Deep Dependency Networks and Advanced Inference Schemes for Multi-Label Classification

Shivvrat Arya
The University of Texas at Dallas

Yu Xiang
The University of Texas at Dallas

Vibhav Gogate
The University of Texas at Dallas

Abstract

We present a unified framework called deep dependency networks (DDNs) that combines dependency networks and deep learning architectures for multi-label classification, with a particular emphasis on image and video data. The primary advantage of dependency networks is their ease of training, in contrast to other probabilistic graphical models like Markov networks. In particular, when combined with deep learning architectures, they provide an intuitive, easy-to-use loss function for multi-label classification. A drawback of DDNs compared to Markov networks is their lack of advanced inference schemes, necessitating the use of Gibbs sampling. To address this challenge, we propose novel inference schemes based on local search and integer linear programming for computing the most likely assignment to the labels given observations. We evaluate our novel methods on three video datasets (Charades, TACoS, Wetlab) and three image datasets (MS-COCO, PASCAL VOC, NUS-WIDE), comparing their performance with (a) basic neural architectures and (b) neural architectures combined with Markov networks equipped with advanced inference and learning techniques. Our results demonstrate the superiority of our new DDN methods over the two competing approaches.

1 INTRODUCTION

In this paper, we focus on the multi-label classification (MLC) task and more specifically, on its two notable instantiations, multi-label action classification (MLAC) for videos and multi-label image classification (MLIC). At a high level, given a pre-defined set of labels (or actions) and a test example (video or image), the goal is to assign each test example to a subset of labels. It is well known that MLC is notoriously difficult because, in practice, the labels are often correlated, and thus, predicting them independently may lead to significant errors. Therefore, most advanced methods explicitly model the relationship or dependencies between the labels, using either probabilistic techniques (Antonucci et al., 2013; Di Mauro et al., 2016; Guo and Xue, 2013; Tan et al., 2015; Wang et al., 2008, 2014) or non-probabilistic/neural methods (Chen et al., 2019a,b; Kong et al., 2013; Liu et al., 2022, 2021; Nguyen et al., 2021; Papagiannopoulou et al., 2015; Qu et al., 2021; Wang et al., 2021a,b; Weng et al., 2023; Zhou et al., 2023).

To this end, motivated by approaches that combine probabilistic graphical models (PGMs) with neural networks (NNs) (Johnson et al., 2016; Krishnan et al., 2015), we jointly train a hybrid model, termed *deep dependency networks* (DDNs) (Guo and Weng, 2020), as illustrated in Figure 1. In a Deep Dependency Network (DDN), a conditional dependency network sits on top of a neural network. The underlying neural network transforms input data (e.g., an image) into a feature set. The dependency network (Heckerman et al., 2000) then utilizes these features to establish a local conditional distribution for each label, considering not only the features but also the other labels. Thus, at a high level, a DDN is a *neuro-symbolic model* where the neural network extracts features from data and the dependency network acts as a symbolic counterpart, learning the weighted constraints between the labels.

However, a limitation of DDNs is that they rely on naive techniques such as Gibbs sampling and mean-field inference for probabilistic reasoning and lack advanced probabilistic inference techniques (Lowd, 2012; Lowd and Shamaei, 2011). This paper addresses these limitations by introducing sophisticated inference schemes tailored for the Most Probable Explanation (MPE) task in DDNs, which involves finding the most likely assignment to the unobserved variables given observations. In essence, a solution to the MPE task, when applied to a probabilistic model defined over labels and observed variables, effectively solves the multi-label classification problem.

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

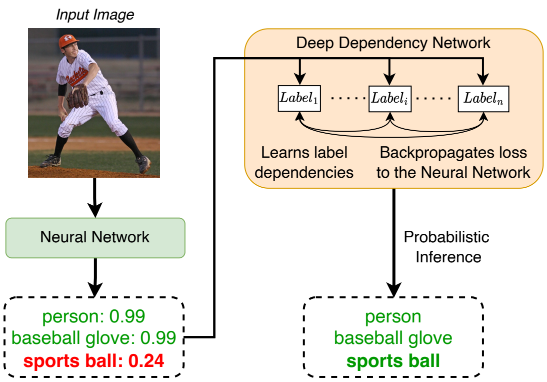


Figure 1: Illustrating improvements from our new inference schemes for DDNs. The DDN learns relationships between labels, and the inference schemes reason over them to accurately identify concealed objects, such as **sports ball**.

More specifically, we propose two new methods for MPE inference in DDNs. Our first method uses a random-walk based local search algorithm. Our second approach uses a piece-wise approximation of the log-sigmoid function to convert the non-linear MPE inference problem in DDNs into an integer linear programming problem. The latter can then be solved using off-the-shelf commercial solvers such as Gurobi (Gurobi Optimization, LLC, 2023).

We evaluate DDNs equipped with our new MPE inference schemes and trained via joint learning on three video datasets: Charades (Sigurdsson et al., 2016), TACoS (Regneri et al., 2013), and Wetlab (Naim et al., 2014), and three image datasets: MS-COCO (Lin et al., 2014), PASCAL VOC 2007 (Everingham et al., 2010), and NUS-WIDE (Chua et al., 2009). We compare their performance to two categories of models: (a) basic neural networks (without dependency networks) and (b) hybrids of Markov random fields (MRFs), an undirected PGM, and neural networks equipped with sophisticated reasoning and learning algorithms. Specifically, we employ three advanced approaches: (1) iterative join graph propagation (IJGP) (Mateescu et al., 2010), a type of generalized Belief propagation method (Yedidia et al., 2000) for marginal inference, (2) integer linear programming (ILP) based techniques for computing most probable explanations (MPE) and (3) a well-known structure learning method based on logistic regression with ℓ_1-regularization (Lee et al., 2006; Wainwright et al., 2006) for pairwise MRFs.

Via a detailed experimental evaluation, we found that, generally speaking, the MRF+NN hybrids outperform NNs, as measured by metrics such as Jaccard index and subset accuracy, especially when advanced inference methods such as IJGP are employed. Additionally, DDNs, when equipped with our novel MILP-based MPE inference approach, often outperform both MRF+NN hybrids and NNs. This enhanced performance of DDNs with advanced MPE solvers is likely attributed to their superior capture of dense label interdependencies, a challenge for MRFs. Notably, MRFs rely on sparsity for efficient inference and learning.

2 PRELIMINARIES

A **log-linear model** or a **Markov random field** (MRF), denoted by $\mathcal{M}$, is an undirected probabilistic graphical model (Koller and Friedman, 2009) that is widely used in many real-world domains for representing and reasoning about uncertainty. MRFs are defined as a triple $\langle \mathbf{X}, \mathcal{F}, \Theta \rangle$ where $\mathbf{X} = \{X_1, \ldots, X_n\}$ is a set of Boolean random variables, $\mathcal{F} = \{f_1, \ldots, f_m\}$ is a set of features such that each feature f_i (we assume that a feature is a Boolean formula) is defined over a subset $\mathbf{D}_i$ of $\mathbf{X}$, and $\Theta = (\theta_1, \ldots, \theta_m)$ are real-valued weights or parameters, namely $\forall \theta_i \in \Theta;\ \theta_i \in \mathbb{R}$ such that each feature f_i is associated with a parameter θ_i. $\mathcal{M}$ represents the following probability distribution:

$$P(\mathbf{x}) = \frac{1}{Z(\Theta)} \exp\left\{\sum_{i=1}^{m} \theta_i f_i\left(\mathbf{x}_{\mathbf{D}_i}\right)\right\} \tag{1}$$

where $\mathbf{x}$ is an assignment of values to all variables in $\mathbf{X}$, $\mathbf{x}_{\mathbf{D}_i}$ is the projection of $\mathbf{x}$ on the variables $\mathbf{D}_i$ of f_i, $f_i(\mathbf{x}_{\mathbf{D}_i})$ is an *indicator function* that equals 1 when $\mathbf{x}_{\mathbf{D}_i}$ evaluates f_i to `True` and is 0 otherwise, and $Z(\Theta)$ is the normalization constant called the *partition function*.

We focus on three tasks over MRFs: (1) structure learning; (2) posterior marginal inference; and (3) finding the most likely assignment to all the non-evidence variables given evidence (this task is often called most probable explanation or MPE inference in short). All of these tasks are at least NP-hard in general, and therefore approximate methods are often preferred over exact ones in practice.

A popular and fast method for structure learning is to learn binary pairwise MRFs (MRFs in which each feature is defined over at most two variables) by training an ℓ_1-regularized logistic regression classifier for each variable given all other variables as features (Lee et al., 2006; Wainwright et al., 2006). ℓ_1-regularization induces sparsity in that it encourages many weights to take the value zero. All non-zero weights are then converted into conjunctive features. Each conjunctive feature evaluates to `True` if both variables are assigned the value 1 and to `False` otherwise. Popular approaches for posterior marginal inference are the Gibbs sampling algorithm and generalized Belief propagation (Yedidia et al., 2000) techniques such as Iterative Join Graph Propagation (Mateescu et al., 2010). For MPE inference, a popular approach is to encode it as an integer linear programming (ILP) problem (Koller and Friedman, 2009) and then use off-the-shelf approaches such as Gurobi Optimization, LLC (2023) to solve the ILP.

Dependency Networks (DNs) (Heckerman et al., 2000) represent the joint distribution using a set of local conditional probability distributions, one for each variable. Each conditional distribution defines the probability of a variable given all of the others. A DN is consistent if there exists a joint probability distribution $P(\mathbf{x})$ such that all conditional distributions $P_i(x_i|\mathbf{x}_{-i})$ where $\mathbf{x}_{-i}$ is the projection of $\mathbf{x}$ on $\mathbf{X} \setminus \{X_i\}$, are conditional distributions of $P(\mathbf{x})$.

A DN is learned from data by learning a classifier (e.g., logistic regression, multi-layer perceptron, etc.) for each variable, and thus DN learning is embarrassingly parallel. However, because the classifiers are independently learned from data, we often get an inconsistent DN. It has been conjectured (Heckerman et al., 2000) that most DNs learned from data are almost consistent in that only a few parameters need to be changed in order to make them consistent.

The most popular inference method over DNs is *fixed-order* Gibbs sampling (Liu, 2008). If the DN is consistent, then its conditional distributions are derived from a joint distribution $P(\mathbf{x})$, and the stationary distribution (namely the distribution that Gibbs sampling converges to) will be the same as $P(\mathbf{x})$. If the DN is inconsistent, then the stationary distribution of Gibbs sampling will be inconsistent with the conditional distributions.

3 TRAINING DEEP DEPENDENCY NETWORKS

In this section, we detail the training process for our proposed Deep Dependency Network model (Guo and Weng, 2020), the hybrid framework for multi-label action classification in videos and multi-label image classification. Two key components are trained jointly: a neural network and a conditional dependency network. The neural network is responsible for extracting high-quality features from video segments or images, while the dependency network models the relationships between these features and their corresponding labels, supplying the neural network with gradient information regarding these relationships.

