and the other passes attention from labels to labels.

Learning and Aligning Probabilistic Subspaces

We assume that both the feature embedding and the label embedding have d dimensions. If each encoder only outputs one Gaussian subspace, in the case of it being an MLP, $\mathcal{D}[q_\phi(\mathbf{z}|\mathbf{y})||q_\psi(\mathbf{z}|\mathbf{x})]$ is simply the KL divergence between two multivariate normal distributions. Since both distributions have diagonal covariance matrices, we can derive the KL divergence to be the following:

$$\begin{aligned}\mathcal{L}_{\text{KL}}(\boldsymbol{x}, \boldsymbol{y}) = &\beta[\sum_{i=1}^{d} \log \frac{\Sigma_{i,i}^{\psi}(\boldsymbol{x})}{\Sigma_{i,i}^{\phi}(\boldsymbol{y})} - d + \sum_{i=1}^{d} \frac{\Sigma_{i,i}^{\phi}(\boldsymbol{y})}{\Sigma_{i,i}^{\psi}(\boldsymbol{x})} + \\ &\sum_{i=1}^{d} \frac{(\mu_i^{\psi}(\boldsymbol{x}) - \mu_i^{\phi}(\boldsymbol{y}))^2}{\Sigma_{i,i}^{\psi}(\boldsymbol{x})}]\end{aligned} \tag{6}$$

However, there are features which MLPs cannot encode, such as English texts and graphs. To have a general multi-label classifier, encoders need to be flexible with regards to network architecture to deal with different types of inputs. Then, it becomes possible that the feature encoder and the label encoder output different numbers of Gaussian subspaces. For example, if we adopt the Transformer encoder (Vaswani et al. 2017) to be the feature encoder and the label encoder, the two encoders may generate two sets of Gaussian subspaces of varying sizes. This is because for every dimension of the input, a mean and a variance are computed, and features and labels often differ in their dimensionality. For example, let us consider when the input text is seven words - "my favorite football player lost a game" and the labels are "sad" and "angry". The feature transformer outputs one Gaussian subspace for each of the seven words, and the label transformer outputs two subspaces. To overcome this issue, suppose the feature encoder outputs J subspaces and the label encoder outputs K subspaces, we compress J subspaces and K subspaces into one subspace by computing a mean vector for μ and for Σ, respectively. Formally, $\mu^{\psi}(\boldsymbol{x})$ and $\Sigma^{\psi}(\boldsymbol{x})$ now become:

$$\mu^{\psi}(\boldsymbol{x}) = \frac{1}{J} \sum_{j=0}^{J-1} \mu^{\psi}(\boldsymbol{x})^{(j)} \tag{7}$$

$$\Sigma^{\psi}(\boldsymbol{x}) = \frac{1}{J} \sum_{j=0}^{J-1} \Sigma^{\psi}(\boldsymbol{x})^{(j)} \tag{8}$$

And we do the same for $\mu^{\phi}(\boldsymbol{y})$ and $\Sigma^{\phi}(\boldsymbol{y})$. With these operations, we can again use Equation 6 to compute the divergence between two probabilistic latent spaces. It is worth noting that because there is not a one-to-one relationship from one feature Gaussian subspace to one label Gaussian subspace, there is no point in aligning individual feature subspaces to individual label subspaces.

Lastly, although we collapse all Gaussian subspaces into a unified one to compute alignment, we still feed unmodified $\{\boldsymbol{z}_{\boldsymbol{x}}^{(j)}\}$ and $\{\boldsymbol{z}_{\boldsymbol{y}}^{(k)}\}$ into the shared decoder to keep as much information as possible.

Learning High-Order Label Correlation

We highlight three features of the shared decoder: (1) The decoder computes the correlation between labels condi-

