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 uncertainty of labels 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 [15, 46, 48, 50, 6, 32, 9]. This task goes beyond the more well studied single-label multi-class classification problem 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 reflects 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.

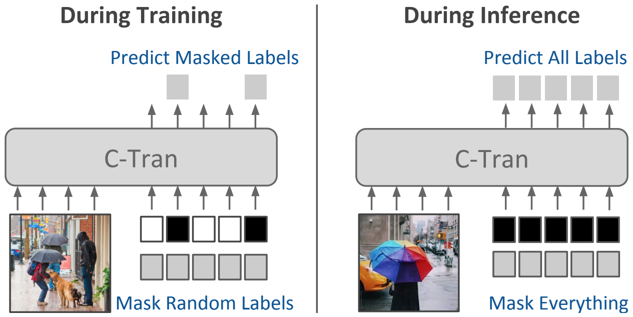

Figure 1. We propose a transformer-based model for multi-label image classification that exploits dependencies among a target set of labels using an encoder transformer. During training, the model learns to reconstruct a partial set of labels given randomly masked input label embeddings and image features. 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 [47]. Transformers have demonstrated a remarkable capability of being able to exploit complex and rich dependencies among sets of inputs using multiple layers of multi-headed self-attention operations. 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. Unlike the Transformer encoders used for language modeling [13], C-Tran uses a label mask training objective that allows us to represent the state of the labels as *positive*, *negative*, or *unknown*. 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 the introduced regular multi-label classification task, we also claim that C-Tran is a more general model for reasoning under

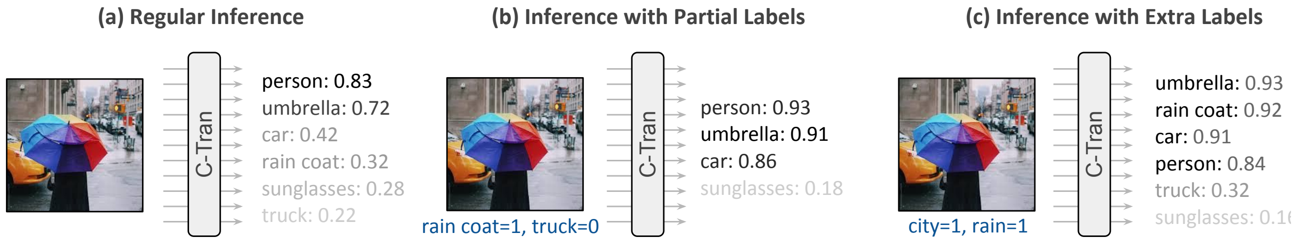

Figure 2. Different inference settings for general multi-label image classification: (a) Standard multi-label classification takes only image features as input. All labels are unknown $\mathbf{y}_u$.; (b) Classification under partial labels takes as input image features as well as a subset of the target 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 $\mathbf{y}_u^t$.

prior label observations. Because our approach explicitly models the uncertainty of the labels 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 in the picture. 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. Our general approach to multi-label image classification with Transformers is able to naturally handle all these scenarios under a unified framework. We compare our results with a competing method relying on iterative inference [49], and against sensitive baselines, demonstrating superior results under variable amounts of partial or extra labels.

The benefits of our proposed framework 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

In this section, we formally explain the three different multi-label image classification inference settings that we use to demonstrate the utility of our approach.

Regular Multi-label Classification. In regular multi-label image classification, 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 accompanied by some labeled text such as captions 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 properly annotated 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, 14, 27].

Inference with Extra Labels. Similar to partially labeled images, there are many cases where we observe extra labels that describe the 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 labels we want to predict, it can still alter our perception about what might be in the image. In this setting, we append any potential extra labels, denoted $\mathbf{y}^e$, to the target label set $\mathbf{y}^t$. If there are ℓ^t target labels, and ℓ^e potential extra labels, we now have a set of $\ell^t + \ell^e$ total labels that we train the model to