3.1 Framework

Let $\mathbf{V}$ denote the set of random variables corresponding to the pixels and $\mathbf{v}$ denote the RGB values of the pixels in a frame or a video segment. Let $\mathbf{E}$ denote the (continuous) output nodes of a neural network which represents a function $\mathcal{N} : \mathbf{v} \mapsto \mathbf{e}$, that takes $\mathbf{v}$ as input and outputs an assignment $\mathbf{e}$ to $\mathbf{E}$. Let $\mathbf{X} = \{X_1, \ldots, X_n\}$ denote the set of indicator variables representing the labels. For simplicity, we assume that $|\mathbf{E}| = |\mathbf{X}| = n$. Given $(\mathbf{V}, \mathbf{E}, \mathbf{X})$, a deep dependency network (DDN) is a pair $\langle \mathcal{N}, \mathcal{D} \rangle$ where $\mathcal{N}$ is a neural network that maps $\mathbf{V} = \mathbf{v}$ to $\mathbf{E} = \mathbf{e}$ and $\mathcal{D}$ is a conditional dependency network (Guo and Gu, 2011) that models $P(\mathbf{x}|\mathbf{e})$ where $\mathbf{e} = \mathcal{N}(\mathbf{v})$.

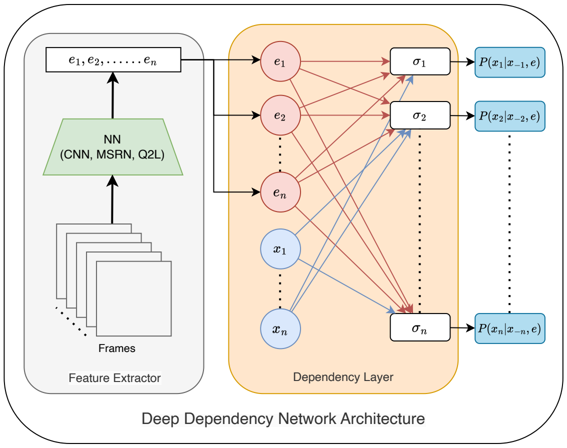


Figure 2: Illustration of Dependency Network for multi-label video classification. The NN takes video clips (frames) as input and outputs the features $e_1, e_2, ..., e_n$ (denoted by red colored nodes). These features are then used by the sigmoid output ($\sigma_1, \ldots, \sigma_n$) of the dependency layer to model the local conditional distributions.

The conditional dependency network represents the distribution $P(\mathbf{x}|\mathbf{e})$ using a collection of local conditional distributions $P_i(x_i|\mathbf{x}_{-i}, \mathbf{e})$, one for each label X_i, where $\mathbf{x}_{-i} = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$.

Thus, a DDN is a discriminative model and represents the conditional distribution $P(\mathbf{x}|\mathbf{v})$ using several local conditional distributions $P(x_i|\mathbf{x}_{-i}, \mathbf{e})$ and makes the following conditional independence assumptions $P(x_i|\mathbf{x}_{-i}, \mathbf{v}) = P(x_i|\mathbf{x}_{-i}, \mathbf{e})$ where $\mathbf{e} = \mathcal{N}(\mathbf{v})$. Figure 2 demonstrates the DDN architecture.

3.2 Learning

We employ the conditional pseudo log-likelihood loss (CPLL) (Besag, 1975) in order to jointly train the two components of DDN, drawing inspiration from the DDN training approach outlined by Guo and Weng (2020). For each training example $(\mathbf{v}, \mathbf{x})$, we send the video/image through the neural network to obtain a new representation $\mathbf{e}$ of $\mathbf{v}$. In the dependency layer, we learn a classifier for each label X_i to model the conditional distribution $P_i(x_i|\mathbf{x}_{-i}, \mathbf{e})$. More precisely, with the representation of the training instance $(\mathbf{e}, \mathbf{x})$, each sigmoid output of the dependency layer indexed by i and denoted by σ_i (see figure 2) uses X_i as the class variable and $(\mathbf{E} \cup \mathbf{X}_{-i})$ as the attributes. The joint training of the model is achieved through CPLL applied to the outputs of the dependency layer (σ_i's).

Let Θ represent the parameter set of the DDN. We employ gradient-based optimization methods (e.g., backpropagation), to minimize the Conditional Pseudo-Likelihood

(CPLL) loss function given below

$$\mathcal{L}(\Theta, \mathbf{v}, \mathbf{x}) = -\sum_{i=1}^{n} \log P_i(x_i|\mathbf{e} = \mathcal{N}(\mathbf{v}), \mathbf{x}_{-i}; \Theta) \quad (2)$$

4 MPE INFERENCE IN DDNs

Unlike a conventional discriminative model, such as a neural network, in a DDN, we cannot predict the output labels by simply making a forward pass over the network. This is because each sigmoid output σ_i (which yields a probability distribution over X_i) of the dependency layer requires an assignment $\mathbf{x}_{-i}$ to all labels except x_i and $\mathbf{x}_{-i}$ is not available at prediction time. Consequently, it is imperative to employ specialized techniques for obtaining output labels in multilabel classification tasks within the context of DDNs.

Given a DDN representing a distribution $P(\mathbf{x}|\mathbf{e})$ where $\mathbf{e} = \mathcal{N}(\mathbf{v})$, the multilabel classification task can be solved by finding the most likely assignment to all the unobserved variables based on a set of observed variables. This task is also called the most probable explanation (MPE) task. Formally, we seek to find $\mathbf{x}^*$ such that:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x} \mid \mathbf{e}) \quad (3)$$

Next, we present three algorithms for solving the MPE task.

4.1 Gibbs Sampling

To perform MPE inference, we first send the video (or frame) through the neural network to yield an assignment $\mathbf{e}$ to all variables in $\mathbf{E}$. Then given $\mathbf{e}$, we generate N samples $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)})$ via Gibbs sampling (see for example Koller and Friedman (2009)), a classic MCMC technique. These samples can then be used to estimate the marginal probability distribution of each label X_i using the following mixture estimator (Liu, 2008):

$$\hat{P}_i\left(x_i|\mathbf{v}\right) = \frac{1}{N}\sum_{j=1}^{N} P_i\left(x_i \mid \mathbf{x}_{-i}^{(j)}, \mathbf{e}\right) \quad (4)$$

Given an estimate of the marginal distribution $\hat{P}_i\left(x_i|\mathbf{v}\right)$ for each variable X_i, we can estimate the MPE assignment using the standard max-marginal approximation:

$$\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{e}) \approx \prod_{i=1}^{n} \max_{x_i} \hat{P}(x_i|\mathbf{e})$$

In other words, we can construct an approximate MPE assignment by finding the value x_i for each variable that maximizes $\hat{P}_i\left(x_i|\mathbf{e}\right)$.

4.2 Local Search Based Methods

Local search algorithms (see for example Selman et al. (1993)) systematically examine the solution space by making localized adjustments, with the objective of either finding an optimal solution or exhausting a pre-established time limit. They offer a viable approach for solving the MPE inference task in DDNs. These algorithms, through their structured exploration and score maximization, are effective in identifying near-optimal label configurations.

In addressing the MPE inference within DDNs, we define the objective function for local search as $\sum_{i=1}^{n} \log\left(P_i(x_i \mid \mathbf{x}_{-i}, \mathbf{e})\right)$, where $\mathbf{e}$ is the evidence provided by $\mathcal{N}$. We propose to use two distinct local search strategies for computing the MPE assignment: random walk and greedy local search.

In *random walk (RW)*, the algorithm begins with a random assignment to the labels and at each iteration, flips a random label (changes the value of the label from a 1 to a 0 or a 0 to a 1) to yield a new assignment. At termination, the algorithm returns the assignment with the highest score explored during the random walk. In *greedy local search*, the algorithm begins with a random assignment to the labels, and at each iteration, with probability p flips a random label to yield a new assignment and with probability $1 - p$ flips a label that yields the maximum improvement in the score. At termination, the algorithm returns the assignment with the highest score explored during its execution.

4.3 Multi-Linear Integer Programming

In this section, we present a novel approach for MPE inference in DDNs by formulating the problem as a Second-Order Multi-Linear Integer Programming task. Specifically, we show that the task of maximizing the scoring function $\sum_{i=1}^{n} \log\left(P_i(x_i \mid \mathbf{x}_{-i}, \mathbf{e})\right)$ is equivalent to the following optimization problem (derivation is provided in the supplementary material):

$$\begin{aligned}
&\underset{\mathbf{x},\mathbf{z}}{\text{maximize}} \sum_{i=1}^{n} \left(x_i \log P_i + (1-x_i)\log(1-P_i)\right) \\
&\text{subject to:} \\
&\quad P_i = \sigma(z_i), \quad \forall i \in \{1,\ldots,n\} \\
&\quad z_i = \sum_{j=1}^{|\mathbf{e}|} w_{ij} e_j + \sum_{\substack{k=1 \\ k\neq i}}^{|\mathbf{x}|} v_{ik} x_k + b_i, \quad \forall i \in \{1,\ldots,n\} \\
&\quad x_i \in \{0,1\}, \quad \forall i \in \{1,\ldots,n\}
\end{aligned} \quad (5)$$

Here, $P_i = P_i(x_i|\mathbf{x}_{-i}, \mathbf{e})$ and w_{ij}'s and v_{ik}'s denote the weights associated with $\mathbf{e}$ and $\mathbf{x}$ for P_i, respectively. The bias term for each P_i is denoted by b_i. In this context, z_i represents the values acquired prior to applying the sigmoid activation function. Substituting the constraint

$P_i(x_i|\mathbf{x}_{-i},\mathbf{e}) = \sigma(z_i) = \frac{1}{1+e^{(-z_i)}}$ in the objective and simplifying, we get

$$\underset{\mathbf{x},\mathbf{z}}{\text{maximize}} \sum_{i=1}^{n} x_i z_i - \log\left(1+e^{z_i}\right) \tag{6}$$

subject to:

$$z_i = \sum_{j=1}^{|\mathbf{e}|} w_{ij} e_j + \sum_{\substack{k=1\\k\neq i}}^{|\mathbf{x}|} v_{ik} x_k + b_i, \quad \forall i \in \{1,\ldots,n\} \tag{7}$$

$$x_i \in \{0,1\}, \quad \forall i \in \{1,\ldots,n\} \tag{8}$$

The optimal value for $\mathbf{x}$ corresponds to the solution of this optimization problem. The second term in the objective specified in equation 6 comprises logarithmic and exponential functions, which are non-linear, thereby making it a *non-linear optimization problem*. To address this non-linearity, we propose the use of piece-wise linear approximations(Asghari et al., 2022; Geißler et al., 2012; Kumar, 2007; Li et al., 2022; Lin et al., 2013; Rovatti et al., 2014) for these terms. Further details about the piece-wise linear approximation can be found in the supplement. Let $g(z_i)$ represent the piece-wise linear approximation of $\log\left(1+e^{z_i}\right)$, then the optimization problem can be expressed as:

$$\begin{aligned} &\underset{\mathbf{x},\mathbf{z}}{\text{maximize}} \sum_{i=1}^{n} x_i z_i - g(z_i) \\ &\text{subject to:} \\ &z_i = \sum_{j=1}^{|\mathbf{e}|} w_{ij} e_j + \sum_{\substack{k=1\\k\neq i}}^{|\mathbf{x}|} v_{ik} x_k + b_i, \quad \forall i \in \{1,\ldots,n\} \\ &x_i \in \{0,1\}, \quad \forall i \in \{1,\ldots,n\} \end{aligned} \tag{9}$$

The optimization problem given in equation 9 is an integer multilinear program of order 2 because it includes terms of the form $x_i x_j$ where both x_i and x_j take values from the set $\{0,1\}$. Since $x_i x_j$ corresponds to a "logical and" between two Boolean variables, all such expressions can be easily encoded as linear constraints yielding an integer linear program (ILP) (see the supplementary material for an example).

