General Multi-label Image Classification with Transformers

Jack Lanchantin, Tianlu Wang, Vicente Ordonez, Yanjun Qi
University of Virginia
{jjl5sw,tianlu,vicente,yq2h}@virginia.edu

Abstract

Multi-label image classification is the task of predicting a set of labels corresponding to objects, attributes or other entities present in an image. In this work we propose the Classification Transformer (C-Tran), a general framework for multi-label image classification that leverages Transformers to exploit the complex dependencies among visual features and labels. Our approach consists of a Transformer encoder trained to predict a set of target labels given an input set of masked labels, and visual features from a convolutional neural network. A key ingredient of our method is a label mask training objective that uses a ternary encoding scheme to represent the state of the labels as positive, negative, or unknown during training. Our model shows state-of-the-art performance on challenging datasets such as COCO and Visual Genome. Moreover, because our model explicitly represents the label state during training, it is more general by allowing us to produce improved results for images with partial or extra label annotations during inference. We demonstrate this additional capability in the COCO, Visual Genome, News-500, and CUB image datasets.

1. Introduction

Images in real-world applications generally portray many objects and complex situations. Multi-label image classification is a visual recognition task that aims to predict a set of labels corresponding to objects, attributes, or actions given an input image [18, 48, 50, 52, 6, 31, 10]. This task goes beyond the more thoroughly studied problem of single-label multi-class classification where the objective is to extract and associate image features with a single concept per image. In the multi-label setting, the output set of labels has some structure that reflect the structure of the world. For example, *dolphin* is unlikely to co-occur with *grass*, while *knife* is more likely to appear next to a *fork*. Effective models for multi-label classification aim to extract good visual features that are predictive of image labels, but also exploit the complex relations and dependencies between visual features and labels, and among labels themselves.

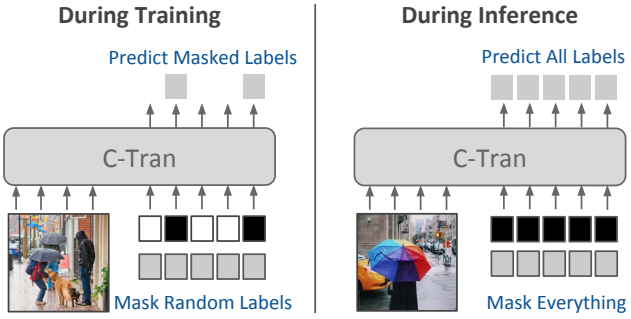

Figure 1. We propose a transformer-based model for multi-label image classification that exploits dependencies among a target set of labels using an encoder transformer. During training, the model learns to reconstruct a partial set of labels given randomly masked input label embeddings and image features. During inference, our model can be conditioned only on visual input or a combination of visual input and partial labels, leading to superior results.

To this end, we present the Classification Transformer (C-Tran), a multi-label classification framework that leverages a Transformer encoder [49]. Transformers have demonstrated a remarkable capability of being able to exploit dependencies among sets of inputs using multi-headed self-attention layers. In our approach, a Transformer encoder is trained to reconstruct a set of target labels given an input set of masked label embeddings and a set of features obtained from a convolutional neural network. C-Tran uses label masking during training to represent the state of the labels as *positive*, *negative*, or *unknown* – analogous to how language models are trained with masked tokens [15]. At test time, C-Tran is able to predict a set of target labels using only input visual features by masking all the input labels as *unknown*. Figure 1 gives an overview of this strategy. We demonstrate that this approach leads to superior results on a number of benchmarks compared to other recent approaches that exploit label relations using graph convolutional networks and other recently proposed strategies.

Beyond obtaining state-of-the-art results on standard multi-label classification, C-Tran is a more general model for reasoning under prior label observations. Because our approach explicitly models the label state (positive, nega-

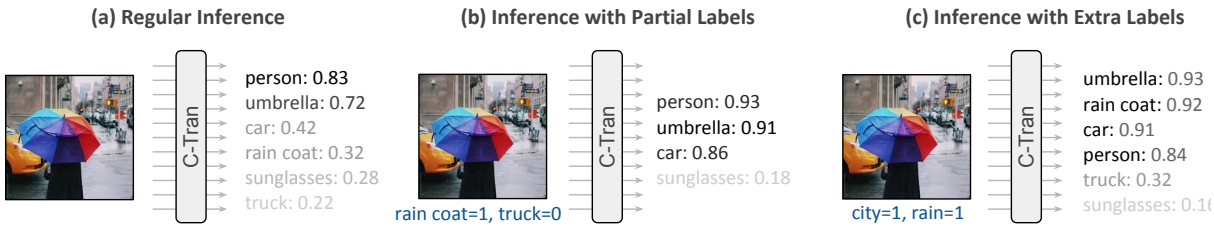

Figure 2. Different inference settings for general multi-label image classification: (a) Standard multi-label classification takes only image features as input. All labels are unknown $\mathbf{y}_u$.; (b) Classification under partial labels takes as input image features as well as a subset of the target labels that are known. The labels *rain coat* and *truck* are known labels $\mathbf{y}_k$, and all others are unknown labels $\mathbf{y}_u$; (c) Classification under extra labels takes as input image features and some related extra information. The labels *city* and *rain* are known extra labels $\mathbf{y}_k^e$, and all others are unknown target labels $\mathbf{y}_u^t$.

tive, or unknown) during training, it can also be used at test time with partial or extra label annotations by setting the state of some of the labels as either *positive* or *negative* instead of masking them out as *unknown*. For instance, consider the example shown in Figure 2(a) where a model is able to predict *person* and *umbrella* with relatively high accuracies, but is not confident for categories such as *rain coat*, or *car* that are clearly present. Suppose we know some labels and set them to their true positive (for *rain coat*) or true negative (for *truck*) values. Provided with this new information, the model is able to predict *car* with a high confidence as it moves mass probability from *truck* to *car*, and predicts other objects such as *umbrella* with even higher confidence than in the original predictions (Figure 2(b)). In general, we consider this setting as realistic since many images also have metadata in the form of extra labels such as location or weather information (Figure 2(c)). This type of conditional inference is a much less studied problem. C-Tran is able to naturally handle all these scenarios under a unified framework. We compare our results with a competing method relying on iterative inference [51], and against sensitive baselines, demonstrating superior results under variable amounts of partial or extra labels.