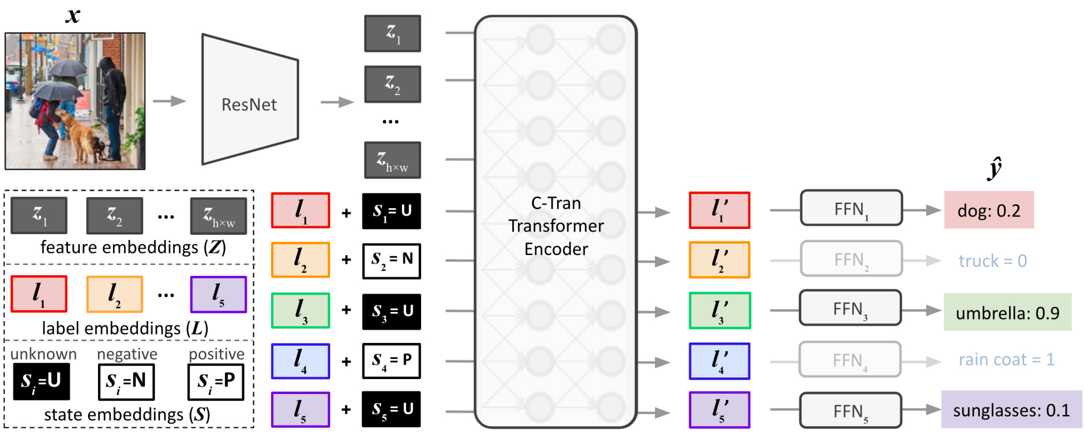

Figure 3. C-Tran architecture and illustration of label mask training for general multi-label image classification. In this training image, the labels *person*, *umbrella*, and *sunglasses* were randomly masked out and used as the unknown labels, $\mathbf{y}_u$. The labels *rain coat* and *truck* are 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 $\hat{\mathbf{y}}_u$.

predict. $\mathbf{y}$ 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: $\hat{\mathbf{y}}_u^t = f(\mathbf{x}, \mathbf{y}_k^e)$. Again, we assume that all training images are fully annotated with their correct target and extra labels.

3. Method: C-Tran

Considering the three inference settings described, we propose Classification Transformers (C-Tran), a general multi-label classification framework that works in all three. 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 the 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 way 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, $\boldsymbol{s}_i \in \mathbb{R}^d$:

$$\tilde{\boldsymbol{l}}_i = \boldsymbol{l}_i + \boldsymbol{s}_i, \quad (1)$$

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 set this label to positive (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 the complex interactions between the image feature and embeddings, we develop our model based on a Transformer [47]. Transformers have proven to be a powerful mechanism for capturing rich dependency information between variables. Our formulation lets us to easily input the image feature 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 the α_{ij} value for all i and j pairs, we update each $\boldsymbol{h}_i$ to $\boldsymbol{h}'_i$ using a weighted sum of all embeddings followed by a nonlinear ReLU layer:

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

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

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

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:

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

3.4. Label Mask Training (LMT)

State embeddings (Eq. 1) lets 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 [44] and the BERT "masked language model" [13] which learn semantic information 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. This allows the C-Tran to be used in any inference setting where there may be 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 [13, 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.

Essentially, our label mask training pipeline tries to minimize the following loss approximately:

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

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: $\mathbf{y}_k$.

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 [20] pretrained on ImageNet [12] as the feature extractor (for CUB, we use the same as [23]). Since the output dimension of ResNet-101 is 2048, we set our embedding size d as 2048. Following [8, 7], training are images resized to 640 × 640 and randomly cropped to 576 × 576 with random horizontal flips. Testing images are center cropped instead. The output of the ResNet-101 model 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 [47]. We use a L=3 layer Transformer with a residual layer [20] around each embedding update and layer norm [1].

Optimization. For training, Adam [22] is used as 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 [17] of $p = 0.1$ 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.

4.1. Regular Inference

Datasets. We use two large-scale regular multi-label classification datasets: COCO-80 and VG-500. COCO [35], 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 [25], 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 [9, 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 [16, 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 [9].

Results. C-Tran achieves state-of-the-art performance across almost 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 lean 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.

In addition to COCO-80 and VG-500, we benchmark our method on two more multi-label image classification datasets. Wang et al. [49] 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 [49], 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 [49] is an inference method introduced for partial label inference that make use of arbitrary amount of known labels. This method backpropagates 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 from [49]. 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 of 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 hyerparameter tuning and just needs a standard one-pass inference. We include qualitative examples in Appendix, demonstrating that effectiveness of our method.

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

	All							Top 3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
CNN-RNN [48]	61.2	-	-	-	-	-	-	66.0	55.6	60.4	69.2	66.4	67.8
RNN-Attention [50]	-	-	-	-	-	-	-	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 [32]	-	-	-	-	-	-	-	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 [20]	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 [16]	-	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 [9]	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[20]	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 [9]	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.