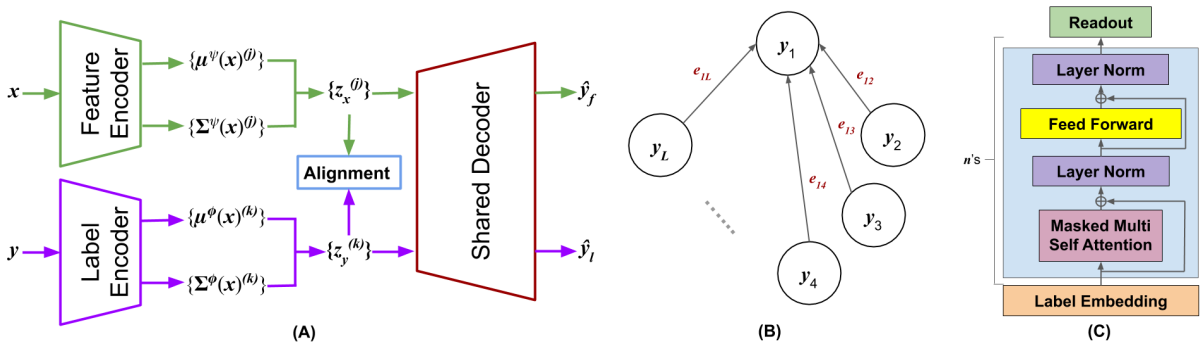

Figure 1: Model architecture of HOT-VAE. (A): Overall network architecture. The feature encoder maps features x to a set of probabilistic latent subspaces using a neural network parameterized by ψ. Similarly, the label encoder with parameter ϕ maps labels y to another set of probabilistic latent subspaces. Then, sampled from their own distributions, $\{z_x^{(j)}\}$ and $\{z_y^{(k)}\}$ are separately fed into a shared decoder. Finally, for the feature branch, the decoder outputs label prediction $\hat{y}_f$, and for the label branch, the decoder outputs reconstruction $\hat{y}_l$. At inference, only the feature branch is used. (B): The shared decoder is a graph with each node being a label. An edge is connecting two nodes if we believe correlation exists between them. By default, there is an edge between every pair of labels. The figure shows all the other nodes are sending messages to the y_1 node with learned attention weights e. (C): Decoder layers. At each layer, masked multi-head self-attention is computed and goes through a feed-forward operation. n is the number of layers. We note that these layers are shared by all label nodes.

tioned on features $\boldsymbol{x}$; thus, the label correlation becomes sensitive to changes in $\boldsymbol{x}$, which enables adaptive learning for label interactions. (2) With the label graph, one can easily impose prior structure between labels by adding and removing edges. If it is known in advance two labels are independent from each other, then leaving out the edge connecting these two label nodes prevents the model from over-fitting and learning noise. (3) Most importantly, the decoder is able to capture higher-order label correlation in a scalable way, as opposed to PCC methods (Wang et al. 2016; Nam et al. 2017) where chain rules are used to model the joint probabilities of labels.

We now turn to a detailed description of HOT-VAE's decoder. Labels are represented as embedded vectors $\{\boldsymbol{u}_1^t, \boldsymbol{u}_2^t, \ldots, \boldsymbol{u}_L^t\}$, where $\boldsymbol{u}_i^t \in \mathbb{R}^d$ and initial $\boldsymbol{u}_i^{t=0}$ is obtained from a learnable embedding matrix $\mathbf{W}^y \in \mathbb{R}^{L\times d}$. Figure 1(C) shows an overview of the decoder layers. Self-attention is computed based on Equations 1-4, and feed forward is computed using Equation 5. Specifically, we use multi-head self-attention (Vaswani et al. 2017), so that a node can attend to multiple other nodes at once. We also apply layer normalization (Ba, Kiros, and Hinton 2016) around each of the attention and feedforward sublayers to alleviate training issues. After the n's layers (thus $\boldsymbol{u}_i^{t=0}$ becoming $\boldsymbol{u}_n^t$), a readout layer predicts each label $\hat{y}_i$, where a readout function R projects $\boldsymbol{u}_i^n$ using a projection matrix $\mathbf{W}^o \in \mathbb{R}^{d\times d}$. The ith row of $\mathbf{W}^o$ is denoted by $\mathbf{W}_i^o$. The resulting vector of size $L \times 1$ is then fed through an element-wise sigmoid function to produce the final probabilities of all labels:

$$\hat{y}_i = R(\boldsymbol{u}_i^n; \mathbf{W}^o) = \text{sigmoid}(\mathbf{W}_i^o \boldsymbol{u}_i^n). \tag{9}$$