The benefits of C-Tran can be summarized as follows:

- Flexibility: It is the first model that can be deployed in multi-label image classification under arbitrary amounts of extra or partial labels. We use a unified model architecture and training method that lets users to apply our model easily in any setting.
- Accuracy: We evaluate our model on six datasets across three inference settings and achieve state-of-the-art results on all six. The label mask training strategy enhances the correlations between visual concepts leading to more accurate predictions.
- Interactivity: The use of state embeddings enables users to easily interact with the model and test any counterfactuals. C-Tran can take human interventions as partial evidence and provides more interpretable and accurate predictions.

2. Problem Setup

We consider three multi-label image classification scenarios as follows:

Regular Multi-label Classification. In this setting the goal is to predict a set of labels for an input image. Let $\mathbf{x}$ be an image, and $\mathbf{y}$ be a ground truth set of ℓ binary labels $\{y_1, y_2, ..., y_\ell\}, y_i \in \{0, 1\}$. The goal of multi-label classification is to construct a classifier, f, to predict a set of labels given an image so that: $\hat{\mathbf{y}} = f(\mathbf{x})$.

Inference with Partial Labels. While regular classification methods aim to predict the full set of ℓ labels given only an input image, some subset of labels $\mathbf{y}_k \subseteq \mathbf{y}$ may be observed, or known, at test time. This is also known as having partial labels available. For example, many images on the web are labeled with text such as captions or comments on social media. In this reformulated setting, the goal is to predict the unknown labels ($\mathbf{y}_u = \mathbf{y} \setminus \mathbf{y}_k$) given both the image *and* the known labels during inference: $\hat{\mathbf{y}}_u = f(\mathbf{x}, \mathbf{y}_k)$. Note that we assume that all labels are available during training. This setting is specifically for *inference* with partially annotated labels, and it differs from other works that tackle the problem of training models from partially annotated data [54, 17, 28].

Inference with Extra Labels. Similar to partially labeled images, there are many cases where we observe extra labels that describe an image, but are not part of the target label set. For example, we may know that an image was taken in a *city*. While *city* might not be one of the target labels, it can still alter our expectations about what else might be present in the image. In this setting, we append any extra labels $\mathbf{y}^e$ to the target label set $\mathbf{y}^t$. If there are ℓ^t target labels, and ℓ^e extra labels, we have a set of $\ell^t + \ell^e$ total labels that we use to train the model. Variable $\mathbf{y}$ now represents the concatenation of all target and extra labels. During inference, the known labels, $\mathbf{y}_k^e$, come from the set of extra labels, but we are only interested in evaluating the unknown target labels $\mathbf{y}_u^t$. In other words, during inference, we want to compute the following: $\hat{\mathbf{y}}_u^t = f(\mathbf{x}, \mathbf{y}_k^e)$.

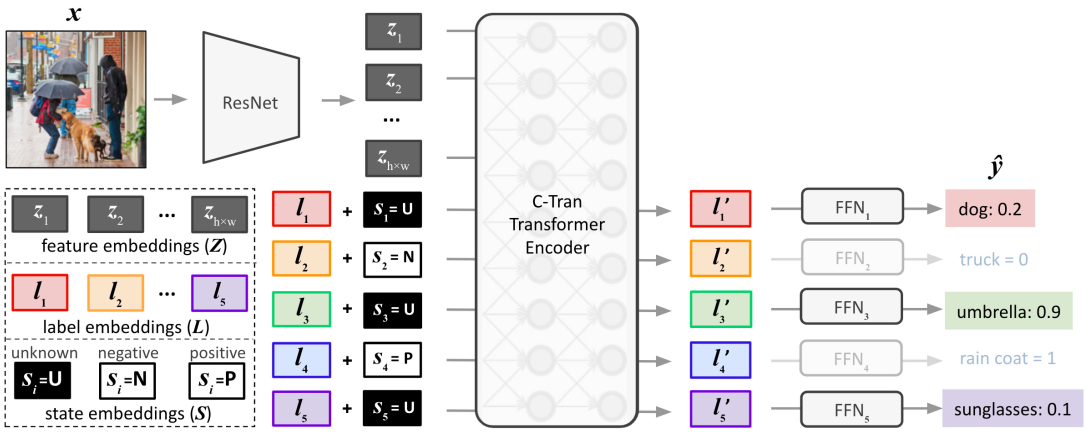

Figure 3. C-Tran architecture and illustration of label mask training for general multi-label image classification. In this training image, the labels *person*, *umbrella*, and *sunglasses* were randomly masked out and used as the unknown labels, $\mathbf{y}_u$. The labels *rain coat* and *truck* are used as the known labels, $\mathbf{y}_k$. Each unknown label is added the unknown state embedding U, and each known label is added its corresponding state embedding: negative (N) , or positive (P). The loss function is only computed on the unknown label predictions $\hat{\mathbf{y}}_u$.

3. Method: C-Tran

We propose Classification Transformers (C-Tran), a general multi-label classification framework that works in all three previously described settings. During inference, our method predicts a set of unknown labels $\mathbf{y}_u$ given an input image $\mathbf{x}$ and a set of known labels $\mathbf{y}_k$. In regular inference no labels are known, in partial label inference some labels are known, and in extra label inference some labels external to the target set are known. In Sections 3.1-3.3, we introduce the C-Tran architecture, and in Section 3.4, we explain our label mask training procedure.