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. [23] 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 [23] incorporate the extra labels as intermediate labels ("concepts" in the original paper). They construct 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, [23] predicts the target class labels $\mathbf{y}^t$ using the following computation graph: $\mathbf{x} \rightarrow \mathbf{y}^e \rightarrow \mathbf{y}^t$. We also consider two baselines from [23]. The first is standard multi-layer perception model that does not use a bottleneck layer. The second is 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 methods, Inception-v3 [43]. Since the target task is multi-class, we evaluate the target predictions using 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., ϵ >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 intervention in the Appendix.

4.4. Ablation and Model Analysis

In this section, 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

	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 [49]	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 [23]	82.7	82.7	82.7	82.7
Multi-task [23]	**83.8**	83.8	83.8	83.8
ConceptBottleneck [23]	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.

Partial Labels	COCO-80		VG-500		NEWS-500		COCO-1000	
Known (ϵ)	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.

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, **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 or without label mask training. As revealed in Figure 4, embeddings learned with label mask training show more meaningful semantic topology; objects belonging to the same group are clustered together.

We also analyze the importance of the number of Transformer layers, L, in the regular inference setting for 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 likely doesn't require a very large number of layers like some other NLP tasks, which use 96 layers [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 [10, 42].

5. Related Work

Multi-label Image Classification. Multi-label classification (MLC) is gaining popularity due to its relevance in real world applications. Recently, [40] 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 [11, 39, 48, 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 [28, 45, 2, 18, 33, 34, 38]. *(4) Label Graph Formulation:* Several recent studies [9, 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 the label dependency.

Inference with Partial Labels Wang et al. proposed feedbackprop, a new inference strategy to handle any set of partial labels at test time [49]. The core idea is to optimize

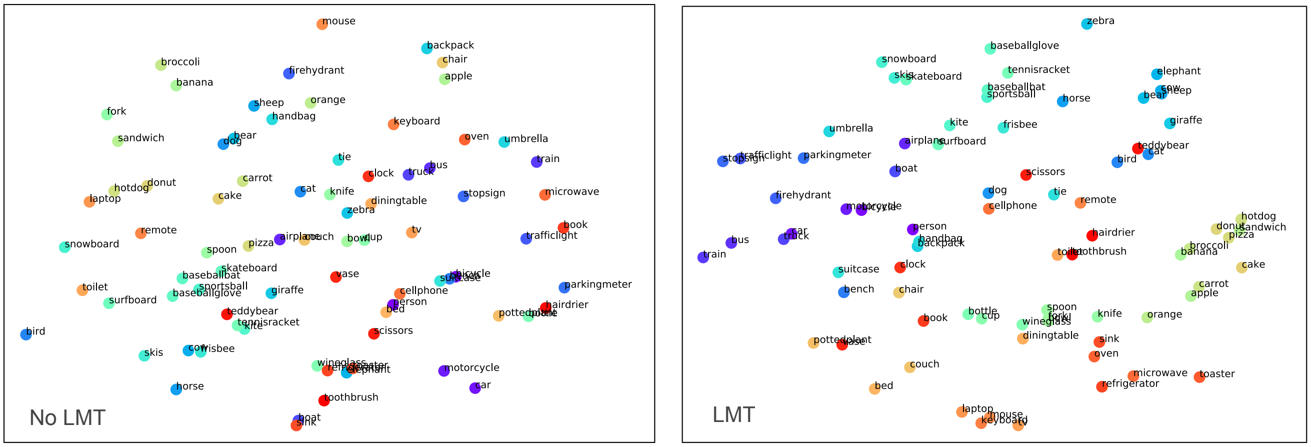

Figure 4. Comparison of the learned label embeddings for COCO-80 using t-SNE. The left figure shows the embedding projections without using label mask training (LMT), and the right shows with LMT. Labels are colored using the COCO object categorization. We can see that using label mask training produces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

intermediate image representations according to *known* labels and then predict the *unknown* labels based on the updated representations. However, this requires many iterations at inference time, resulting in a significantly slower classifier. Additionally, the model is never exposed to partial evidence during training, which limits the potential improvement. Several methods [24, 21] utilize partial labels using a fixed set of labels. However, these cannot generalize to arbitrary sets of known labels. In more realistic inference settings, there may be any subset 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, instead integrates a novel representation indicating each label state as *positive*, *negative* or *unknown*. This representation enables us to leverage partial signals into the model training, and make our model compatible with any known label set during inference. Notably, C-Tran is the first learning method that can exploit arbitrary amounts of partial evidence during both training and inference.