As mentioned above, to pass the message from encoder to decoder, we feed unmodified $\{\boldsymbol{z}_{\boldsymbol{x}}^{(j)}\}$ and $\{\boldsymbol{z}_{\boldsymbol{y}}^{(k)}\}$ into the shared decoder to keep as much information as possible. For clarity, we look at the feature branch, and the label branch works similarly. We denote the message passing module from $\{\boldsymbol{z}_{\boldsymbol{x}}^{(j)}\}$ to label nodes by $\boldsymbol{W}_{\text{fy}}$ and the message passing module between label nodes by $\boldsymbol{W}_{\text{yy}}$. We compute the initial state $\boldsymbol{u}_i^{t'}$ of the decoder w.r.t. the feature branch as:

$$\boldsymbol{m}_i^t = \boldsymbol{u}_i^t + \sum_{j=1}^{J} M_{\text{atn}}(\boldsymbol{u}_i^t, \boldsymbol{z}_{\boldsymbol{x}}^{(j)}; \boldsymbol{W}_{\text{fy}}), \tag{10}$$

$$\boldsymbol{u}_i^{t'} = \boldsymbol{m}_i^t + U_{\text{mlp}}(\boldsymbol{m}_i^t; \boldsymbol{W}_{\text{fy}}). \tag{11}$$

After $\boldsymbol{u}_i^t$ is updated to $\boldsymbol{u}_i^{t'}$ with feature information, the message passing between labels then begins,

$$\boldsymbol{m}_i^{t'} = \boldsymbol{u}_i^{t'} + \sum_{l\in\mathcal{N}(i)} M_{\text{atn}}(\boldsymbol{u}_i^{t'}, \boldsymbol{u}_l^{t'}; \boldsymbol{W}_{\text{yy}}), \tag{12}$$

$$\boldsymbol{u}_i^{t+1} = \boldsymbol{m}_i^{t'} + U_{\text{mlp}}(\boldsymbol{m}_i^{t'}; \boldsymbol{W}_{\text{yy}}). \tag{13}$$

We can also incorporate prior knowledge for label-to-label message passing by simple pre-processing: starting from a complete label graph where every node is connected to every other node, we can remove the edge between a pair of nodes if they never belong to any sample simultaneously in the training set. Depending on a dataset's domain, additional expert knowledge from the domain can be enforced on a dataset-to-dataset basis. For instance, for predicting species distribution, one can incorporate evolutionary relationships between species, often represented as tree-structures (Ovaskainen et al. 2017; Letunic and Bork 2007).

Finally, we describe how HOT-VAE learns higher-order correlation between labels. If there is a single layer in the decoder, then messages are passed once from labels to labels, which computes correlation between any pair of labels. Going to higher order, we only need to increase the num-

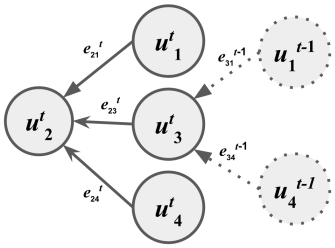

Figure 2: An example of higher-order correlation between labels. u_i^t is the hidden representation of y_i at time t. The figure shows how correlation is formed within the label triplets $\{y_2, y_3, y_1\}$ and $\{y_2, y_3, y_4\}$.

ber of layers in the decoder. n times of message passing between labels enable learning $(n+1)$-order label correlation. In Figure 2, we present a visual explanation, showing how correlation is learned for two label triplets $\{\boldsymbol{y}_2, \boldsymbol{y}_3, \boldsymbol{y}_1\}$ and $\{\boldsymbol{y}_2, \boldsymbol{y}_3, \boldsymbol{y}_4\}$. In this example, at time $t-1$, label 3 collects information from labels 1, 4. At time t, label 2 further collects information from label 3. By this time, two paths has been built from $1 \to 3 \to 2$ and $4 \to 3 \to 2$. Therefore, the presence of label 2 conditions on the label pairs (1, 3) and (4, 3). This can be easily extended to high orders.

Loss Function

The whole model can be trained in an end-to-end fashion with the Adam optimizer (Kingma and Ba 2015). The overall loss function consists of four parts. We denote the true binary label vector by $\boldsymbol{y}$. For both the feature and the label branches, we compute binary cross entropy (BCE) over all outputs $\boldsymbol{y}_i$ for every sample:

$$\mathcal{L}_{\text{BCE}} = (BCE(\boldsymbol{y}, \hat{\boldsymbol{y}}_f) + BCE(\boldsymbol{y}, \hat{\boldsymbol{y}}_l)) \tag{14}$$

where BCE is defined by:

$$BCE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{L}\sum_{i=1}^{L} -(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i))$$