3.1. Feature, Label, and State Embeddings

Image Feature Embeddings $\boldsymbol{Z}$: Given input image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$, the feature extractor outputs a tensor $\boldsymbol{Z} \in \mathbb{R}^{h\times w\times d}$, where h, w, and d are the output height, width, and channel, respectively. We can then consider each vector $\boldsymbol{z}_i \in \mathbb{R}^d$ from $\boldsymbol{Z}$, with i ranging from 1 to P (where $P = h \times w$), to be representative of a subregion that maps back to patches in the original image space.

Label Embeddings $\boldsymbol{L}$: For every image, we retrieve a set of label embeddings $\boldsymbol{L} = \{\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_\ell\}$, $\boldsymbol{l}_i \in \mathbb{R}^d$, which are representative of the ℓ possible labels in $\mathbf{y}$. Label embeddings are learned from an embedding layer of size $d \times \ell$.

Adding Label Knowledge via State Embeddings $\boldsymbol{S}$: In traditional architectures, there is no mechanism to encode partially known or extra labels as input to the model. To address this drawback, we propose a technique to easily incorporate such information. Given label embedding $\boldsymbol{l}_i$, we simply add a "state" embedding vector, $\boldsymbol{s}_i \in \mathbb{R}^d$:

$$\tilde{\boldsymbol{l}}_i = \boldsymbol{l}_i + \boldsymbol{s}_i, \tag{1}$$

where the $\boldsymbol{s}_i$ takes on one of three possible states: unknown (U), negative (N), or positive (P). For instance, if label y_i is a known positive value prior to inference (meaning that we have prior knowledge that the label is present in the image), $\boldsymbol{s}_i$ is the positive embedding, P. The state embeddings are retrieved from a learned embedding layer of size $d \times 3$, where the unknown state vector (U) is fixed with all zeros.

State embeddings enable a user to (1) not use any prior information by adding the unknown embedding, (2), use partially labeled or extra information by adding the negative and positive embeddings to those labels, and (3) easily test interventions in the model by asking "how does the prediction change if a label is changed to either positive or negative?". We note that using prior information is completely optional as input to our model during testing, enabling it to also flexibly handle the regular inference setting.

3.2. Modeling Feature and Label Interactions with a Transformer Encoder

To model interactions between image features and label embeddings we leverage Transformers [49], as these are effective models for capturing dependencies between variables. Our formulation allows us to easily input image features and label embeddings jointly into a Transformer encoder. Transformer encoders are suitable because they are order invariant, allowing for any type of dependencies between all features and labels to be learned.

Let $H = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_{h\times w}, \tilde{\boldsymbol{l}}_1, ..., \tilde{\boldsymbol{l}}_\ell\}$ be the set of embeddings that are input to the Transformer encoder. In Transformers, the importance, or weight, of embedding $\boldsymbol{h}_j \in H$ with respect to $\boldsymbol{h}_i \in H$ is learned through *self-attention*. The attention weight, α_{ij}^t between embedding i and j is computed in the following manner. First, we compute a normalized scalar attention coefficient α_{ij} between embeddings i and j. After computing α_{ij} for all i and j pairs, we update each embedding $\boldsymbol{h}_i$ to $\boldsymbol{h}_i'$ using a weighted sum of

all embeddings followed by a nonlinear ReLU layer:

$$\alpha_{ij} = \text{softmax}\big((\mathbf{W}^q \boldsymbol{h}_i)^\top (\mathbf{W}^k \boldsymbol{h}_j)/\sqrt{d}\big), \quad (2)$$

$$\bar{\boldsymbol{h}}_i = \sum_{j=1}^{M} \alpha_{ij} \mathbf{W}^v \boldsymbol{h}_j, \quad (3)$$

$$\boldsymbol{h}'_i = \text{ReLU}(\bar{\boldsymbol{h}}_i \mathbf{W}^r + \boldsymbol{b}_1)\mathbf{W}^o + \boldsymbol{b}_2, \quad (4)$$

where $\mathbf{W}^k$ is the key weight matrix, $\mathbf{W}^q$ is the query weight matrix, $\mathbf{W}^v$ is the value weight matrix, $\mathbf{W}^r$ and $\mathbf{W}^o$ are transformation matrices, and $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are bias vectors. This update procedure can be repeated for L layers where the updated embeddings $\boldsymbol{h}'_i$ are fed as input to the successive Transformer encoder layer. The learned weight matrices $\{\mathbf{W}^k, \mathbf{W}^q, \mathbf{W}^v, \mathbf{W}^r, \mathbf{W}^o\} \in \mathbb{R}^{d\times d}$ are not shared between layers. We denote the final output of the Transformer encoder after L layers as $H' = \{\boldsymbol{z}'_1, ..., \boldsymbol{z}'_{h\times w}, \boldsymbol{l}'_1, ..., \boldsymbol{l}'_\ell\}$.

3.3. Label Inference Classifier

Lastly, after feature and label dependencies are modeled via the Transformer encoder, a classifier makes the final label predictions. We use an independent feedforward network (FFN_i) for final label embedding $\boldsymbol{l}'_i$. FFN_i contains a single linear layer, where weight $\mathbf{w}_i^c$ for label i is a $1 \times d$ vector, and σ is a simoid function:

$$\hat{y}_i = \text{FFN}_i(\boldsymbol{l}'_i) = \sigma\big((\mathbf{w}_i^c \cdot \boldsymbol{l}'_i) + b_i\big) \quad (5)$$

3.4. Label Mask Training (LMT)

State embeddings (Eq. 1) let us easily incorporate known labels as input to C-Tran. However, we want our model to be flexible enough to handle any amount of known labels during inference. To solve this problem, we introduce a novel training procedure called Label Mask Training (LMT) that forces the model to learn label correlations, and allows C-Tran to generalize to any inference setting.