Many works tackle the problem of partial label multi-label classification *training* [54, 14, 27]. While this sounds similar to our setting, there are several key distinctions. First, these methods focus on the case of "partial annotations", which means that they assume not all labels are annotated correctly during training. We assume that all labels are correctly annotated during training. Second, partial label training methods cannot be easily extended to the partial label inference setting. In other words, these methods can just take images as input and fail to incorporate extra useful information (partial/extra labels) during inference.

Inference with Extra Labels [23] introduces Concept Bottleneck Models which incorporates intermediate concept labels as a bottleneck layer for the target label classification. Similar to [24], this model assumes that the concept labels are a fixed set. While interpretability is an advantage, bottleneck models [29, 26] rely on the assumption that the manually curated concepts are sufficient features for target class prediction, contradicting the feature learning approach of deep learning. C-Tran, 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. Importantly, we do not have any assumptions of the size of labels (concepts) to be known during inference.

6. Conclusion

This paper proposes a novel deep learning method, called C-Tran, for a wide variety of "multi-label image classification" applications. Our approach is easy to implement, requires no extra resources, and can effectively leverage any amount of partial or extra labels during inference. C-Tran learns sample-adaptive interactions through attention and can discover how the labels attend to different parts of an input image. We showcase the effectiveness of our approach in regular multi-label classification settings and multi-label classification with partially observed or extra labels. C-Tran outperforms all state-of-the-art methods in all scenarios. We further provide a quantitative and qualitative analysis showing that C-Tran boosts the performance by explicitly modeling the interactions between target labels and between image features and target labels. As the next steps, we plan to extend C-Tran to hierarchical scene categorization applications. We also plan to explore the design of better training strategies to make C-Tran generalize to settings where some labels have never been observed in 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. 4

[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. 7

[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. 7

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 4, 7

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 11

[6] Shang-Fu Chen, Yi-Chen Chen, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Order-free rnn with visual attention for multi-label classification. *arXiv preprint arXiv:1707.05495*, 2017. 1, 6

[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. 4, 5, 6, 7

[8] Tianshui Chen, Muxin Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. Learning semantic-specific graph representation for multi-label image recognition. *arXiv preprint arXiv:1908.07325*, 2019. 4, 5, 6, 7

[9] 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. 1, 5, 6, 7

[10] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 7

[11] Krzysztof Dembczynski, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In ., 2010. 7

[12] 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. 4

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 4, 11

[14] 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. 2, 8

[15] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, pages 681–687, 2002. 1

[16] 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. 5, 6

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 5

[18] 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. 7, 12

[19] Kazuyuki Hara, Daisuke Saitoh, and Hayaru Shouno. Analysis of dropout learning regarded as ensemble learning. In *International Conference on Artificial Neural Networks*. Springer, 2016. 12

[20] 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. 4, 6

[21] 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. 8

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[23] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. *arXiv preprint arXiv:2007.04612*, 2020. 4, 6, 7, 8

[24] 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. 8

[25] 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. 5

[26] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th international conference on computer vision*, pages 365–372. IEEE. 8

[27] Kaustav Kundu and Joseph Tighe. Exploiting weakly supervised visual patterns to learn from partial annotations. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 8

[28] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ., 2001. 7

[29] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009. 8

[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. 7

[31] Jack Lanchantin, Arshdeep Sekhon, and Yanjun Qi. Neural message passing for multi-label classification. *ECML*, abs/1904.08049, 2019. 12

[32] 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. 1, 6

[33] Qiang Li, Maoying Qiao, Wei Bian, and Dacheng Tao. Conditional graphical lasso for multi-label image classification. In *CVPR*, pages 2977–2986, 06 2016. 7

[34] 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. 7

[35] 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. 5

[36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 7

[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. 7

[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. 7

[39] 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. 7

[40] 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. 7

[41] Hongyu Su and Juho Rousu. Multilabel classification through random graph ensembles. In *Asian Conference on Machine Learning*, pages 404–418, 2013. 12

[42] Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*, 2019. 7

[43] 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. 6

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

[45] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6(Sep):1453–1484, 2005. 7

[46] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3), 2006. 1

[47] 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. 1, 3, 4, 11

[48] 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. 1, 6, 7

[49] 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. 2, 5, 7

[50] 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. 1, 6

[51] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 11

[52] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. Hcp: A flexible cnn framework for multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1901–1907, Sep. 2016. 11