In our experiments, we solved the ILP using Gurobi (Gurobi Optimization, LLC, 2023), which is an anytime ILP solver. Another useful feature of the ILP formulation is that we can easily incorporate prior knowledge (e.g., an image may not have more than ten objects/labels) into the ILP under the assumption that the knowledge can be reliably modeled using linear constraints.

5 EXPERIMENTAL EVALUATION

In this section, we evaluate the proposed methods on two multi-label classification tasks: (1) multi-label activity classification using three video datasets; and (2) multi-label image classification using three image datasets. We begin by describing the datasets and metrics, followed by the experimental setup, and conclude with the results. All models were implemented utilizing PyTorch and were trained and evaluated on a machine with an NVIDIA A40 GPU and an Intel(R) Xeon(R) Silver 4314 CPU.

5.1 Datasets and Metrics

We evaluated our algorithms on three video datasets: (1) Charades (Sigurdsson et al., 2016); (2) TACoS (Regneri et al., 2013); and (3) Wetlab (Naim et al., 2015). Charades dataset comprises videos of people performing daily indoor activities while interacting with various objects. We adopted the train-test split instructions given in PySlowFast (Fan et al., 2020) with 7,986 training and 1,863 validation videos. TACoS consists of third-person videos of a person cooking in a kitchen. The dataset comes with hand-annotated labels of actions, objects, and locations for each video frame. From the complete set of these labels, we selected 28 labels resulting in 60,313 training frames and 9,355 test frames across 17 videos. Wetlab features experimental activities in labs, consisting of five training videos (100,054 frames) and one test video (11,743 frames) with 57 distinct labels.

For multi-label image classification (MLIC), we examined: (1) MS-COCO (Lin et al., 2014); (2) PASCAL VOC 2007 (Everingham et al., 2010); and (3) NUS-WIDE (Chua et al., 2009). MS-COCO, a well-known dataset for detection and segmentation, comprises 122,218 labeled images with an average of 2.9 labels per image. We used its 2014 version. NUS-WIDE is a real-world web image dataset that contains 269,648 images from Flickr. Each image has been manually annotated with a subset of 81 visual classes that include objects and scenes. PASCAL VOC 2007 contains 5,011 train-validation and 4,952 test images, with each image labeled with one or more of the 20 available object classes.

We follow the instructions provided in (Qu et al., 2021) to do the train-test split for NUS-WIDE and PASCAL VOC. We evaluated the performance on the TACoS, Wetlab, MS-COCO, NUS-WIDE, and VOC datasets using Subset Accuracy (SA), Jaccard Index (JI), Hamming Loss (HL), Macro F1 Score (Macro F1), Micro F1 Score (Micro F1) and F1 Score (F1). With the exception of Hamming Loss, superior performance is indicated by higher scores in all considered metrics. We omit SA for the Charades dataset due to the infeasibility of achieving reasonable scores given its large label space. Additionally, Hamming Loss (HL) has been excluded for the MS-COCO dataset as the performance of all methods is indistinguishable.

Given that the focus in MPE inference is on identifying the most probable label configurations rather than individual label probabilities, the use of Mean Average Precision (mAP) as a performance metric is not applicable to our study. Therefore, our primary analysis relies on SA, JI, HL, Macro F1, Micro F1, and F1. Precision-based metrics are detailed in the supplementary material.

5.2 Experimental Setup and Methods

We used three types of architectures in our experiments: (1) Baseline neural networks, which are specific to each dataset; (2) neural networks augmented with MRFs, which we will refer to as deep random fields or DRFs in short; and (3) a dependency network on top of the neural networks called deep dependency networks (DDNs).

Neural Networks. We choose four different types of neural networks, and they act as a baseline for the experiments and as a feature extractor for DRFs and DDNs. Specifically, we experimented with: (1) 2D CNNs, (2) 3D CNNs, (3) transformers, and (4) CNNs with attention module and graph attention networks (GAT) (Veličković et al., 2018). This helps us show that our proposed method can improve the performance of a wide variety of neural architectures, even those which model label relationships, because unlike the latter, it performs probabilistic inference.

For the Charades dataset, we use the PySlowFast (Fan et al., 2020) implementation of the SlowFast Network (Feichtenhofer et al., 2019) (a state-of-the-art 3D CNN for video classification). For TACoS and Wetlab datasets, we use InceptionV3 (Szegedy et al., 2016), a state-of-the-art 2D CNN model for image classification. For the MS-COCO dataset, we used Query2Label (Q2L) (Liu et al., 2021), which uses transformers to pool class-related features. Q2L also learns label embeddings from data to capture the relationships between the labels. Finally, we used the multi-layered semantic representation network (MSRN) (Qu et al., 2021) for NUS-WIDE and PASCAL VOC. MSRN also models label correlations and learns semantic representations at multiple convolutional layers. We adopt pre-trained models and hyper-parameters from existing repositories for Charades, MS-COCO, NUS-WIDE, and PASCAL VOC. For TaCOS and Wetlab datasets, we fine-tuned an InceptionV3 model that was pre-trained on the ImageNet dataset.

Deep Random Fields (DRFs). As a baseline, we used a model that combines MRFs with neural networks. This DRF model is similar to DDN except that we use an MRF instead of a DN to compute $P(\mathbf{x}|\mathbf{e})$. We trained the MRFs generatively; namely, we learned a joint distribution $P(\mathbf{x}, \mathbf{e})$, which can be used to compute $P(\mathbf{x}|\mathbf{e})$ by instantiating evidence. We chose generative learning because we learned the structure of the MRFs from data, and discriminative structure learning is slow in practice (Koller and Friedman, 2009).

For inference over MRFs, we used Gibbs sampling (GS), Iterative Join Graph Propagation (IJGP) (Mateescu et al., 2010), and Integer Linear Programming (ILP) methods. Thus, three versions of DRFs corresponding to the inference scheme were used. We refer to these schemes as DRF-GS, DRF-ILP, and DRF-IJGP, respectively. Note that IJGP and ILP are advanced schemes, and we are unaware of their use for multi-label classification. Our goal is to test whether advanced inference schemes help improve the performance of deep random fields.

Deep Dependency Networks (DDNs). We trained the Deep Dependency Networks (DDNs) using the joint learning loss described in equation 2. We examined four unique inference methods for DDNs: (1) DDN-GS, employing Gibbs Sampling; (2) DDN-RW, leveraging a random walk local search; (3) DDN-Greedy, implementing a greedy local search; and (4) DDN-ILP, utilizing Integer Linear Programming to optimize the objective given in equation 9. It is noteworthy that, until now, only DDN-GS has been used for inference in Dependency Networks. DDN-RW and DDN-Greedy, which are general-purpose local search techniques, are our proposals for MPE inference on Dependency Networks. Lastly, DDN-ILP introduces a novel approach using optimization techniques with the objective of enhancing MPE inference on dependency networks.

Hyperparameters. For DRFs, in order to learn a sparse structure (using the logistic regression with ℓ_1 regularization method of Wainwright et al. (2006)), we increased the regularization constant associated with the ℓ_1 regularization term until the number of neighbors of each node in $\mathcal{G}$ is bounded between 2 and 10. We enforced this sparsity constraint in order to ensure that the inference schemes, specifically IJGP and ILP, are accurate and the model does not overfit the training data. IJGP, ILP, and GS are anytime methods; for them, we used a time-bound of one minute per example.

For DDNs, we used ℓ_1 regularization for the dependency layer. Through cross-validation, we selected the regularization constants from the set $\{0.1, 0.01, 0.001\}$. In the context of joint learning, the learning rates of the DDN model were adjusted using an extended version of the learning rate scheduler from PySlowFast (Fan et al., 2020). We impose a strict time constraint of 60 seconds per example for DDN-GS, DDN-RW, DDN-Greedy, and DDN-ILP. The DDN-ILP method is solved using Gurobi Optimization, LLC (2023) and utilizes accurate piece-wise linear approximations for non-linear functions, subject to a predefined error tolerance of 0.001.

Table 1: Comparison of our methods with the feature extractor for MLAC. The best/second best values are bold/underlined.

Dataset	Metric	SlowFast	InceptionV3	DRF-GS	DRF-ILP	DRF-IJGP	DDN-GS	DDN-RW	DDN-Greedy	DDN-ILP
Charades	JI	0.29	-	0.22	0.31	0.32	0.31	0.30	0.31	**0.33**
	HL ↓	**0.052**	-	0.194	0.067	0.054	0.054	0.056	0.056	**0.052**
	Macro F1	0.32	-	0.16	0.18	0.19	0.30	0.13	0.33	**0.36**
	Micro F1	0.45	-	0.31	0.21	0.29	0.43	0.24	0.44	**0.47**
	F1	0.42	-	0.28	0.22	0.26	0.41	0.22	0.41	**0.44**
TACoS	SA	-	0.40	0.47	0.51	0.44	0.54	0.46	0.56	**0.63**
	JI	-	0.61	0.65	0.65	0.70	0.69	0.64	0.69	**0.72**
	HL ↓	-	0.082	0.044	**0.030**	0.043	0.041	0.042	0.040	0.040
	Macro F1	-	0.26	0.54	0.42	0.54	0.52	0.59	0.61	**0.62**
	Micro F1	-	0.48	0.75	0.73	**0.83**	0.74	0.77	0.80	0.79
	F1	-	0.44	0.70	0.66	**0.78**	0.68	0.75	0.75	0.76
Wetlab	SA	-	0.35	0.35	0.60	0.58	0.63	0.46	0.55	**0.65**
	JI	-	0.64	0.52	0.73	0.74	**0.78**	0.63	0.68	0.76
	HL ↓	-	0.017	0.058	0.014	0.014	**0.011**	0.025	0.014	0.014
	Macro F1	-	0.24	0.22	0.27	0.24	**0.28**	0.22	**0.28**	**0.28**
	Micro F1	-	0.76	0.69	0.81	**0.82**	0.80	0.66	0.80	0.81
	F1	-	0.72	0.68	0.77	0.77	0.79	0.68	0.79	**0.80**

Table 2: Comparison of our methods with the feature extractor for MLIC. The best/second best values are bold/underlined.