Inspired by the Cloze task [46] and BERT's masked language model training [15] which works by predicting missing words from their context, we implement a similar procedure. During training, we randomly *mask* a certain amount of labels, and use the ground truth of the other labels (via state embeddings) to predict the masked labels. This differs from masked language model training in that we have a fixed set of inputs (all possible labels) and we randomly mask a subset of them for each sample.

Given that there are ℓ possible labels, the number of "unknown" (i.e. masked) labels for a particular sample, n, is chosen at random between 0.25ℓ and ℓ. Then, n unknown labels, denoted $\mathbf{y}_u$, are sampled randomly from all possible labels $\mathbf{y}$. The unknown state embedding is added to each unknown label. The rest are "known" labels, denoted $\mathbf{y}_k$ and the corresponding ground truth state embedding (positive or negative) is added to each. We call these known labels because the ground truth value is used as input to C-Tran alongside the image. Our model predicts the unknown labels $\mathbf{y}_u$, and binary cross entropy is used to update the model parameters. By masking random amounts of unknown labels (and therefore using random amounts of known labels) during training, the model learns many possible known label combinations, and adapts the model to be used with arbitrary amounts of known information.

We mask out at least 0.25ℓ labels for each training samples for several reasons. First, most masked language model training methods mask out around 15% of the words [15, 4]. Second, we want our model to be able to incorporate anywhere from 0 to 0.75ℓ known labels during inference. We assume that knowing more than 75% of the labels is an unrealistic inference scenario. Our label mask training pipeline thus aims to minimize the following loss:

$$L = \sum_{n=1}^{N_{tr}} \mathbb{E}_{p(\mathbf{y}_k)}\{\text{CE}(\hat{\mathbf{y}}_u^{(n)}, \mathbf{y}_u^{(n)})|\mathbf{y}_k\}, \quad (6)$$

where CE represents the cross entropy loss function. $\mathbb{E}_{p(\mathbf{y}_k)}(\cdot|\mathbf{y}_k)$ denotes to calculate the expectation regarding the probability distribution of known labels: $\mathbf{y}_k$. We provide an explanation of the LMT algorithm in the Appendix.

3.5. Implementation Details

Image Feature Extractor. For fair comparisons, we use the same image size and pretrained feature extractor as the previous state-of-the-art in each setting. For all datasets except CUB, we use the ResNet-101 [21] pretrained on ImageNet [14] as the feature extractor (for CUB, we use the same as [25]). Since the output dimension of ResNet-101 is 2048, we set our embedding size d as 2048. Following [8, 7], images are resized to 640×640 and randomly cropped to 576×576 with random horizontal flips during training. Testing images are center cropped instead. The output of ResNet-101 is an $18\times18\times d$ tensor, so there are a total of 324 feature embedding vectors, $\boldsymbol{z}_i \in \mathbb{R}^d$.

Transformer Encoder. In order to allow a particular embedding to attend to multiple other embeddings (or multiple groups), C-Tran uses 4 attention heads [49]. We use a L=3 layer Transformer with a residual layer [21] around each embedding update and layer norm [1].

Optimization. Our model, including the pretrained feature extractor, is trained end-to-end. We use Adam [24] for the optimizer with betas=$(0.9, 0.999)$ and weight decay=0. We train the models with a batch size of 16 and a learning rate of 10^{-5}. We use dropout with $p = 0.1$ for regularization.

4. Experimental Setup and Results

In the following subsections, we explain the datasets, baselines, and results for the three multi-label classification inference settings.

	All							Top 3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
CNN-RNN [50]	61.2	-	-	-	-	-	-	66.0	55.6	60.4	69.2	66.4	67.8
RNN-Attention [52]	-	-	-	-	-	-	-	79.1	58.7	67.4	84.0	63.0	72.0
Order-Free RNN [6]	-	-	-	-	-	-	-	79.1	58.7	67.4	84.0	63.0	72.0
ML-ZSL [31]	-	-	-	-	-	-	-	74.1	64.5	69.0	-	-	-
SRN [58]	77.1	81.6	65.4	71.2	82.7	69.9	75.8	85.2	58.8	67.4	87.4	62.5	72.9
ResNet101 [21]	77.3	80.2	66.7	72.8	83.9	70.8	76.8	84.1	59.4	69.7	89.1	62.8	73.6
Multi-Evidence [19]	-	80.4	70.2	74.9	85.2	72.5	78.4	84.5	62.2	70.6	89.1	64.3	74.7
ML-GCN [10]	83.0	85.1	72.0	78.0	85.8	75.4	80.3	89.2	64.1	74.6	90.5	66.5	76.7
SSGRL [8]	83.8	**89.9**	68.5	76.8	**91.3**	70.8	79.7	**91.9**	62.5	72.7	**93.8**	64.1	76.2
KGGR [7]	84.3	85.6	72.7	78.6	87.1	75.6	80.9	89.4	64.6	75.0	91.3	66.6	77.0
C-Tran	**85.1**	86.3	**74.3**	**79.9**	87.7	**76.5**	**81.7**	90.1	**65.7**	**76.0**	92.1	**71.4**	**77.6**

Table 1. Results of *regular inference* on COCO-80 dataset. The threshold is set to 0.5 to compute precision, recall and F1 scores (%). Our method consistently outperforms previous methods across multiple metrics under the settings of all and top-3 predicted labels. Best results are shown in bold. "-" denotes that the metric was not reported.