Further, since the decoder iteratively updates the label node from $t=0$ to n, we can also pass those intermediate states from $t=1$ to $n-1$ through a readout layer and enforce BCE loss on these states.

$$\mathcal{L}_{\text{INT}} = (\sum_{t=1}^{n-1} BCE(\boldsymbol{y}, \hat{\boldsymbol{y}}_f^t) + \sum_{t=1}^{n-1} BCE(\boldsymbol{y}, \hat{\boldsymbol{y}}_l^t)) \tag{15}$$

We also include ranking loss (Zhang and Zhou 2014) defined as follows, which is shown beneficial in many multi-label tasks:

$$RL(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{|Y||\bar{Y}|} \sum_{(r,s)\in(Y,\bar{Y})} \exp(-\hat{y}_r - \hat{y}_s)$$

where Y is the set of indices for true positive labels and $\bar{Y}$ is the set of indices for true negative labels. $\hat{y}_r$ and $\hat{y}_s$ are the corresponding r-th and s-th probabilities outputted by the model. The ranking loss penalizes when a relevant label to the sample is ranked higher than an irrelevant label. Again, the ranking loss is calculated for both two branches:

$$\mathcal{L}_{\text{RANK}} = (RL(\boldsymbol{y}, \hat{\boldsymbol{y}}_f) + RL(\boldsymbol{y}, \hat{\boldsymbol{y}}_l)) \tag{16}$$

Finally, with the KL divergence computed for the two branches included, the overall loss function becomes:

$$\mathcal{L} = \lambda_0 \mathcal{L}_{\text{BCE}} + \lambda_1 \mathcal{L}_{\text{INT}} + \lambda_2 \mathcal{L}_{\text{RANK}} + \beta \mathcal{L}_{\text{KL}} \tag{17}$$

$\lambda_0, \lambda_1, \lambda_2$ and β controls the weights of the four loss terms.

Experiments

We illustrate the power of HOT-VAE on eight real-world datasets covering a variety of application domains including ecology, images, texts, etc. We first present the main experiment, where HOT-VAE is compared to several other state-of-the-art MLC methods on all the datasets. We evaluate their performance with conventional metrics such as F-measure and accuracy. To verify that our model also makes a meaningful improvement in ecology, we perform analysis on 12 metrics measuring discrimination power, calibration, etc. for levels of species occurrence, species richness, and community composition (Norberg et al. 2019). Finally, we present ablation studies showing the benefits of having high-order label correlation and incorporating prior knowledge.

Setup

Datasets. The datasets we use to run experiments are: *eBird* (Chen et al. 2017), a crowd-sourced bird presence-absence dataset collected from birders' observations; *bibtex* and *bookmarks* (Katakis, Tsoumakas, and Vlahavas 2008), collections of text objects associated with tags; *mir-flickr* (Huiskes and Lew 2008) and *scene* (Boutell et al. 2004), collections of images with tags; *reuters* (Lewis et al. 2004), natural language texts with predefined categories based on their content; *sider* (Kuhn et al. 2016), side effects of drug molecules; and *yeast* (Nakai and Kanehisa 1992), a biology database of the protein localization sites. They are all available online [1] [2].

These datasets vary in many aspects including the number of samples ranging from 1427 to 87856, number of labels from 6 to 208, feature dimensions from 15 to 368998. They also cover a wide spectrum of input types: some are raw English text with words ordered sequentially, some are binary features, and the other are real-value vectors (e.g., images). We present label statistics for each dataset in Table 1, which is useful information to consider when designing models. One can see that it is common for samples to have more than two labels. For instance, the median number of labels per sample in *eBird* is 18, which suggests incorporating high-order label correlation into a model will lead to a stronger modelling capacity to learn better joint representations.

We split each dataset into a training set, a validation set, and a test set in the same way as Bai, Kong, and Gomes (2020) and Lanchantin, Sekhon, and Qi (2019) do.

[1] http://mulan.sourceforge.net/datasets-mlc.html

[2] https://ebird.org/home

	Median Labels /Sample	Max Labels /Sample	Median Samples /Label	Max Samples /Label
eBird	18	96	5793	29340
bookmarks	1	44	381	4642
bibtex	2	28	54	689
mirflickr	5	17	799	4120
reuters	1	15	18	2877
scene	1	3	168	903
sider	16	26	851	1185
yeast	4	11	334	903

Table 1: Dataset Label Statistics. This shows that many samples have more than two labels to be predicted, and each dataset has a varying degree of label density.