Dataset	Metric	Q2L	MSRN	DRF-GS	DRF-ILP	DRF-IJGP	DDN-GS	DDN-RW	DDN-Greedy	DDN-ILP
MS-COCO	SA	0.51	-	0.35	0.54	**0.55**	**0.55**	0.53	**0.55**	**0.55**
	JI	0.80	-	0.69	0.82	0.82	0.82	0.81	0.82	**0.83**
	Macro F1	**0.86**	-	0.81	0.84	0.85	**0.86**	0.82	0.85	0.85
	Micro F1	0.86	-	0.82	0.83	0.85	**0.87**	0.85	0.86	0.86
	F1	**0.88**	-	0.79	0.82	0.85	**0.88**	0.86	0.87	**0.88**
NUS-WIDE	SA	-	0.31	0.28	0.32	0.32	**0.33**	0.25	0.30	**0.33**
	JI	-	0.64	0.55	0.59	0.63	0.62	0.56	0.61	**0.65**
	HL ↓	-	**0.015**	0.020	0.016	0.017	0.016	0.016	0.016	**0.015**
	Macro F1	-	**0.56**	0.23	0.28	0.32	0.52	0.26	0.29	0.51
	Micro F1	-	0.73	0.63	0.70	0.71	0.71	0.70	0.72	**0.74**
	F1	-	**0.71**	0.62	0.67	0.69	0.69	0.68	0.69	**0.71**
PASCAL-VOC	SA	-	0.71	0.73	0.76	0.76	0.76	0.82	0.86	**0.89**
	JI	-	0.85	0.83	0.88	0.87	0.87	0.89	0.91	**0.95**
	HL ↓	-	0.015	0.021	0.019	0.019	0.008	**0.006**	0.007	**0.006**
	Macro F1	-	0.89	0.75	0.81	0.77	0.94	0.94	0.95	**0.96**
	Micro F1	-	0.90	0.85	0.87	0.86	0.94	0.94	0.95	**0.96**
	F1	-	0.91	0.83	0.87	0.86	0.96	0.93	0.96	**0.97**

5.3 Results

We compare the baseline neural networks with three versions of DRFs and four versions of DDNs using the six metrics and six datasets given in Section 5.1. The results are presented in tables 1 and 2.

Comparison between baseline neural network and DRFs. We observe that IJGP and ILP outperform the baseline neural networks (which include transformers for some datasets) in terms of JI, SA, and HL on four out of the six datasets. IJGP typically outperforms GS and ILP on JI. In terms of F1 metrics, IJGP and ILP methods tend to perform better than Gibbs Sampling (GS) approaches, but they are less effective than baseline NNs. ILP's superiority in SA—a metric that scores 1 for an exact label match and 0 otherwise—can be attributed to its precise most probable explanation (MPE) inference. An accurate MPE inference, when paired with a precise model, is likely to achieve high SA scores. Note that getting a higher SA is much harder in datasets having large number of labels. Specifically, SA does not distinguish between models that predict *almost* correct labels and completely incorrect outputs. We observe that advanced inference schemes, particularly IJGP and ILP, are superior on average to GS.

Comparison between baseline neural networks and DDNs. Our study has shown that the DDN model with the proposed MILP based inference method is superior to the baseline neural networks in four out of six datasets, with notable enhancements in SA and JI metrics, such as an 18%, 23%, and 30% improvement in SA for the PASCAL-VOC, TACoS, and Wetlab datasets, respectively. Moreover, the MILP-based method either significantly surpasses or matches all alternative inference strategies for DDNs. While Gibbs Sampling-based inference is typically better than the Random Walk based approach in DDNs, its performance against the greedy sampling method varies.

We observe that on the MLIC task, the DDN with advanced

MPE inference outperforms Q2L and MSRN, even though both Q2L and MSRN model label correlations. This suggests that DDNs are either able to uncover additional relationships between labels during the learning phase or better reason about them during the inference phase or both. In particular, both Q2L and MSRN do not use MPE inference to predict the labels because they do not explicitly model the joint probability distribution over the labels.

Figure 3 displays the images and their predicted labels by both Q2L and DDN-ILP on the MS-COCO dataset. Our method not only adds the labels omitted by Q2L but also eliminates several incorrect predictions. In the first two images, our approach rectifies label omissions by Q2L, conforming to the ground truth. In the third image, our method removes erroneous predictions. The last image illustrates a case where DDN underperforms compared to Q2L by failing to identify a ground-truth label. Additional instances are available in the appendix.

Comparison between DRFs and DDNs. Based on our observations, we found that jointly trained DDNs in conjunction with the proposed inference method consistently lead to superior performance compared to the top-performing DRFs across all datasets. Nonetheless, in certain situations, DRFs that employ advanced inference strategies produce results that closely match those of DDNs for both JI and SA, making DRFs a viable option, particularly when limited GPU resources are available for training and optimization of both JI and SA is prioritized.

In summary, the empirical results indicate that Deep Dependency Networks (DDNs) equipped with advanced inference strategies consistently outperform conventional neural networks and MRF+NN hybrids in multi-label classification tasks. Furthermore, these advanced inference methods surpass traditional sampling-based techniques for DDNs. The superior performance of both DDNs and DRFs utilizing advanced inference techniques supports the value of such mechanisms and suggests that further advancement in this area has the potential to unlock additional capabilities within these models.

6 RELATED WORK

A large number of methods have been proposed that train PGMs and NNs jointly. For example, Zheng et al. (2015) proposed to combine conditional random fields (CRFs) and recurrent neural networks (RNNs), Arnab et al. (2016); Larsson et al. (2017, 2018); Schwing and Urtasun (2015) showed how to combine CNNs and CRFs, Chen et al. (2015) proposed to use densely connected graphical models with CNNs, and Johnson et al. (2016) combined latent graphical models with neural networks. The combination of PGMs and NNs has also been applied to improve performance on a wide variety of real-world tasks. Notable examples include human pose estimation (Liang et al., 2018; Song et al., 2017; Tompson et al., 2014; Yang et al., 2016), semantic labeling of body parts (Kirillov et al., 2016), stereo estimation (Knöbelreiter et al., 2017), language understanding (Yao et al., 2014), face sketch synthesis (Zhu et al., 2021) and crowd-sourcing aggregation (Li et al., 2021)). These hybrid models have also been used for solving a range of computer vision tasks such as semantic segmentation (Arnab et al., 2018; Guo and Dou, 2021), image crowd counting (Han et al., 2017), visual relationship detection (Yu et al., 2022), modeling for epileptic seizure detection (Craley et al., 2019), face sketch synthesis (Zhang et al., 2020), semantic image segmentation (Chen et al., 2018; Lin et al., 2016), 2D Hand-pose Estimation (Kong et al., 2019), depth estimation from a single monocular image (Liu et al., 2015), animal pose tracking (Wu et al., 2020) and pose estimation (Chen and Yuille, 2014).

To date, dependency networks have been used to solve various tasks such as collective classification (Neville and Jensen, 2003), binary classification (Gámez et al., 2006, 2008), multi-label classification (Guo and Gu, 2011), part-of-speech tagging (Tarantola and Blanc, 2002), relation prediction (Figueiredo et al., 2021), text classification (Guo and Weng, 2020) and collaborative filtering (Heckerman et al., 2000). However, DDNs have traditionally been restricted to Gibbs sampling (Heckerman et al., 2000) and mean-field inference (Lowd and Shamaei, 2011), showing limited compatibility with advanced probabilistic inference methods (Lowd, 2012). This study marks the inaugural attempt to incorporate advanced inference methods for the MPE task in DDNs, utilizing jointly trained networks for MLC scenarios.

7 CONCLUSION AND FUTURE WORK

More and more state-of-the-art methods for challenging applications of computer vision tasks usually use deep neural networks. Deep neural networks are good at extracting features in vision tasks like image classification, video classification, object detection, image segmentation, and others. Nevertheless, for more complex tasks involving multi-label classification, these methods cannot model crucial information like inter-label dependencies and infer about them. In this work, we present novel inference algorithms for Deep Dependency Networks (DDNs), a powerful neuro-symbolic approach, that exhibit consistent superiority compared to traditional neural network baselines, sometimes by a substantial margin. These algorithms offer an improvement over existing Gibbs Sampling-based inference schemes used for DDNs, without incurring significant computational burden. Importantly, they have the capacity to infer inter-label dependencies that are commonly overlooked by baseline techniques utilizing transformers, attention modules, and Graph Attention Networks (GAT). By formulating the inference procedure as an optimization

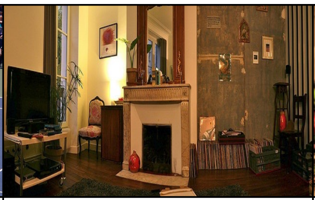
Image				
Ground Truth	person, sports ball, tennis racket, chair, clock	chair, potted plant, tv, remote, book, vase	person, bottle, pizza, clock	bird, skateboard, couch
Q2L	person (1.00), tennis racket (1.00), chair (0.96), clock (0.95), **[sports ball (0.33)]**	chair (0.99), potted plant (0.99), tv (1.00), book (1.00), vase (0.96), [**remote (0.25)**]	person (1.00), bottle (0.99), pizza (1.00), **potted plant (0.74)**, clock (0.77), **vase (0.80),**	bird (1.00), skateboard (1.00), couch (0.73)
DDN	person, sports ball, tennis racket, chair, clock	chair, potted plant, tv, remote, book, vase	person, bottle, pizza, clock	bird, skateboard

Figure 3: Comparison of labels predicted by Q2L (Liu et al., 2021) and our DDN-ILP scheme on the MS-COCO dataset. Labels in bold represent the difference between the predictions of the two methods, assuming that a threshold of 0.5 is used (i.e., every label whose probability > 0.5 is considered a predicted label). Due to the MPE focus in DDN-ILP, only label configurations are generated, omitting corresponding probabilities. The first three column shows examples where DDN improves over Q2L, while the last column (outlined in red) shows an example where DDN is worse than Q2L.

problem, our approach permits the integration of domain-specific constraints, resulting in a more knowledgeable and focused inference process. In particular, our optimization-based approach furnishes a robust and computationally efficient mechanism for inference, well-suited for handling intricate multi-label classification tasks.

Avenues for future work include: applying the setup described in the paper to other multi-label classification tasks in computer vision, natural language understanding, and speech recognition; converting DDNs to MRFs for better inference (Lowd, 2012); exploring and validating the neuro-symbolic benefits of DDNs such as improved accuracy in predictions and enhanced interpretability of model decisions; etc.