	All							Top 3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
ResNet101[21]	30.9	39.1	25.6	31.0	61.4	35.9	45.4	39.2	11.7	18.0	75.1	16.3	26.8
ML-GCN [10]	32.6	42.8	20.2	27.5	66.9	31.5	42.8	39.4	10.6	16.8	77.1	16.4	27.1
SSGRL [8]	36.6	-	-	-	-	-	-	-	-	-	-	-	-
KGGR [7]	37.4	47.4	24.7	32.5	**66.9**	36.5	47.2	48.7	12.1	19.4	78.6	17.1	28.1
C-Tran	**38.4**	**49.8**	**27.2**	**35.2**	**66.9**	**39.2**	**49.5**	**51.1**	**12.5**	**20.1**	**80.2**	**17.5**	**28.7**

Table 2. Results of *regular inference* on VG-500 dataset. All metrics and setups are the same as Table 1. Our method achieves notable improvement over previous methods.

4.1. Regular Inference

Datasets. We use two large-scale regular multi-label classification datasets: COCO-80 and VG-500. COCO [34], is a commonly used large scale dataset for multi-label classification, segmentation, and captioning. It contains $122,218$ images containing common objects in their natural context. The standard multi-label formulation for COCO, which we call COCO-80, includes 80 object class annotations for each image. We use $82,081$ images as training data and evaluate all methods on a test set consisting of $40,137$ images. The Visual Genome dataset [27], contains $108,077$ images with object annotations covering thousands of categories. Since the label distribution is very sparse, we only consider the 500 most frequent objects and use the VG-500 subset introduced in [7]. VG-500 consists of $98,249$ training images and $10,000$ test images.

Baselines and Metrics. For COCO-80, we compare to ten well known multi-label classification methods. For VG-500 we compare to four previous methods that used this dataset. Referencing previous works [10, 8, 7], we employ several metrics to evaluate the proposed method and existing methods. Concretely, we report the average per-class precision (CP), recall (CR), F1 (CF1) and the average overall precision (OP), recall (OR), F1 (OF1), under the setting that a predicted label is positive if the output probability is greater than 0.5. We also report the mean average precision (mAP). A detailed explanation of the metrics are shown in the Appendix. For fair comparisons to previous works [19, 58], we also consider the setting where we evaluate the Top-3 predicted labels following. In general, **mAP**, **OF1**, and **CF1** are the most important metrics [10].

Results. C-Tran achieves state-of-the-art performance almost across all metrics on both datasets, as shown in Table 1 and Table 2. Considering that COCO-80 and VG-500 are two widely studied multi-label datasets, absolute mAP increases of 0.8 and 1.0, respectively, can be considered notable improvements. Importantly, we do not use any predefined feature and label relationship information (e.g. pretrained word embeddings). This signals that our method can effectively learn the relationships.

4.2. Inference with Partial Labels

Datasets. We use four datasets to validate our approach in the partial label setting. In all four datasets, we simulate four amounts of partial labels during inference. More specifically, for each testing image, we select ϵ percent of labels as known. ϵ is set to 0% / 25% / 50% / 75% in our experiments. ϵ=0% denotes no known labels, and is equivalent to the regular inference setting.

	COCO-80				VG-500				NEWS-500				COCO-1000			
Partial Labels Known (ϵ)	0%	25%	50%	75%	0%	25%	50%	75%	0%	25%	50%	75%	0%	25%	50%	75%
Feedbackprop [51]	80.1	80.6	80.8	80.9	29.6	30.1	30.8	31.6	14.7	21.1	23.7	25.9	29.2	30.1	31.5	33.0
C-Tran	**85.1**	**85.2**	**85.6**	**86.0**	**38.4**	**39.3**	**40.4**	**41.5**	**18.1**	**29.7**	**35.5**	**39.4**	**34.3**	**35.9**	**37.4**	**39.1**

Table 3. Results of *inference with partial labels* on four multi-label image classification datasets. Mean average precision score (%) is reported. Across four simulated settings where different amounts of partial labels are available (ϵ), our method significantly outperforms the competing method. With more partial labels available, we achieve larger improvement.

Extra Label Groups Known (ϵ)	0%	36%	54%	71%
Standard [25]	82.7	82.7	82.7	82.7
Multi-task [25]	**83.8**	83.8	83.8	83.8
ConceptBottleneck [25]	80.1	87.0	93.0	97.5
C-Tran	**83.8**	**90.0**	**97.0**	**98.0**

Table 4. Results of *inference with extra labels* on CUB-312 dataset. We report the accuracy score (%) for the 200 multi-class target labels. We achieve similar or greater accuracy than the baselines across all amounts of known extra label groups.

In addition to COCO-80 and VG-500, we benchmark our method on two more multi-label image classification datasets. Wang et al. [51] derived the top 1000 frequent words from the accompanying captions of COCO images to use as target labels, which we call COCO-1000. There are $82,081$ images for training, and $5,000$ images for validation and testing, respectively. We expect that COCO-1000 provides more and stronger dependencies compared to COCO-80. We also use the NEWS-500 dataset [51], which was collected from the BBC News. Similar to COCO-1000, the target label set consists of 500 most frequent nouns derived from image captions. There are $151,873$ images for training, $10,304$ for validation and $10,451$ for testing.

Baselines and Metrics. Feedback-prop [51] is an inference method introduced for partial label inference that make use of arbitrary amount of known labels. This method back-propagates the loss on the known labels to update the intermediate image representations during inference. We use the LF method on ResNet-101 Convolutional Layer 13 as in [51]. We compute the mean average precision (mAP) score of predictions on unknown labels.