[53] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 5

[54] Ming-Kun Xie and Sheng-Jun Huang. Partial multi-label learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2, 8

[55] Xiangyang Xue, Wei Zhang, Jie Zhang, Bin Wu, Jianping Fan, and Yao Lu. Correlative multi-label multi-instance image annotation. In *Proceedings of the 2011 International*

Conference on Computer Vision, ICCV '11, pages 651–658, USA, 2011. IEEE Computer Society. 12

[56] Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. Exploit bounding box annotations for multi-label object recognition. In Lourdes Agapito, Tamara Berg, Jana Kosecka, and Lihi Zelnik-Manor, editors, *Proceedings - 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 280–288, United States of America, 2016. IEEE, Institute of Electrical and Electronics Engineers. 11

[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. 7

[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. 5, 6, 11

A. Appendix

A.1. Qualitative Examples

Inference with Partial Labels In Figure 5, we show qualitative results on COCO-80 demonstrating the use of partial labels. In these examples, we first show the predictions for ResNet-101, as well as C-Tran without using partial labels. The last column shows the C-Tran predictions when using $\epsilon = 25\%$ partial labels (which is 21 labels for COCO-80) as observed, or known prior to inference. For many examples, certain labels cannot be predicted well without using partial labels.

Inference with Extra Labels In Figure 6, we show qualitative results on CUB-312 demonstrating the use of extra labels. In the CUB-312 dataset, the extra labels are high level concepts of bird species that are not target labels. In these examples, we first show the predictions for C-Tran without using extra labels labels, and the last column shows the C-Tran predictions when using $\epsilon = 54\%$ of the extra labels (which is 60 labels for CUB-312) as observed, or known prior to inference. We can see that many bird species predictions are completely changed after using the extra labels as input to our model.

A.2. Detailed Diagram of C-Tran Settings

In Figure 7 shows a detailed diagram of all possible training and inference settings used in our paper, and how C-Tran is used in each setting. By using the same random mask training, we can apply our model to any of the three inference settings.

A.3. Multi-Label Classification Metrics

$$\begin{aligned}\text{OP} &= \frac{\sum_i N_i^c}{\sum_i N_i^p} \\ \text{OR} &= \frac{\sum_i N_i^c}{\sum_i N_i^g} \\ \text{OF1} &= \frac{2 \times \text{OP} \times \text{OR}}{\text{OP} + \text{OR}}\end{aligned} \tag{7}$$

$$\begin{aligned}\text{CP} &= \frac{1}{C}\sum_i \frac{N_i^c}{N_i^p} \\ \text{CR} &= \frac{1}{C}\sum_i \frac{N_i^c}{N_i^g} \\ \text{CF1} &= \frac{2 \times \text{CP} \times \text{CR}}{\text{CP} + \text{CR}}\end{aligned} \tag{8}$$

where C is the number of labels, N_i^c is true positives for the i-th label, N_i^p is the total number of images for which the i-th label is predicted, and N_i^g is the number of ground truth images for the i-th label.

A.4. More Discussions of C-Tran

Connecting to Transformers and BERT

Our proposed method, C-Tran, draws much inspiration from works in natural language processing. The transformer model [47] proposed "self attention" for natural language translation. Self attention allows each word in the target sentence to attend to all other words (both in the source sentence and the target sentence) for translation. [13] introduced BERT for language modeling. BERT uses self attention with masked words to pretrain a language model.

Self attention and BERT are both examples of complete graphs, but on sentences rather than image features and labels. C-Tran uses the same self-attention mechanisms as [47] and [13], but instead of using only the word embeddings from a sentence, we use feature and label embeddings.

In computer vision, [5] used Transformers for object detection. Our method varies in several distinct ways. First, we are primarily interested in using partial evidence for image classification, and our unique state embeddings allow C-Tran to use such evidence. Second, we model image and label features jointly in a Transformer encoder, whereas [5] use an encoder/decoder framework. Our method allows the image features to be updated conditioned on the labels, which is a key characteristic of our model.