ACKNOWLEDGMENTS

This work was supported in part by the DARPA Perceptually-Enabled Task Guidance (PTG) Program under contract number HR00112220005, by the DARPA Assured Neuro Symbolic Learning and Reasoning (ANSR) under contract number HR001122S0039 and by the National Science Foundation grant IIS-1652835.

References

A. Antonucci, G. Corani, D. D. Mauá, and S. Gabaglio. An ensemble of bayesian networks for multilabel classification. In *Twenty-third international joint conference on artificial intelligence*, 2013.

A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr. Higher order conditional random fields in deep neural networks. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 524–540, Cham, 2016. Springer International Publishing.

A. Arnab, S. Zheng, S. Jayasumana, B. Romera-Paredes, M. Larsson, A. Kirillov, B. Savchynskyy, C. Rother, F. Kahl, and P. H. Torr. Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation: Combining Probabilistic Graphical Models with Deep Learning for Structured Prediction. *IEEE Signal Processing Magazine*, 35(1):37–52, Jan. 2018. ISSN 1558-0792. doi: 10.1109/MSP.2017.2762355.

M. Asghari, A. M. Fathollahi-Fard, S. M. J. Mirzapour Al-e-hashem, and M. A. Dulebenets. Transformation and Linearization Techniques in Optimization: A State-of-the-Art Survey. *Mathematics*, 10(2):283, Jan. 2022. ISSN 2227-7390. doi: 10.3390/math10020283.

J. Besag. Statistical Analysis of Non-Lattice Data. *The Statistician*, 24:179–195, 1975.

L.-C. Chen, A. Schwing, A. Yuille, and R. Urtasun. Learning deep structured models. In *International Conference on Machine Learning*, pages 1785–1794. PMLR, 2015.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, Apr. 2018. ISSN 0162-8828, 2160-9292. doi: 10.1109/TPAMI.2017.2699184.

T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin. Learning Semantic-Specific Graph Representation for Multi-Label Image Recognition, Aug. 2019a.

X. Chen and A. L. Yuille. Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Z.-M. Chen, X.-S. Wei, X. Jin, and Y. Guo. Multi-Label Image Recognition with Joint Class-Aware Map Disentangling and Label Correlation Embedding. *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 622–627, July 2019b. doi: 10.1109/ICME.2019.00113.

T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. NUS-WIDE: A real-world web image database from National University of Singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 1–9, Santorini, Fira Greece, July 2009. ACM. ISBN 978-1-60558-480-5. doi: 10.1145/1646396.1646452.

J. Craley, E. Johnson, and A. Venkataraman. Integrating Convolutional Neural Networks and Probabilistic Graphical Modeling for Epileptic Seizure Detection in Multichannel EEG. In A. C. S. Chung, J. C. Gee, P. A. Yushkevich, and S. Bao, editors, *Information Processing in Medical Imaging*, volume 11492, pages 291–303. Springer International Publishing, Cham, 2019. ISBN 978-3-030-20350-4 978-3-030-20351-1. doi: 10.1007/978-3-030-20351-1_22. Series Title: Lecture Notes in Computer Science.

N. Di Mauro, A. Vergari, and F. Esposito. Multi-label classification with cutset networks. In A. Antonucci, G. Corani, and C. P. Campos, editors, *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, volume 52 of *Proceedings of Machine Learning Research*, pages 147–158, Lugano, Switzerland, 06–09 Sep 2016. PMLR.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-009-0275-4.

H. Fan, Y. Li, B. Xiong, W.-Y. Lo, and C. Feichtenhofer. Pyslowfast. `https://github.com/facebookresearch/slowfast`, 2020.

C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, October 2019.

L. F. d. Figueiredo, A. Paes, and G. Zaverucha. Transfer learning for boosted relational dependency networks through genetic algorithm. In *International Conference on Inductive Logic Programming*, pages 125–139. Springer, 2021.

J. A. Gámez, J. L. Mateo, and J. M. Puerta. Dependency networks based classifiers: learning models by using independence test. In *Third European Workshop on Probabilistica Graphical Models (PGM06)*, pages 115–122. Citeseer, 2006.

J. A. Gámez, J. L. Mateo, T. D. Nielsen, and J. M. Puerta. Robust classification using mixtures of dependency networks. In *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models*, pages 129–136, 2008.

B. Geißler, A. Martin, A. Morsi, and L. Schewe. Using Piecewise Linear Functions for Solving MINLPs. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, The IMA Volumes in Mathematics and Its Applications, pages 287–314, New York, NY, 2012. Springer. ISBN 978-1-4614-1927-3. doi: 10.1007/978-1-4614-1927-3_10.

I. Griva, S. G. Nash, and A. Sofer. *Linear and Nonlinear Optimization*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2. ed edition, 2009. ISBN 978-0-89871-661-0.

Q. Guo and Q. Dou. Semantic Image Segmentation based on SegNetWithCRFs. *Procedia Computer Science*, 187: 300–306, 2021. ISSN 18770509. doi: 10.1016/j.procs.2021.04.066.

X. Guo and Y. Weng. Deep Dependency Network for Multi-label Text Classification. In Y. Peng, Q. Liu, H. Lu, Z. Sun, C. Liu, X. Chen, H. Zha, and J. Yang, editors, *Pattern Recognition and Computer Vision*, Lecture Notes in Computer Science, pages 298–309, Cham, 2020. Springer International Publishing. ISBN 978-3-030-60636-7. doi: 10.1007/978-3-030-60636-7_25.

Y. Guo and S. Gu. Multi-label classification using conditional dependency networks. In *Twenty-Second International Joint Conference on Artificial Intelligence*, pages 1300–1305, 2011.

Y. Guo and W. Xue. Probabilistic multi-label classification with sparse feature learning. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 1373–1379. AAAI Press, 2013. ISBN 9781577356332.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL `https://www.gurobi.com`.

K. Han, W. Wan, H. Yao, L. Hou, and School of Communication and Information Engineering, Shanghai University Institute of Smart City, Shanghai University 99 Shangda Road, BaoShan District, Shanghai 200444, China. Image Crowd Counting Using Convolutional Neural Network and Markov Random Field. *Journal of*

Advanced Computational Intelligence and Intelligent Informatics, 21(4):632–638, July 2017. ISSN 1883-8014, 1343-0130. doi: 10.20965/jaciii.2017.p0632.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1(Oct):49–75, 2000.

M. J. Johnson, D. Duvenaud, A. B. Wiltschko, S. R. Datta, and R. P. Adams. Composing graphical models with neural networks for structured representations and fast inference. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2954–2962, 2016.

A. Kirillov, D. Schlesinger, S. Zheng, B. Savchynskyy, P. H. S. Torr, and C. Rother. Joint Training of Generic CNN-CRF Models with Stochastic Optimization, 2016.

P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock. End-to-end training of hybrid cnn-crf models for stereo. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1456–1465, 2017. doi: 10.1109/CVPR.2017.159.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. MIT Press, 2009. ISBN 9780262013192.

D. Kong, Y. Chen, H. Ma, X. Yan, and X. Xie. Adaptive graphical model network for 2d handpose estimation. In *BMVC*, 2019.

X. Kong, B. Cao, and P. S. Yu. Multi-label classification by mining label and instance correlations from heterogeneous information networks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 614–622, Chicago Illinois USA, Aug. 2013. ACM. ISBN 978-1-4503-2174-7. doi: 10.1145/2487575.2487577.

R. G. Krishnan, U. Shalit, and D. Sontag. Deep kalman filters. *stat*, 1050:25, 2015.

M. Kumar. Converting some global optimization problems to mixed integer linear problems using piecewise linear approximations. Master's thesis, University of Missouri–Rolla, 2007.

M. Larsson, J. Alvén, and F. Kahl. Max-Margin Learning of Deep Structured Models for Semantic Segmentation. In P. Sharma and F. M. Bianchi, editors, *Image Analysis*, Lecture Notes in Computer Science, pages 28–40, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59129-2. doi: 10.1007/978-3-319-59129-2_3.

M. Larsson, A. Arnab, F. Kahl, S. Zheng, and P. Torr. A Projected Gradient Descent Method for CRF Inference allowing End-To-End Training of Arbitrary Pairwise Potentials. Technical Report arXiv:1701.06805, arXiv, Jan. 2018. arXiv:1701.06805 [cs] type: article.

S.-i. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of markov networks using l_1-regularization. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

S.-Y. Li, S.-J. Huang, and S. Chen. Crowdsourcing aggregation with deep Bayesian learning. *Science China Information Sciences*, 64(3):130104, Mar. 2021. ISSN 1674-733X, 1869-1919. doi: 10.1007/s11432-020-3118-7.

Z. Li, Y. Zhang, B. Sui, Z. Xing, and Q. Wang. FPGA Implementation for the Sigmoid with Piecewise Linear Fitting Method Based on Curvature Analysis. *Electronics*, 11(9):1365, Jan. 2022. ISSN 2079-9292. doi: 10.3390/electronics11091365.

G. Liang, X. Lan, J. Wang, J. Wang, and N. Zheng. A Limb-Based Graphical Model for Human Pose Estimation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(7):1080–1092, July 2018. ISSN 2168-2232. doi: 10.1109/TSMC.2016.2639788. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics: Systems.

G. Lin, C. Shen, A. Van Den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2016.

M.-H. Lin, J. G. Carlsson, D. Ge, J. Shi, and J.-F. Tsai. A Review of Piecewise Linearization Methods. *Mathematical Problems in Engineering*, 2013:e101376, Nov. 2013. ISSN 1024-123X. doi: 10.1155/2013/101376.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

B. Liu, X. Liu, H. Ren, J. Qian, and Y. Wang. Text multi-label learning method based on label-aware attention and semantic dependency. *Multimedia Tools and Applications*, 81(5):7219–7237, Feb. 2022. ISSN 1573-7721. doi: 10.1007/s11042-021-11663-9.

F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5162–5170, 2015.

J. Liu. *Monte Carlo strategies in scientific computing*. Springer Verlag, New York, Berlin, Heidelberg, 2008. ISBN 0-387-95230-6.

S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu. Query2Label: A Simple Transformer Way to Multi-Label Classification, July 2021.

D. Lowd. Closed-form learning of markov networks from dependency networks. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 533–542, 2012.

D. Lowd and A. Shamaei. Mean Field Inference in Dependency Networks: An Empirical Study. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25(1): 404–410, Aug. 2011. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v25i1.7936.

R. Mateescu, K. Kask, V. Gogate, and R. Dechter. Join-graph propagation algorithms. *Journal of Artificial Intelligence Research*, 37:279–328, 2010.

I. Naim, Y. Song, Q. Liu, H. Kautz, J. Luo, and D. Gildea. Unsupervised alignment of natural language instructions with video segments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), Jun. 2014. doi: 10.1609/aaai.v28i1.8939.