Results. As shown in Table 3, C-Tran outperforms Feedbackprop, in all ϵ percentages of partially known labels on all datasets. In addition, as the percentage of partial labels increases, the improvement of C-Tran over Feedbackprop also increases. These results demonstrate that our method can effectively leverage known labels and is very flexible with the amount of known labels. Feedbackprop updates image features which implicitly encode some notion of label correlation. C-Tran, instead, explicitly models the correlations between labels and features, leading to improved results especially when partial labels are known. On the other hand, Feedback-prop requires careful hyperparameter tuning on a separate validation set and needs time-consuming iterative feature updates. Our method does not require any hyperparameter tuning and just needs a standard one-pass inference. Further comparisons and qualitative examples are included in the Appendix.

4.3. Inference with Extra Labels

Datasets. For the extra label setting, we use the Caltech-UCSD Birds-200-2011 (CUB) dataset [53]. It contains 9,430 training samples and 2,358 testing samples. We conduct a multi-classification task with 200 bird species on this dataset. Multi-class classification is a specific instantiation of multi-label classification, where the target classes are mutually exclusive. In other words, each image has only one correct label. We use the processed CUB dataset from Koh et al. [25] where they include 112 extra labels related to bird species. We call this dataset CUB-312. They further cluster extra labels into 28 groups and use varying amounts of known groups at inference time. To make a fair comparison, we consider four different amounts of extra label groups for inference: 0 group (0%), 10 groups (36%), 15 groups (54%), and 20 groups (71%).

Baselines and Metrics. Concept Bottleneck Models [25] incorporate the extra labels as intermediate labels ("concepts" in the original paper). These models use a bottleneck layer to first predict the extra labels, and then use those predictions to predict bird species. I.e., if we let $\mathbf{y}^e$ be the extra information labels, [25] predicts the target class labels $\mathbf{y}^t$ using the following computation graph: $\mathbf{x} \rightarrow \mathbf{y}^e \rightarrow \mathbf{y}^t$. As in [25], we also consider two baselines: A standard multi-layer perceptron, and a multi-task learning model that predicts the target and concept labels jointly. For fair comparison, we use the same feature extraction method for all experiments, Inception-v3 [44]. We evaluate target predictions using multi-class accuracy scores.

Results. Table 4 shows that C-Tran achieves an improved accuracy over Concept Bottleneck models on the CUB-312 task when using any amount of extra label groups. Notably, the multi-task learning model produces the best performing results when ϵ=0. However, it is not able to incorporate known extra labels (i.e., $\epsilon > 0$). C-Tran instead, consistently achieves the best performance. Additionally, we can test interventions, or counterfactuals, using C-Tran. For example, "grey beak" is one of the extra labels, and we can set the

state embedding of "grey beak" to be positive or negative and observe the change in bird class predictions. We provide samples of extra label interventions in the Appendix.

4.4. Ablation and Model Analysis

We conduct ablation studies to analyze the contributions of each C-Tran component. We examine two settings: regular inference (equivalent to 0% known partial labels) and 50% known partial label inference. We evaluate on four datasets: COCO-80, VG-500, NEWS-500, and COCO-1000. First, we remove the image features $\mathbf{Z}$ and predict unknown labels given only known labels. This experiment, C-Tran (no image), tells us how much information model can learn just from labels. Table 5 shows that we get relatively high mean average precision scores on some datasets (NEWS-500 and COCO-1000). This indicates that even without image features, C-Tran is able to effectively learn rich dependencies from label annotations .

Second, we remove the label mask training procedure to test the effectiveness of this technique. More specifically, we remove all label state embeddings, $\mathbf{S}$; thus all labels are unknown during training. Table 5 shows that for both settings, regular (0%) and 50% partial labels known, the performance drops without label mask training. This signifies two critical findings of label mask training: (1) it helps with dependency learning as we see improvement when no partial labels are available during inference. This is particularly true for datasets that have strong label co-occurrences, such as NEWS-500 and COCO-1000. (2) given partial labels, it can significantly improve prediction accuracy. We provide a t-SNE plot [36] of the label embeddings learned with and without label mask training. As shown in Figure 4, embeddings learned with label mask training exhibit a more meaningful semantic topology; i.e. objects belonging to the same group are clustered together.

We also analyze the importance of the number of Transformer layers, L, for regular inference in COCO-80. Mean average precision scores for 2, 3, and 4 layers were 85.0, 85.1, and 84.3, respectively. This indicates : (1) our method is fairly robust to the number of Transformer layers, (2) multi-label classification does not seem to require a very large number of layers as in some NLP tasks [4]. While we show C-Tran is a powerful method in many multi-label classification settings, we recognize that Transformer layers are memory-intensive for a large number of inputs. This limits the number of possible labels ℓ in our model. Using four NVIDIA Titan X GPUs, the upper bound of ℓ is around 2000 labels. However, it is possible to increase the number of labels. We currently use the ResNet-101 output channel size ($d = 2048$) for our Transformer hidden layer size. This can be linearly mapped to a smaller number. Additionally, we could apply one of the Transformer variations that have been proposed to model very large input sizes [11, 43].

Partial Labels Known (ϵ)	COCO-80		VG-500		NEWS-500		COCO-1000	
	0%	50%	0%	50%	0%	50%	0%	50%
C-Tran (no image)	3.60	21.7	2.70	24.6	6.50	33.3	1.50	27.8
C-Tran (no LMT)	84.8	85.0	38.3	38.8	16.9	17.1	33.1	34.0
C-Tran	**85.1**	**85.6**	**38.4**	**40.4**	**18.1**	**35.5**	**34.3**	**37.4**

Table 5. C-Tran component ablation results. Mean average precision score (%) is reported. Our proposed Label Mask Training technique (LMT) improves the performance, especially when partial labels are available.