Connecting to Graph Based Neural Relational Learning Another line of recent works employ object localization techniques[56, 52] or attention mechanism[51, 58] to locate semantic meaningful regions and try to identify underlying relations between regions and outputs. However, these methods either require expensive bounding box annotations

Images	True Labels	ResNet-101	C-Tran	C-Tran + partial labels	
 ID:000000362831	fork knife, spoon, bowl, chair, diningtable	**fork,** sandwich, **diningtable,** **spoon,** cup	**fork,** **knife,** **diningtable,** person, cake	spoon=1, trafficlight=0, bench=0, dog=0, ...	**fork,** **knife,** **diningtable,** person, **bowl**
 ID:000000106216	person, car, truck, parkingmeter, horse,	**person,** **car,** **truck,** **horse,** bicycle	**car,** **person,** **truck,** **horse,** bicycle	bicycle=0, motorcycle=0, train=0, boat=0 ...	**car,** **person,** **truck,** **horse,** **parkingmeter**
 ID:000000243213	person, bench, backpack, tennisracket, bottle, chair	**person,** **tennisracket,** **chair,** tie, sportsball	**person,** **tennisracket,** **chair,** sportsball, **bench**	backpack=1 parkingmeter=0, bird=0, zebra=0, ...	**person,** **tennisracket,** **chair,** **bottle,** **bench**
 ID:000000170129	airplane, train	**airplane,** boat, car, truck, person	**airplane**, boat, person, car, bird	car=0, motorcycle=0, bus=0, truck=0, ...	**airplane**, boat, person, bird, **train**
 ID: 000000262896	bottle, spoon, diningtable, cellphone, book	**bottle,** fork, **diningtable,** bowl, **spoon**	fork, **spoon**, bowl, **book,** **diningtable**	diningtable=1, bicycle=0, car=0, truck=0, ...	**spoon**, bowl, **book,** **bottle,** **cellphone**

Figure 5. Qualitative examples of C-Tran + partial labels on the COCO-80 dataset. In the last column, we use $\epsilon = 25\%$ partial labels, some of which are shown. Correctly predicted labels are in bold.

or merely get regions of interest roughly due to the lack of label supervision. One recent study by [55] also showed that modeling the associations between image feature regions and labels helps to improve multi-label performance. In our work, C-Tran uses graph attentions and enables each target label to attend differentially to relevant parts of an input image.

For multi-label classfication(MLC), [18] formulate MLC using a label graph and they introduced a conditional dependency SVM where they first trained separate classifiers for each label given the input and all other true labels and used Gibbs sampling to find the optimal label set. The main drawback is that this method requires separate classifiers for each label. [41] proposes a method to label the pairwise edges of randomly generated label graphs, and requires some chosen aggregation method over all random graphs. The authors introduce the idea that variation in the graph structure shifts the inductive bias of the base learners. One recent study [31] used graph neural networks for multi-label classification on sequential inputs. The proposed method models the label-to-label dependencies using GNNs, however, does not represent input features and labels in one coherent graph. A key aspect of C-Tran is that the Transformer encoder can be viewed as a fully connected graph which is able to learn any relationships between features and labels. The Transformer attention mechanism can be regarded as a form of graph ensemble learning [19]. Above all, previous methods using graphs to model label dependencies do not allow for partial evidence information to be included in the prediction.