I. Naim, Y. C. Song, Q. Liu, L. Huang, H. Kautz, J. Luo, and D. Gildea. Discriminative unsupervised alignment of natural language instructions with corresponding video segments. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 164–174, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1017.

J. Neville and D. Jensen. Collective classification with relational dependency networks. In *Workshop on Multi-Relational Data Mining (MRDM-2003)*, page 77, 2003.

H. D. Nguyen, X.-S. Vu, and D.-T. Le. Modular Graph Transformer Networks for Multi-Label Image Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):9092–9100, May 2021. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v35i10.17098.

C. Papagiannopoulou, G. Tsoumakas, and I. Tsamardinos. Discovering and Exploiting Deterministic Label Relationships in Multi-Label Learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 915–924, Sydney NSW Australia, Aug. 2015. ACM. ISBN 978-1-4503-3664-2. doi: 10.1145/2783258.2783302.

X. Qu, H. Che, J. Huang, L. Xu, and X. Zheng. Multi-layered Semantic Representation Network for Multi-label Image Classification, June 2021.

M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics (TACL)*, 1:25–36, 2013.

R. Rovatti, C. D'Ambrosio, A. Lodi, and S. Martello. Optimistic MILP modeling of non-linear optimization problems. *European Journal of Operational Research*, 239 (1):32–45, 2014.

A. G. Schwing and R. Urtasun. Fully Connected Deep Structured Networks. Technical report, arXiv, Mar. 2015.

B. Selman, H. A. Kautz, and B. Cohen. Local search strategies for satisfiability testing. In *Cliques, Coloring, and Satisfiability*, 1993. URL https://api.semanticscholar.org/CorpusID:3215289.

G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016.

J. Song, L. Wang, L. V. Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5563–5572, 2017.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.308.

M. Tan, Q. Shi, A. van den Hengel, C. Shen, J. Gao, F. Hu, and Z. Zhang. Learning graph structure for multi-label image classification via clique generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4100–4109, June 2015.

C. Tarantola and E. Blanc. Dependency networks and bayesian networks for web mining. *WIT Transactions on Information and Communication Technologies*, 28, 2002.

J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *CoRR*, abs/1406.2984, 2014.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

M. J. Wainwright, J. Lafferty, and P. Ravikumar. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

H. Wang, M. Huang, and X. Zhu. A generative probabilistic model for multi-label classification. In *2008 Eighth IEEE International Conference on Data Mining*, pages 628–637. IEEE, 2008.

L. Wang, Y. Liu, H. Di, C. Qin, G. Sun, and Y. Fu. Semi-supervised dual relation learning for multi-label classification. *IEEE Transactions on Image Processing*, 30: 9125–9135, 2021a.

R. Wang, R. Ridley, X. Su, W. Qu, and X. Dai. A novel reasoning mechanism for multi-label text classification. *Information Processing & Management*, 58(2):102441, Mar. 2021b. ISSN 03064573. doi: 10.1016/j.ipm.2020.102441.

S. Wang, J. Wang, Z. Wang, and Q. Ji. Enhancing multi-label classification by modeling dependencies among labels. *Pattern Recognition*, 47(10):3405–3413, Oct. 2014. ISSN 00313203. doi: 10.1016/j.patcog.2014.04.009.

W. Weng, B. Wei, W. Ke, Y. Fan, J. Wang, and Y. Li. Learning label-specific features with global and local label correlation for multi-label classification. *Applied Intelligence*, 53(3):3017–3033, Feb. 2023. ISSN 1573-7497. doi: 10.1007/s10489-022-03386-7.

A. Wu, E. K. Buchanan, M. Whiteway, M. Schartner, G. Meijer, J.-P. Noel, E. Rodriguez, C. Everett, A. Norovich, E. Schaffer, N. Mishra, C. D. Salzman, D. Angelaki, A. Bendesky, T. I. B. L. The International Brain Laboratory, J. P. Cunningham, and L. Paninski. Deep graph pose: a semi-supervised deep graphical model for improved animal pose tracking. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6040–6052. Curran Associates, Inc., 2020.

W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3073–3082, June 2016. doi: 10.1109/CVPR.2016.335. ISSN: 1063-6919.

K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao. Recurrent conditional random field for language understanding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4077–4081, Florence, Italy, May 2014. IEEE. ISBN 978-1-4799-2893-4. doi: 10.1109/ICASSP.2014.6854368.

J. S. Yedidia, W. Freeman, and Y. Weiss. Generalized Belief Propagation. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000.

D. Yu, B. Yang, Q. Wei, A. Li, and S. Pan. A probabilistic graphical model based on neural-symbolic reasoning for visual relationship detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10609–10618, 2022.

M. Zhang, N. Wang, Y. Li, and X. Gao. Neural Probabilistic Graphical Model for Face Sketch Synthesis. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2623–2637, July 2020. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2019.2933590.

S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537, Santiago, Chile, Dec. 2015. IEEE. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.179.

W. Zhou, Z. Xia, P. Dou, T. Su, and H. Hu. Double Attention Based on Graph Attention Network for Image Multi-Label Classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 19(1):1–23, Jan. 2023. ISSN 1551-6857, 1551-6865. doi: 10.1145/3519030.

M. Zhu, J. Li, N. Wang, and X. Gao. Learning Deep Patch representation for Probabilistic Graphical Model-Based Face Sketch Synthesis. *International Journal of Computer Vision*, 129(6):1820–1836, June 2021. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-021-01442-2.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A DERIVING THE OBJECTIVE FUNCTION FOR MOST PROBABLE EXPLANATIONS (MPE)

Let us consider the scoring function we aim to maximize, given as follows:

$$\underset{\mathbf{x}}{\text{maximize}} \sum_{i=1}^{n} \log\left(P_i(x_i \mid \mathbf{x}_{-i}, \mathbf{e})\right) \tag{10}$$

Given that x_i is binary, namely, $x_i \in \{0, 1\}$, we can express the scoring function as follows:

$$\underset{\mathbf{x}}{\text{maximize}} \sum_{i=1}^{n} \left(x_i \log\left(P_i(X_i = 1 \mid \mathbf{x}_{-i}, \mathbf{e})\right) + \ (1 - x_i) \log\left(1 - P_i(X_i = 1 \mid \mathbf{x}_{-i}, \mathbf{e})\right) \right) \tag{11}$$

where

$$P_i(X_i = 1 | \mathbf{x}_{-i}, \mathbf{e}) = \sigma\left(\sum_{j=1}^{|\mathbf{e}|} w_{ij} e_j + \sum_{\substack{k=1 \\ k \neq i}}^{|\mathbf{x}|} v_{ik} x_k + b_i \right)$$

Let $p_i = P_i(X_i = 1 | \mathbf{x}_{-i}, \mathbf{e})$ and $z_i = \sum_{j=1}^{|\mathbf{e}|} w_{ij} e_j + \sum_{\substack{k=1 \\ k \neq i}}^{|\mathbf{x}|} v_{ik} x_k + b_i$. Then, the MPE task, which involves optimizing the scoring function given above, can be expressed as:

$$\underset{\mathbf{x},\mathbf{z}}{\text{maximize}} \sum_{i=1}^{n} \left(x_i \log p_i + (1 - x_i) \log(1 - p_i) \right) \tag{12}$$

subject to:

$$p_i = \sigma(z_i), \quad \forall i \in \{1, \ldots, n\} \tag{13}$$

$$z_i = \sum_{j=1}^{|\mathbf{e}|} w_{ij} e_j + \sum_{\substack{k=1 \\ k \neq i}}^{|\mathbf{x}|} v_{ik} x_k + b_i, \quad \forall i \in \{1, \ldots, n\} \tag{14}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\} \tag{15}$$

The first constraint in the formulation, as expressed by 13, pertains to the sigmoid function employed in the logistic regression module. The subsequent constraint, defined by 14, formalizes the product between the weights and inputs in the Dependency Network (DN). Here, e_i represents evidence values supplied to the DN, while $\mathbf{x}$ represents the decision variables that are subject to optimization. Finally, 15 specifies that the input variables must be constrained to integer values of 0 or 1.

The assembled constraints collectively simulate a forward pass through the network. The objective function is designed to maximize the scoring function pertinent to the Most Probable Explanation (MPE). This framework adeptly aligns the optimization task to maximize the MPE score.

The substitution of the constraint $P_i(x_i | \mathbf{x}_{-i}, \mathbf{e}) = \sigma(z_i) = \frac{1}{1+e^{(-z_i)}}$ into the objective function and the subsequent algebraic simplification result in a simplified formulation as follows:

$$\begin{aligned}
& x_i \log p_i + (1 - x_i) \log(1 - p_i) \\
&= x_i \log\left(\frac{e^{z_i}}{1 + e^{z_i}}\right) + (1 - x_i) \log\left(1 - \frac{e^{z_i}}{1 + e^{z_i}}\right) \\
&= x_i \log\left(\frac{e^{z_i}}{1 + e^{z_i}}\right) + (1 - x_i) \log\left(\frac{1}{1 + e^{z_i}}\right) \\
&= x_i \log e^{z_i} - x_i \log(1 + e^{z_i}) - \log(1 + e^{z_i}) + x_i \log(1 + e^{z_i}) \\
&= x_i \log e^{z_i} - \log(1 + e^{z_i}) \\
&= x_i z_i - \log(1 + e^{z_i})
\end{aligned} \tag{16}$$

Substituting equation 16 in the objective function of our optimization problem, we get

$$\underset{\mathbf{x},\mathbf{z}}{\text{maximize}} \sum_{i=1}^{n} x_i z_i - \log\left(1 + e^{z_i}\right) \tag{17}$$

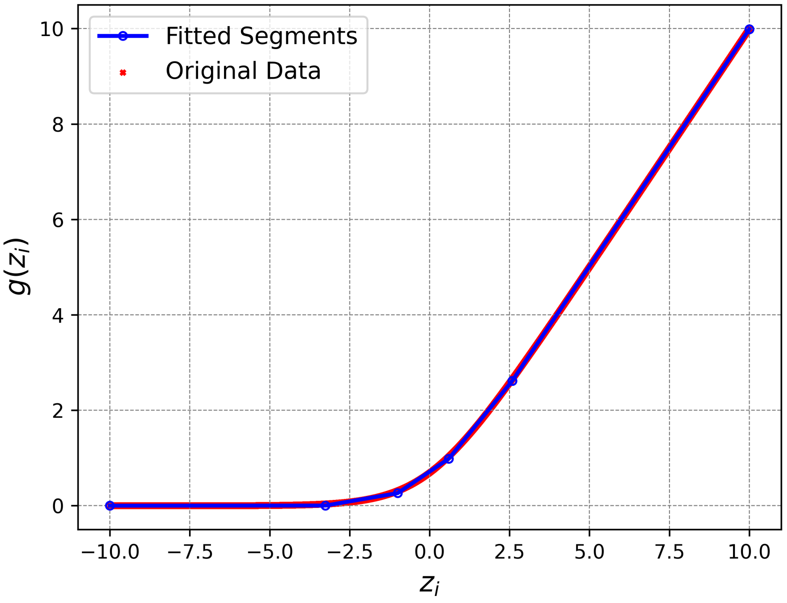
subject to:

$$z_i = \sum_{j=1}^{|\mathbf{e}|} w_{ij} e_j + \sum_{\substack{k=1 \\ k \neq i}}^{|\mathbf{x}|} v_{ik} x_k + b_i, \quad \forall i \in \{1, \ldots, n\} \tag{18}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\} \tag{19}$$