Baseline Comparisons. SLEEC (Bhatia et al. 2015b) makes no low-rank assumption, and it learns embbeddings perserving pairwise distances between only the nearest label vectors. C2AE (Yeh et al. 2017) is a two-branch autoencoder. It first produces a latent vector for features and a latent vector for labels; then these two latent embedding are associated by deep canonical correlation analysis (DCCA). seq2seq (Nam et al. 2017) applies a recurrent-neural-neural (RNN) based encoder-to-decoder model where the encoder RNN encodes features and the decoder predicts each positive label sequentially. LaMP (Lanchantin, Sekhon, and Qi 2019) consists of multiple attention-based neural message passing modules including one sending messages bewteen features, one from features to labels, and one between labels. MPVAE (Bai, Kong, and Gomes 2020) is a two-branch disentangled VAE building on a covariance-aware multivariate probit model, which can learn pairwise label correlation.

Other details. We choose two encoders: one is a three-layer MLP which is used for *eBird*, *scene*, *sider*, *sider*, and *yeast*; the other one is a two-layer FMP (Lanchantin, Sekhon, and Qi 2019) for the other datasets. For decoders, we use both complete and prior graphs (with edges removed if two labels never co-occur in any training sample) for each dataset and select the one exhibiting better performance.

Main Experiments

In Table 2, we present the main experimental result evaluated on three F1-scores: example-based F1 (ebF1), micro-averaged F1 (miF1), and macro-averaged F1 (maF1). F1-score is the harmonic mean of precision and recall of the predictions. ebF1 is the average of the F1-score for each test sample. miF1 aggregates total true positives, false negatives, and false positives for all class labels, and computes a F1-score. maF1 computes the F1-score independently for each class and returns the average with equal weights for all classes. Larger F1-scores indicate better performance, with the highest possible value being 1, meaning perfect precision and recall. We note that high ebF1s show strong results over all test samples, high miF1s indicate strong performance on the most frequent labels, and high maF1 implies strong performance on less frequent labels. We select the model performing best on validation set based on maF1.

Dataset	SLEEC	C2AE	seq2seq	LaMP	MPVAE	ours
eBird	0.2578	0.5007	0.4768	0.4768	0.5511	**0.5747**
bookmarks	-	-	0.3620	0.3551	-	**0.3630**
bibtex	0.4490	0.3346	0.3930	0.4469	0.4534	**0.4693**
mirflickr	0.4163	0.5011	0.4216	0.4918	0.5138	**0.5192**
reuters	-	-	0.8944	0.9060	-	**0.9128**
scene	0.7184	0.6978	0.7469	0.7279	0.7505	**0.7762**
sider	0.5807	0.7682	0.3560	0.7662	0.7687	**0.7708**
yeast	0.6426	0.6142	0.5744	0.6242	0.6479	**0.6498**
Dataset	SLEEC	C2AE	seq2seq	LaMP	MPVAE	ours
eBird	0.4124	0.5459	0.5260	0.5170	0.5933	**0.6270**
bookmarks	-	-	0.3290	0.3593	-	**0.3682**
bibtex	0.4074	0.3884	0.3840	0.4733	0.4800	**0.4823**
mirflickr	0.4127	0.5448	0.4640	0.5352	0.5516	**0.5559**
reuters	-	-	0.8575	0.8890	-	**0.8910**
scene	0.6993	0.7131	0.7442	0.7156	0.7422	**0.7567**
sider	0.6965	0.7978	0.3890	0.7977	0.8002	**0.8026**
yeast	0.6531	0.6258	0.5999	0.6407	0.6554	**0.6595**
Dataset	SLEEC	C2AE	seq2seq	LaMP	MPVAE	ours
eBird	0.3625	0.4260	0.3298	0.3806	0.4936	**0.5350**
bookmarks	-	-	0.2370	0.2939	-	**0.2984**
bibtex	0.2937	0.2680	0.2820	0.3763	0.3863	**0.3953**
mirflickr	0.3636	0.3931	0.3333	0.3871	**0.4217**	0.4078
reuters	-	-	0.4567	0.5600	-	**0.5748**
scene	0.6990	0.7284	0.7490	0.7449	0.7504	**0.7639**
sider	0.5917	0.6674	0.2070	0.6684	**0.6904**	0.6653
yeast	0.4251	0.4272	0.4333	0.4802	0.4817	**0.4885**

Table 2: Top: ebF1 scores; Middle: miF1 scores; Bottom: maF1 scores produced by all the methods for each dataset. We mark the best scores with bold texts.