5. Related Work

Our work relates to the prior literature in image categorization. While a lot of work focuses on single-label classification [14], there is also an ample body of work on both multi-label prediction and exploiting label dependencies [13, 39, 35, 22]. There is also an increasing recognition of the importance of being able to handle partial labels both during training and inference. We review some of this work in this section as follows:

Multi-label Image Classification. Multi-label classification (MLC) is gaining popularity due to its relevance in real world applications. Recently, Stock et al [42] showed that the remaining error in ImageNet is not due to the feature extraction, but rather that ImageNet is annotated with single labels even when some images depict more than one object.

Recent literature addressing multi-label classification roughly fall into four groups. *(1) Conditional Prediction:* The first type, autoregressive models [12, 41, 50, 37] estimate the true joint probability of output labels given the input by using the chain rule, predicting one label at a time. *(2) Shared Embedding Space:* The second group learns to project input features and output labels into a shared latent embedding space [57, 3]. *(3) Structured Output:* The third kind describes label dependencies using structured output inference formulation [29, 47, 2, 20, 32, 33, 56, 38]. *(4) Label Graph Formulation:* Several recent studies [10, 30, 8, 7] used graph neural networks to model label dependency an obtained state-of-the-art results. All methods relied on knowledge-based graphs being built from label co-occurrence statistics. Our proposed model is most similar to *(4)*, but it does not need extra knowledge to build a graph and can automatically learn label dependencies.

Inference with Partial Labels, Wang et al. proposed feedback propagation to handle any set of partial labels at test time [51]. The idea is to optimize intermediate image representations according to *known* labels and then predict *unknown* labels based on updated representations. Yang et al [55] use this type of approach to pivot information across captions in different languages. Huang et al [23] use feedback consistency to improve adversarial robustness. However, these methods require many iterations at infer-

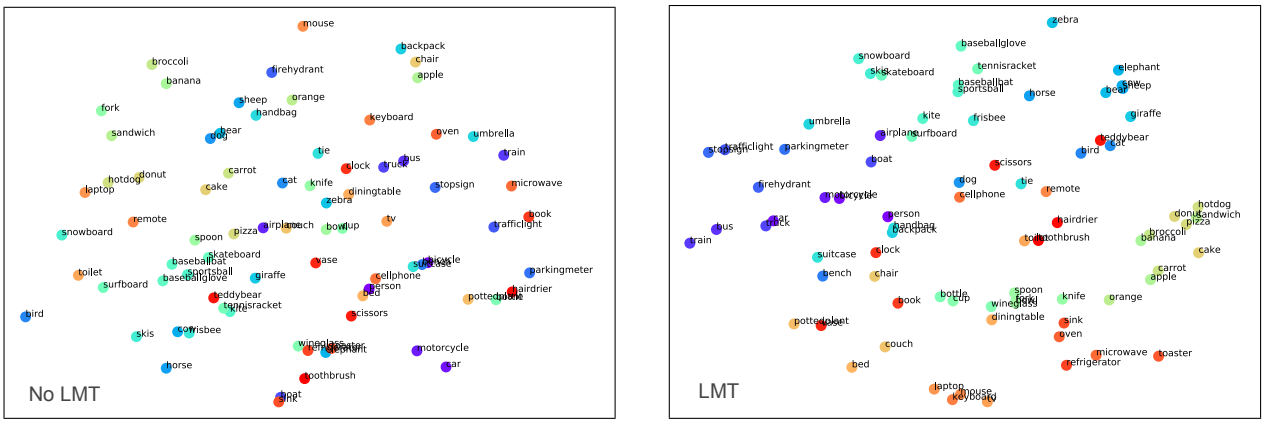

Figure 4. Comparison of the learned label embeddings for COCO-80 using t-SNE. The left figure shows the embedding projections without using label mask training (LMT), and the right shows with LMT. Labels are colored using the COCO object categorization. We can see that using label mask training produces much semantically stronger label representations.

ence time, and particularly the model in [51] is not exposed to partial evidence during training, which limits potential gains. Several methods [26, 22] utilize partial labels using a fixed set of labels. In realistic settings, however there could be an arbitrary set of known labels available during inference. If there are ℓ total labels, then the number of known labels, n=$|\mathbf{y}_k|$ ranges from 0 to ℓ-1. The number of possible known label sets is then $\binom{\ell}{n}$. C-Tran, integrates a novel representation indicating each label state as *positive*, *negative* or *unknown*. This representation enables us to leverage partial signals during training, and make our model compatible with any known label set during inference. Notably, C-Tran can exploit arbitrary amounts of partial evidence during both training and inference.

Many prominent works also tackle the problem of *training* models with partial label annotations [54, 17, 28]. While this might seem similar to our setting, the key distinction is that these methods assume that images have incomplete or partial labels only during training. However, partial label training methods make no assumptions about the inference settings and thus cannot be easily extended to the scenario where partial labels are available at test time. We consider our line of work complementary to these efforts as these are not mutually exclusive.

Inference with Extra Labels, Koh et al [25] introduces Concept Bottleneck Models which incorporate intermediate concept labels as a bottleneck layer for the target label classification. Similar to [26], this model assumes that the concept labels are a fixed set. Our model goes further by relaxing the need for a fixed set and uses state embeddings instead of a concept bottleneck layer to represent each concept as *known* (positive or negative) or *unknown*. This representation enables C-Tran to leverage partial labels (concepts) during training, and make our model compatible with any known labels (concepts) during inference.

Transformers for Computer Vision Several recent works have used Transformers in computer vision applications [5, 9, 40, 16, 45]. Some of these models replace a significant part of the visual recognition pipeline with a transformer [16, 40, 5] while others use a transformer on top of features computed by a convolutional neural network [9, 45]. Our model is architecturally similar to the latter, with a focus on using arbitrary amounts of output labels as *input* to the model.