A.A Utilizing Piecewise Linear Approximation for Non-linear Functions

Figure SF4: Piece-wise linear approximation of $log(1 + e^{z_i})$



The objective function expressed in equation equation 17 is non-linear due to the inclusion of the term $\log(1 + e^{z_i})$. $f(z)$ can be used as a piecewise approximation for $\log(1 + e^{z_i})$.

$$f(z) = \begin{cases} z & z \gg 1 \\ e^z & z \ll 1 \\ \log(1 + e^z) & \text{otherwise} \end{cases} \tag{20}$$

To obtain a piecewise linear approximation of $\log(1 + e^{z_i})$, a single linear function suffices for $z \gg 1$, while the majority of the linear pieces are utilized for approximating the function near 1. A piecewise linear approximation of the function is detailed in the figure SF4. We employ five segments for the approximation. These piecewise functions can be integrated as linear constraints, facilitating the conversion of the non-linear objective into a linear objective with ease. We also present the piecewise equations for the approximation as follows -

Table ST3: Piecewise Linear Approximation for $g(z) \approx \log(1 + e^{z_i})$

z	$g(z)$
$]-\infty, -3.257)$	0
$[-3.257, -0.998)$	$(2^{-4} + 2^{-5} + 2^{-6} + 2^{-8})z + 0.379$
$[-0.998, 0.602)$	$(2^{-2} + 2^{-3} + 2^{-4} + 2^{-7} + 2^{-8})z + 0.715$
$[0.602, 2.584)$	$(2^{-1} + 2^{-2} + 2^{-4} + 2^{-7})z + 0.492$
$[2.584, +\infty[$	z

Thus after replacing $\log(1 + e^{z_i})$ with its piece-wise approximation ($g(z)$) we get

$$\begin{aligned}
&\underset{\mathbf{x},\mathbf{z},\alpha}{\text{maximize}} \sum_{i=1}^{n} x_i z_i - g(z_i) \\
&\text{subject to:} \\
&z_i = \sum_{j=1}^{|\mathbf{e}|} w_{ij} e_j + \sum_{\substack{k=1\\k\neq i}}^{|\mathbf{x}|} v_{ik} x_k + b_i, \quad \forall i \in \{1, \dots, n\} \\
&x_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, n\} \\
&g(z_i) = \alpha_{1i} \times 0 + \alpha_{2i} \times ((2^{-4} + 2^{-5} + 2^{-6} + 2^{-8})z_i + 0.379) + \\
&\qquad\qquad \alpha_{3i} \times ((2^{-2} + 2^{-3} + 2^{-4} + 2^{-7} + 2^{-8})z_i + 0.715) + \\
&\qquad\qquad \alpha_{4i} \times ((2^{-1} + 2^{-2} + 2^{-4} + 2^{-7})z_i + 0.492) + \\
&\qquad\qquad \alpha_{5i} z_i, \quad \forall i \in \{1, \dots, n\} \\
&\sum_{j=1}^{5} \alpha_{ji} = 1, \quad \forall i \in \{1, \dots, n\} \\
&\alpha_{ji} \in \{0, 1\}, \quad \forall j \in \{1, 2, 3, 4, 5\}, \forall i \in \{1, \dots, n\} \\
&\alpha_{1i} = 1 \Rightarrow z_i < -3.257, \quad \forall i \in \{1, \dots, n\} \\
&\alpha_{2i} = 1 \Rightarrow -3.257 \leq z_i < -0.998, \quad \forall i \in \{1, \dots, n\} \\
&\alpha_{3i} = 1 \Rightarrow -0.998 \leq z_i < 0.602, \quad \forall i \in \{1, \dots, n\} \\
&\alpha_{4i} = 1 \Rightarrow 0.602 \leq z_i < 2.584, \quad \forall i \in \{1, \dots, n\} \\
&\alpha_{5i} = 1 \Rightarrow z_i \geq 2.584, \quad \forall i \in \{1, \dots, n\}
\end{aligned} \tag{21}$$

The utilization of binary variables, α_{ji}, allows for a piece-wise linear approximation of the logarithm of $(1+e^{z_i})$, with these variables serving as selectors to determine the appropriate linear segment based on the value of z_i. It is worth noting that the final five constraints are indicator constraints, which can be linearized using the big-M method as described in Griva et al. (2009). Consequently, the problem can be formulated as a mixed-integer multi-linear optimization problem. The mixed-integer multi-linear problem can be further converted to a mixed-integer linear problem (MILP) using the approach describe next.

B FORMULATING LOGICAL AND BETWEEN BINARY VARIABLES AS LINEAR CONSTRAINTS

Consider an optimization problem involving three binary variables X_1, X_2, and X_3, where x_1, x_2, and x_3 denote their respective assignments. We define an objective function that incorporates products of these binary variables as follows -

$$\max_{x_1, x_2, x_3} \; x_1 x_2 - x_2 x_3 + x_1 x_3 \tag{22}$$

Although constraints are not incorporated in this instance, the same methodology can be extended to constrained optimization problems. Subsequently, auxiliary variables z_1, z_2, and z_3 are introduced to account for each of the binary products.

The optimization problem can be stated as follows:

$$\begin{aligned} \max_{x_1,x_2,x_3} \quad & z_1 - z_2 + z_3 \\ \text{s.t.} \quad & x_1 \wedge x_2 = z_1, \\ & x_2 \wedge x_3 = z_2, \\ & x_1 \wedge x_3 = z_3. \end{aligned} \tag{23}$$

The optimization problem may be expressed by incorporating additional linear constraints to represent each boolean product. The resulting problem is presented as follows:

$$\begin{aligned} \max_{x_1,x_2,x_3} \quad & z_1 - z_2 + z_3 \\ \text{s.t.} \quad & x_1 + x_2 - 1 \leq z_1, \\ & z_1 \leq x_1, \\ & z_1 \leq x_2, \\ & x_2 + x_3 - 1 \leq z_2, \\ & z_2 \leq x_2, \\ & z_2 \leq x_3, \\ & x_1 + x_3 - 1 \leq z_3, \\ & z_3 \leq x_1, \\ & z_3 \leq x_3. \end{aligned} \tag{24}$$

The problem formulation outlined in Section A, and more specifically, equation equation 21, includes expressions of the form $x_i x_j$. These terms serve to represent the logical AND operation between binary variables x_i and x_j, and are a direct result of the product $x_i \times g_i$. Consequently, this optimization problem is classified as a mixed-integer multi-linear optimization problem.

The formulation detailed in this section enables the capture of these logical AND operations between binary variables through linear constraints, thereby enabling the utilization of Mixed-Integer Linear Programming (MILP) solvers, e.g., Gurobi Optimization, LLC (2023), to find the optimal solution (or an anytime, near optimal solution if a time bound is specified).

C GIBBS SAMPLING FOR DDNS

Algorithm 1 Gibbs Sampling for DDNs

```
Input: video segment/image $\mathbf{v}$, number of samples $N$, DDN $\langle \mathcal{N}, \mathcal{D} \rangle$
Output: An estimate of the marginal probability distribution over each label $X_i$ of the DDN given $\mathbf{v}$
$\mathbf{e} \leftarrow \mathbb{N}(\mathbf{v})$
Randomly initialize $\mathbf{X} = \mathbf{x}^{(0)}$
for $j = 1$ to $N$ do
    $\pi \leftarrow$ Generate random permutation of $[1, n]$
    for $i = 1$ to $n$ do
        $x^{(j)}_{\pi(i)} \sim P_{\pi(i)}(x_{\pi(i)} | \mathbf{x}^{(j)}_{\pi(1):\pi(i-1)}, \mathbf{x}^{(j-1)}_{\pi(i+1):\pi(n)}, \mathbf{e})$
for $i = 1$ to $n$ do
    $\hat{P}_i(x_i|\mathbf{v}) = \frac{1}{N} \sum_{j=1}^{N} P_i(x_i | \mathbf{x}^{(j)}_{-i}, \mathbf{e})$
return $\{\hat{P}_i(x_i|\mathbf{v}) | i \in \{1, \ldots, n\}\}$
```

This section describes the Gibbs Sampling inference procedure for DDNs (see Algorithm 1). Gibbs Sampling serves as an approximation method for the Most Probable Explanation (MPE) inference task in DDNs by offering max-marginals, which can subsequently be utilized to approximate MPE. The inputs to the algorithm are (1) a video segment/image $\mathbf{v}$, (2) the number of samples N and (3) trained DDN model $\langle \mathcal{N}, \mathcal{D} \rangle$. The algorithm begins (see step 1) by extracting features $\mathbf{e}$ from the video segment/image $\mathbf{v}$ by sending the latter through the neural network $\mathcal{N}$ (which represents the function $\mathbb{N}$). Then in steps 2–8, it generates N samples via Gibbs sampling. The Gibbs sampling procedure begins with a random assignment to all the labels (step 2). Then at each iteration (steps 3–8), it first generates a random permutation π over the n labels and samples the labels one by one along the order π (steps 5–7). To sample a label indexed by $\pi(i)$ at iteration j, we compute $P_{\pi(i)}(x_{\pi(i)}|\mathbf{x}^{(j)}_{\pi(1):\pi(i-1)}, \mathbf{x}^{(j-1)}_{\pi(i+1):\pi(n)}, \mathbf{e})$ from the DN $\mathcal{D}$ where $\mathbf{x}^{(j)}_{\pi(1):\pi(i-1)}$ and $\mathbf{x}^{(j-1)}_{\pi(i+1):\pi(n)}$ denote the assignments to all labels ordered before $x_{\pi(i)}$ at iteration j and the assignments to all labels ordered after $x_{\pi(i)}$ at iteration $j-1$ respectively.

After N samples are generated via Gibbs sampling, the algorithm uses them to estimate (see steps 9–11) the (posterior) marginal probability distribution at each label X_i given $\mathbf{v}$ using the mixture estimator (Liu, 2008). The algorithm terminates (see step 12) by returning these posterior estimates.