Dataset	SLEEC	C2AE	seq2seq	LaMP	MPVAE	ours
eBird	0.8156	0.7712	0.8236	0.8113	0.8286	**0.8455**
bookmarks	-	-	0.9900	**0.9917**	-	**0.9917**
bibtex	0.9818	0.9867	0.9850	0.9876	0.9875	**0.9878**
mirflickr	0.8698	0.8973	0.8839	0.8969	0.8978	**0.8980**
reuters	-	-	0.9962	0.9970	-	**0.9971**
scene	0.8937	0.8934	**0.9456**	0.9025	0.9094	0.9155
sider	0.6750	0.7487	0.5930	0.7510	0.7547	**0.7555**
yeast	0.7824	0.7635	**0.8177**	0.7857	0.7920	0.7947

Table 3: HAs for all the methods on each dataset.

We note that *reuters* deals with sequential input and *bookmarks* has over 380 thousand features; hence, we only compare to seq2seq and LaMP, which can handle position information or inputs with extremely high dimensions. HOT-VAE outperforms all the baseline approaches on ebF1, yielding a 2.13% improvement to MPVAE and a 5.69% improvement to LaMP on average. Our model again produces the best performance within all the methods on miF1, improving MPVAE by 1.63% and LaMP by 4.88% on average. HOT-VAE performs less well for predicting the rare labels on *sider* and *mirflickr*, indicated by maF1, but it still produces an improvement on most of the datasets.

We also test HOT-VAE on Hamming accuracy (HA). Table 3 summaries the HAs on each dataset. seq2seq performs better than HOT-VAE on *scene* and *yeast*, because they directly maximize subset accuracy, and this works well when #labels is small. Otherwise, HOT-VAE has the best HAs.

	Occurrences				Richness				Community			
	A	D	C	P	A	D	C	P	A	D	C	P
MPVAE	0.3	0.7	679.8	0.3	18.3	0.4	0.4	4.0	(0.2, 0.2, 0.2)	(0.4, 0.1, 0.0)	(0.3, 0.3, 0.3)	**(0.0, 0.0, 0.0)**
HOT-VAE	**0.2**	**0.8**	**124.9**	**0.2**	**10.8**	**0.6**	**0.2**	**3.0**	**(0.1, 0.2, 0.1)**	**(0.5, 0.4, 0.3)**	**(0.2, 0.2, 0.2)**	(0.1, 0.1, 0.1)

Table 4: HOT-VAE vs. MPVAE on 12 ecological metrics. For accuracy (A), calibration (C), and precision (P), scores are the smaller, the better; for discrimination (D), scores are the larger, the better. The better scores are marked with bold texts.

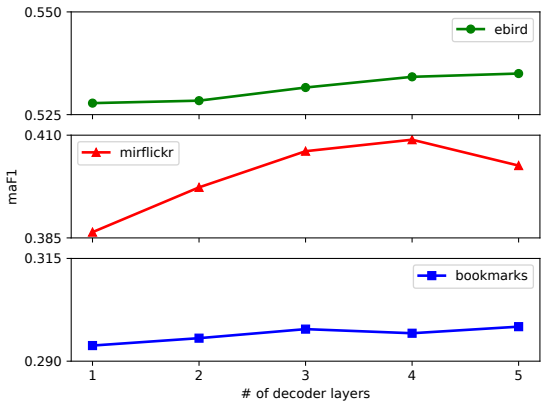

Figure 3: maF1 scores for the datasets *ebird*, *mirflickr*, *bookmarks* when the number of decoder layers increases.

Measuring Predictive Performance on Species

Ecologists use joint species distribution models for a number of distinct applications. To determine if HOT-VAE produces ecologically meaningful improvements in model performance we evaluated the 12 metrics presented in Norberg et al. (2019) used to assess performance predicting species occurrences, species richness, and community composition. The occurrence metrics measure a model's performance for predicting presence/absence of individual species, the richness metrics measure the ability to predict the total number of species that occur at a given location and time, and the community metrics measure the ability to predict species occurrences at location pairs. Four metrics are applied for predicting each of species occurrences, species richness, and community composition: accuracy, calibration, precision, and discriminative power of predictions. We compare our model to MPVAE, the best performing baseline method.