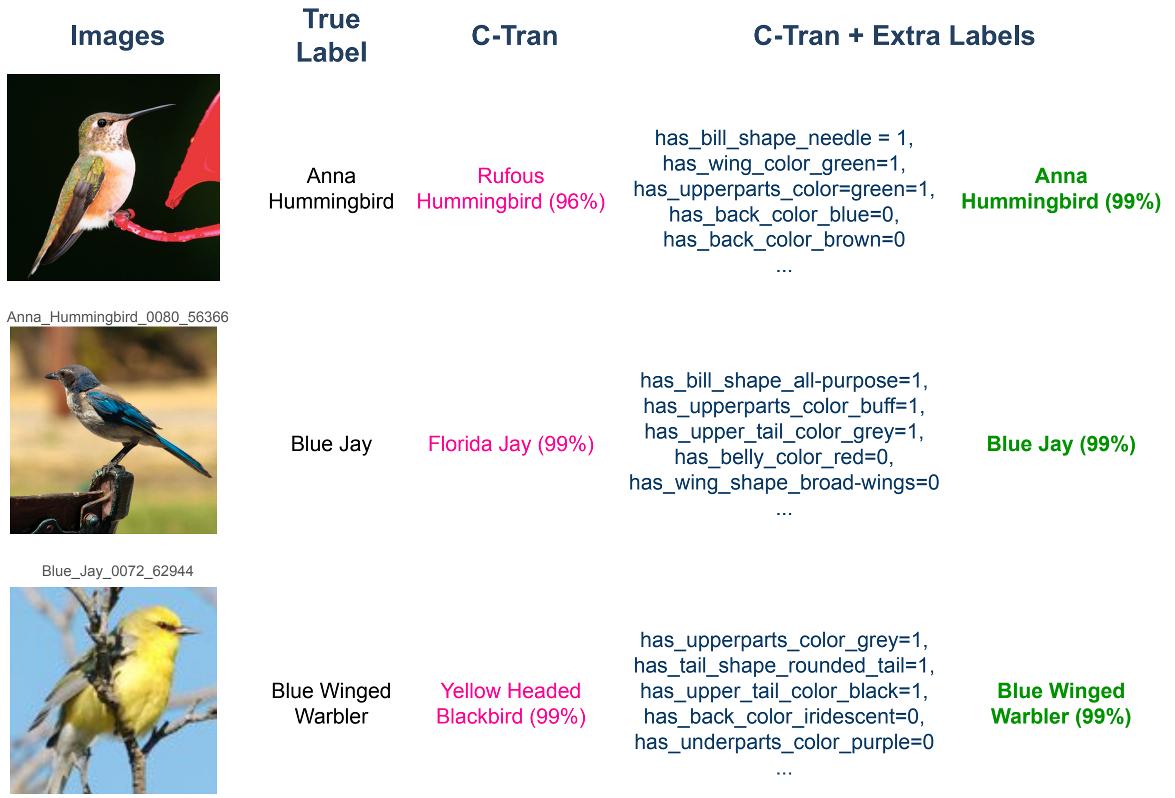
Images
True Label
C-Tran
C-Tran + Extra Labels
Anna Hummingbird
Rufous Hummingbird (96%)
has_bill_shape_needle = 1,
has_wing_color_green=1,
has_upperparts_color=green=1,
has_back_color_blue=0,
has_back_color_brown=0
...
Anna Hummingbird (99%)
Anna_Hummingbird_0080_56366
Blue Jay
Florida Jay (99%)
has_bill_shape_all-purpose=1,
has_upperparts_color_buff=1,
has_upper_tail_color_grey=1,
has_belly_color_red=0,
has_wing_shape_broad-wings=0
...
Blue Jay (99%)
Blue_Jay_0072_62944
Blue Winged Warbler
Yellow Headed Blackbird (99%)
has_upperparts_color_grey=1,
has_tail_shape_rounded_tail=1,
has_upper_tail_color_black=1,
has_back_color_iridescent=0,
has_underparts_color_purple=0
...
Blue Winged Warbler (99%)

Figure 6. Qualitative examples of C-Tran + extra labels on the CUB-312 dataset. In the last column, we use $\epsilon = 54\%$ extra labels, some of which are shown.

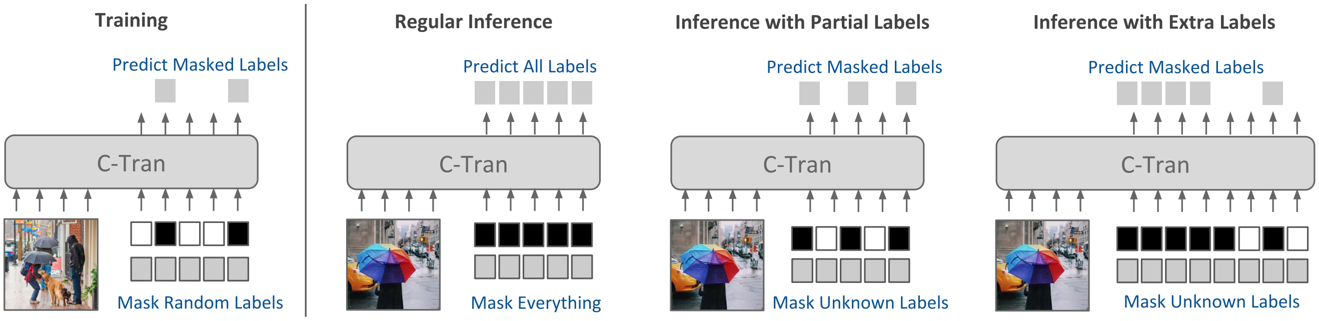

Figure 7. Detailed example of the general training method and three different inference settings where C-Tran can be applied.