6. Conclusion

We propose C-Tran, a novel and flexible deep learning model for multi-label image classification. Our approach is easy to implement and can effectively leverage an arbitrary set of partial or extra labels during inference. C-Tran learns sample-adaptive interactions through attention and discovers how labels attend to different parts of an image. We show the effectiveness of our approach in regular multi-label classification and multi-label classification with partially observed or extra labels. C-Tran outperforms state-of-the-art methods in a wide range of scenarios. We further provide a quantitative and qualitative analysis showing that C-Tran obtains gains by explicitly modeling interactions between target labels and between image features and target labels. Further work could extend C-Tran for hierarchical scene categorization, and explore training strategies to make C-Tran generalize to settings where some labels have never been observed during training.

Acknowledgements This work was partly supported by the National Science Foundation under NSF CAREER award No. 1453580 to Y.Q. and a Leidos gift award to V.O. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] David Belanger, Bishan Yang, and Andrew McCallum. End-to-end learning for structured prediction energy networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 429–439. JMLR. org, 2017.

[3] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in neural information processing systems*, pages 730–738, 2015.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[6] Shang-Fu Chen, Yi-Chen Chen, Chih-Kuan Yeh, and Yu-Chiang Wang. Order-free rnn with visual attention for multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[7] Tianshui Chen, Liang Lin, Xiaolu Hui, Riquan Chen, and Hefeng Wu. Knowledge-guided multi-label few-shot learning for general image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[8] Tianshui Chen, Muxin Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. Learning semantic-specific graph representation for multi-label image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 522–531, 2019.

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020.

[10] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-Label Image Recognition with Graph Convolutional Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics.

[12] Krzysztof Dembczynski, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *ICML*, 2010.

[13] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *European conference on computer vision*, pages 48–64. Springer, 2014.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[17] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 647–657, 2019.

[18] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, pages 681–687, 2002.

[19] Weifeng Ge, Sibei Yang, and Yizhou Yu. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1277–1286, 2018.

[20] Yuhong Guo and Suicheng Gu. Multi-label classification using conditional dependency networks. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1300, 2011.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Hexiang Hu, Guang-Tong Zhou, Zhiwei Deng, Zicheng Liao, and Greg Mori. Learning structured inference neural networks with label relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2960–2968, 2016.

[23] Yujia Huang, James Gornet, Sihui Dai, Zhiding Yu, Tan Nguyen, Doris Tsao, and Anima Anandkumar. Neural networks with recurrent generative feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors,

Advances in Neural Information Processing Systems, volume 33, pages 535–545. Curran Associates, Inc., 2020.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daum III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020.

[26] Michal Koperski, Tomasz Konopczynski, Rafal Nowak, Piotr Semberecki, and Tomasz Trzcinski. Plugin networks for inference under partial evidence. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2883–2891, 2020.

[27] 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 Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016.

[28] Kaustav Kundu and Joseph Tighe. Exploiting weakly supervised visual patterns to learn from partial annotations. *Advances in Neural Information Processing Systems*, 33, 2020.

[29] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ., 2001.

[30] Jack Lanchantin, Arshdeep Sekhon, and Yanjun Qi. Neural message passing for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 138–163. Springer, 2019.

[31] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Multi-label zero-shot learning with structured knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1576–1585, 2018.

[32] Qiang Li, Maoying Qiao, Wei Bian, and Dacheng Tao. Conditional graphical lasso for multi-label image classification. In *CVPR*, pages 2977–2986, 06 2016.

[33] Xin Li, Feipeng Zhao, and Yuhong Guo. Multi-label image classification with a probabilistic label enhancement model. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pages 430–439, Arlington, Virginia, USA, 2014. AUAI Press.

[34] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[35] Yong Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6985–6994, 2018.

[36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[37] Jinseok Nam, Eneldo Loza Mencía, Hyunwoo J Kim, and Johannes Fürnkranz. Maximizing subset accuracy with recurrent neural networks in multi-label classification. In *Advances in Neural Information Processing Systems*, pages 5419–5429, 2017.

[38] Tejaswi Nimmagadda and Anima Anandkumar. Multi-object classification and unsupervised scene understanding using deep learning features and latent tree probabilistic models. *arXiv preprint arXiv:1505.00308*, 2015.

[39] Vicente Ordonez, Wei Liu, Jia Deng, Yejin Choi, Alexander C Berg, and Tamara L Berg. Predicting entry-level categories. *International Journal of Computer Vision*, 115(1):29–43, 2015.

[40] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

[41] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, 2009.

[42] Pierre Stock and Moustapha Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 498–512, 2018.

[43] Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 331–335, Florence, Italy, July 2019. Association for Computational Linguistics.

[44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[45] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. Instance-level image retrieval using reranking transformers. *arXiv preprint arXiv:2103.12236*, 2021.

[46] Wilson L Taylor. cloze procedure: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

[47] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6(Sep):1453–1484, 2005.

[48] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3), 2006.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[50] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2285–2294, 2016.

[51] Tianlu Wang, Kota Yamaguchi, and Vicente Ordonez. Feedback-prop: Convolutional neural network inference under partial evidence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[52] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE international conference on computer vision*, pages 464–472, 2017.

[53] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[54] Ming-Kun Xie and Sheng-Jun Huang. Partial multi-label learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[55] Ziyan Yang, Leticia Pinto-Alva, Franck Dernoncourt, and Vicente Ordonez. Using visual feature space as a pivot across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3673–3678, Online, Nov. 2020. Association for Computational Linguistics.

[56] Mark Yatskar, Vicente Ordonez, Luke Zettlemoyer, and Ali Farhadi. Commonly uncommon: Semantic sparsity in situation recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7196–7205, 2017.

[57] Chih-Kuan Yeh, Wei-Chieh Wu, Wei-Jen Ko, and Yu-Chiang Frank Wang. Learning deep latent space for multi-label classification. In *AAAI*, pages 2838–2844, 2017.

[58] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2027–2036, 2017.