The experimental results for the 12 measures are shown in Table 4. For accuracy, calibration, and precision, smaller values indicate better predictive performance; for discrimination power, larger values are better. HOT-VAE outperforms MPVAE for 11 of the 12 metrics. We only have worse precision at the community composition level; this is because this community composition metric only considers pair-wise co-occurrence of species, and MPVAE is optimized for modeling interactions specifically between pairs of species.

Ablation Studies

We further investigate how incorporating high-order label correlation impacts predictive performance. We postulate that, for learning n-order label correlation, increasing n would be particularly beneficial for improving predictions of rare labels: when treating classifying each label as individual problem, there is very limited information to learn from for the rare labels; including higher-order interactions between labels provides much additional information to a model.

	ebird	*scene*	*reuters*	*sider*	*bibtext*
prior	**0.8259**	**0.9471**	**0.9915**	**0.5151**	0.9405
complete	0.8251	0.9355	0.9882	0.5120	**0.9457**

Table 5: medianAUC for each dataset when decoder is on a complete label graph and on a prior label graph.

To verify this hypothesis, we train HOT-VAE with different numbers of decoder layers and see how maF1 scores vary under these settings, as high maF1 scores indicate strong results on less frequent labels. We select three datasets *ebird*, *mirflickr*, and *bookmarks*, which have median labels 18, 5, and 1, respectively. In Figure 3, we show maF1 scores for these datasets at n being 1-5. For the datasets with more labels per sample, larger n has more positive impacts on maF1 scores. It is also possible increasing n beyond some threshold negatively affects performance as the model may start to overfit. In general, for the datasets with dense labels, having $n = 4, 5$ produces best results.

Additionally, we evaluate the effect of incorporating prior knowledge. We train HOT-VAE on both complete label graphs and prior label graphs, in which we remove the edges between the label pairs if they never positively correspond to any training sample. To have one metric summarizing the overall performance, we use medianAUC (described in the supplementary material), which represents the probability that a random positive sample is ranked higher than a random negative sample. In Table 5, we show medianAUC for each dataset when HOT-VAE is trained with these two label graphs respectively. We see that using prior graphs slightly improves predictive performance in many cases and is at least comparable to using complete graphs in all cases.

Conclusion

In this paper, we propose an attention-based HOT-VAE for multi-label classification to address the complex relations between real-world objects. HOT-VAE learns high-order correlation between labels conditioned on features; in other words, not only can it captures relations within multiple objects, but also the relations are adaptive to any feature change. Experimental results show that HOT-VAE improves over the state-of-the-art techniques.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. We thank Di Chen and Yiwei Bai for the helpful discussions. We thank the eBird participants for their contributions and the eBird team for their support. Daniel Fink was funded by The Leon Levy Foundation, The Wolf Creek Foundation, and the National Science Foundation (ABI sustaining: DBI-1939187). The other authors were supported by NSF awards CCF-1522054 (Expeditions in computing) and CNS-1059284 (Infrastructure), AFOSR Multidisciplinary University Research Initiatives (MURI) Program FA9550-18-1-0136, ARO award W911NF-17-1-0187, and an award from the Toyota Research Institute.

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* .

Bai, J.; Kong, S.; and Gomes, C. 2020. Disentangled Variational Autoencoder based Multi-Label Classification with Covariance-Aware Multivariate Probit Model. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 4313–4321. doi:10.24963/ijcai.2020/595. Special track on AI for CompSust and Human well-being.

Bhatia, K.; Jain, H.; Kar, P.; Varma, M.; and Jain, P. 2015a. Sparse local embeddings for extreme multi-label classification. In *Advances in neural information processing systems*, 730–738.

Bhatia, K.; Jain, H.; Kar, P.; Varma, M.; and Jain, P. 2015b. Sparse Local Embeddings for Extreme Multi-label Classification. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28*, 730–738. Curran Associates, Inc. URL http://papers.nips.cc/paper/5969-sparse-local-embeddings-for-extreme-multi-label-classification.pdf.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern recognition* 37(9): 1757–1771.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* .

Chen, C.; Wang, H.; Liu, W.; Zhao, X.; Hu, T.; and Chen, G. 2019a. Two-stage label embedding via neural factorization machine for multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3304–3311.

Chen, D.; Xue, Y.; Fink, D.; Chen, S.; and Gomes, C. P. 2017. Deep multi-species embedding. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 3639–3646.

Chen, D.; Xue, Y.; and Gomes, C. 2018. End-to-End Learning for the Deep Multivariate Probit Model. In *International Conference on Machine Learning*, 932–941.

Chen, Z.-M.; Wei, X.-S.; Wang, P.; and Guo, Y. 2019b. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5177–5186.

Cheng, W.; Hüllermeier, E.; and Dembczynski, K. J. 2010. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 279–286.

D'Amen, M.; Rahbek, C.; Zimmermann, N. E.; and Guisan, A. 2017. Spatial predictions at the community level: from current approaches to future frameworks. *Biological Reviews* 92(1): 169–187.

Evans, D.; Che-Castaldo, J.; Crouse, D.; Davis, F.; Epanchin-Niell, R.; Flather, C.; Frohlich, R.; Goble, D.; Li, Y.-W.; Male, T.; Master, L.; Moskwik, M.; Neel, M.; Noon, B.; Parmesan, C.; Schwartz, M.; Scott, J.; and Williams, B. 2016. Species recovery in the United States: increasing the effectiveness of the Endangered Species Act. *Issues in Ecology* 20: 1–28.

Huiskes, M. J.; and Lew, M. S. 2008. The MIR Flickr Retrieval Evaluation. In *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*. New York, NY, USA: ACM.

Katakis, I.; Tsoumakas, G.; and Vlahavas, I. 2008. Multilabel Text Classification for Automated Tag Suggestion. *ECML PKDD Discovery Challenge 2008* 75.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*. URL http://arxiv.org/abs/1412.6980.

Kuhn, M.; Letunic, I.; Jensen, L. J.; and Bork, P. 2016. The SIDER database of drugs and side effects. *Nucleic acids research* 44(D1): D1075–D1079.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289.

Lanchantin, J.; Sekhon, A.; and Qi, Y. 2019. Neural message passing for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 138–163. Springer.

Letunic, I.; and Bork, P. 2007. Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation. *Bioinformatics* 23(1): 127–128.

Lewis, D. D.; Yang, Y.; Rose, T. G.; and Li, F. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research* 5(Apr): 361–397.

Nakai, K.; and Kanehisa, M. 1992. A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics* 14(4): 897–911.

Nam, J.; Mencía, E. L.; Kim, H. J.; and Fürnkranz, J. 2017. Maximizing subset accuracy with recurrent neural networks in multi-label classification. In *Advances in neural information processing systems*, 5413–5423.

Norberg, A.; Abrego, N.; Blanchet, F. G.; Adler, F. R.; Anderson, B. J.; Anttila, J.; Araújo, M. B.; Dallas, T.; Dunson, D.; Elith, J.; et al. 2019. A comprehensive evaluation of predictive performance of 33 species distribution models at species and community levels. *Ecological Monographs* 89(3): e01370.

Ovaskainen, O.; Tikhonov, G.; Norberg, A.; Guillaume Blanchet, F.; Duan, L.; Dunson, D.; Roslin, T.; and Abrego, N. 2017. How to make more out of community data? A conceptual framework and its implementation as models and software. *Ecology Letters* 20(5): 561–576.

Read, J.; Pfahringer, B.; and Holmes, G. 2008. Multi-label classification using ensembles of pruned sets. In *2008 eighth IEEE international conference on data mining*, 995–1000. IEEE.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20(1): 61–80.

Tang, L.; Xue, Y.; Chen, D.; and Gomes, C. P. 2018. Multi-Entity Dependence Learning With Rich Context via Conditional Variational Auto-Encoder. In *AAAI*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; and Xu, W. 2016. CNN-RNN: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2285–2294.

Xu, D.; Shi, Y.; Tsang, I. W.; Ong, Y.; Gong, C.; and Shen, X. 2020. Survey on Multi-Output Learning. *IEEE Transactions on Neural Networks and Learning Systems* 31(7): 2409–2429.

Yeh, C.-K.; Wu, W.-C.; Ko, W.-J.; and Wang, Y.-C. F. 2017. Learning deep latent spaces for multi-label classification. *arXiv preprint arXiv:1707.00418* .

Zhang, M.; and Zhou, Z. 2014. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26(8): 1819–1837.

Zhang, M.-L.; and Zhou, Z.-H. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition* 40(7): 2038–2048.