D PRECISION BASED EVALUATION METRICS

In Tables ST4 and ST5, we present a comprehensive account of precision-focused metrics, specifically Mean Average Precision (mAP) and Label Ranking Average Precision (LRAP). It should be noted that these metrics are not directly applicable to the Most Probable Explanation (MPE) task, as they necessitate scores (probabilities) of the output labels. However, given that SlowFast, InceptionV3, Q2L, MSRN, and DDN-GS can provide such probabilities, their precision scores are anticipated to be elevated. We employed Gibbs Sampling to approximate the MPE, enabling the derivation of marginal probabilities to approximate max-marginals and, thereby, the MPE.

In majority of the instances, the Gibbs Sampling-based inference on DDNs yields performance metrics closely aligned with baseline methods. Variability exists, with some metrics exceeding the baseline in the context of MLAC and falling short in the case of MLIC. Notably, the Integer Linear Programming (ILP)-based approach for DDN outperforms all alternative methods across three datasets when evaluated on LRAP. DRF-based methods and DDN-specific inference techniques generally underperform on these precision metrics. Again, this outcome is anticipated, given that apart from Gibbs Sampling, none of the other techniques can generate probabilistic scores for labels, leading to their inferior performance.

Table ST4: Comparison of our methods with the feature extractor for MLAC - Precision-based metrics. The best/second best values are bold/underlined.

Method	Charades		TACoS		Wetlab	
	mAP	LRAP	mAP	LRAP	mAP	LRAP
SlowFast (Fan et al., 2020)	0.39	0.53				
InceptionV3 (Szegedy et al., 2016)			0.70	0.81	0.79	0.82
DRF - GS	0.27	0.44	0.56	0.79	0.54	0.76
DRF - ILP	0.19	0.28	0.40	0.67	0.63	0.73
DRF - IJGP	0.31	0.44	0.56	0.81	0.79	0.85
DDN - GS	**0.40**	**0.55**	**0.75**	0.84	**0.84**	**0.87**
DDN - RW	0.19	0.28	0.49	0.66	0.63	0.77
DDN - Greedy	0.32	0.30	0.50	0.69	0.65	0.78
DDN - ILP	0.36	0.53	0.51	**0.85**	0.65	**0.87**

E ANNOTATIONS COMPARISON BETWEEN Q2L AND DDN-MLP-JOINT ON THE MS-COCO DATASET

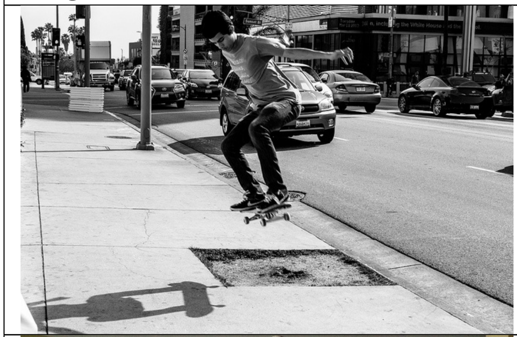
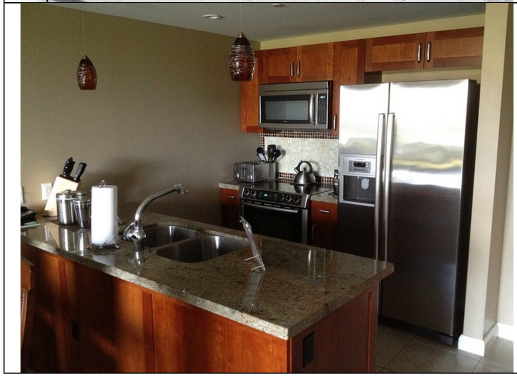
Table ST6 presents a qualitative evaluation of the label predictions produced by our DDN-ILP inference scheme compared to the baseline Q2L using randomly selected images from the MS-COCO dataset. This analysis provides a perspective on the advantages of DDN-ILP over Q2L, particularly in terms of correcting errors generated by the feature extractor. In the initial set of seven rows, DDN-ILP adds additional correct labels to the Q2L output. For certain images, the objects overlooked by Q2L are inherently challenging to identify. In these instances, the DDN-ILP method, which infers labels based on label relationships, effectively rectifies the limitations of Q2L. DDN-ILP refines Q2L's predictions in the following seven instances by removing incorrect predictions and aligning precisely with the ground truth labels. Finally, we

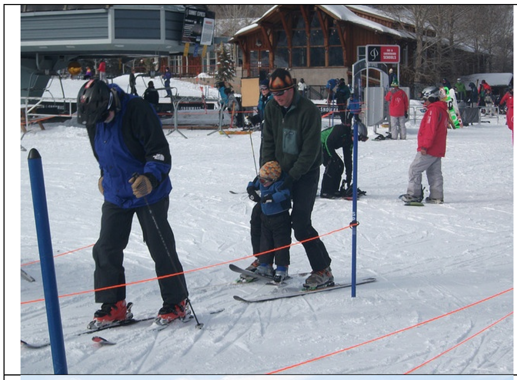
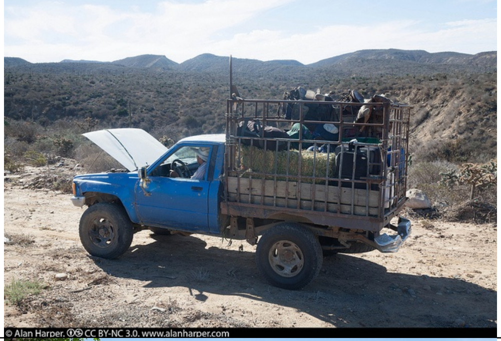
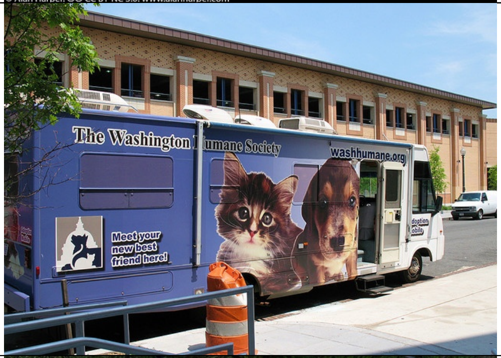
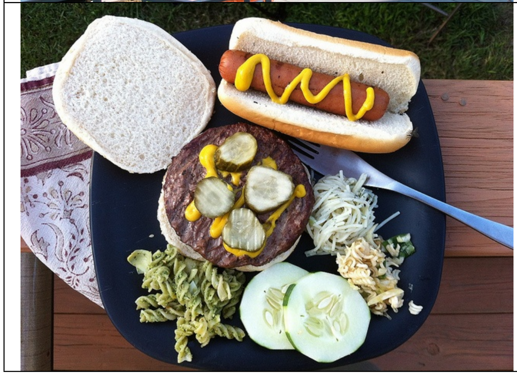
Table ST5: Comparison of our methods with the feature extractor for MLIC for Precision based metrics. The best/second best values are bold/underlined.

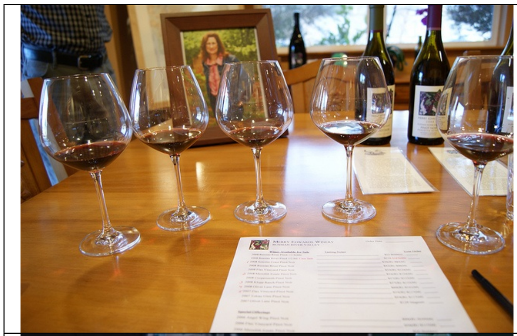
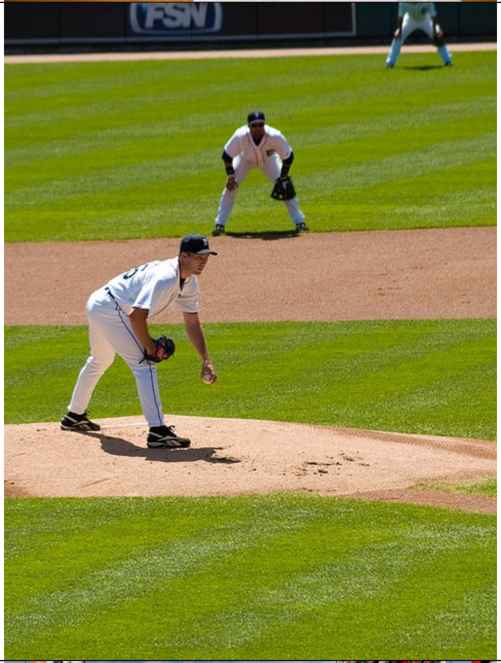
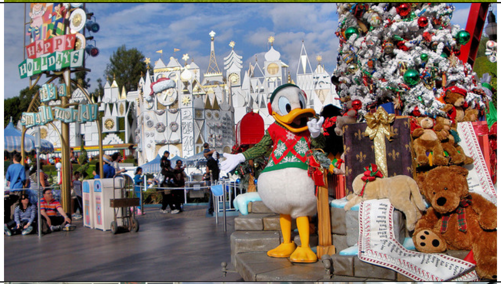
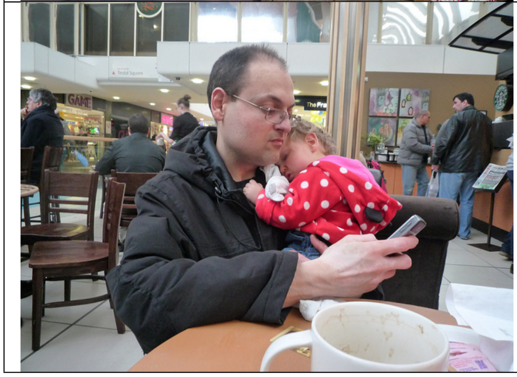
Method	MS-COCO		NUS-WIDE		PASCAL-VOC	
	mAP	LRAP	mAP	LRAP	mAP	LRAP
Q2L (Liu et al., 2021)	**0.91**	**0.96**				
MSRN (Qu et al., 2021)			**0.62**	**0.85**	0.96	**0.98**
DRF - GS	0.75	0.86	0.40	0.74	0.77	0.93
DRF - ILP	0.74	0.82	0.25	0.59	0.81	0.88
DRF - IJGP	0.74	0.90	0.41	0.75	0.83	0.94
DDN - GS	0.84	0.93	0.50	0.82	0.92	0.96
DDN - RW	0.74	0.82	0.24	0.61	0.91	0.94
DDN - Greedy	0.74	0.83	0.35	0.68	0.93	0.95
DDN - ILP	0.85	0.83	0.48	0.82	**0.97**	**0.98**

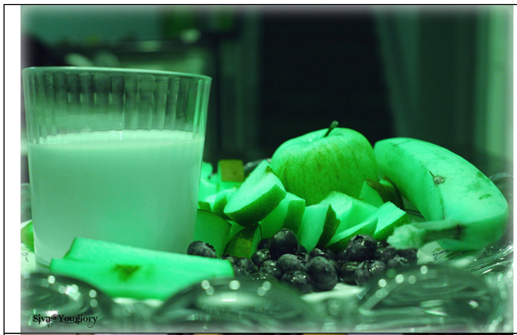
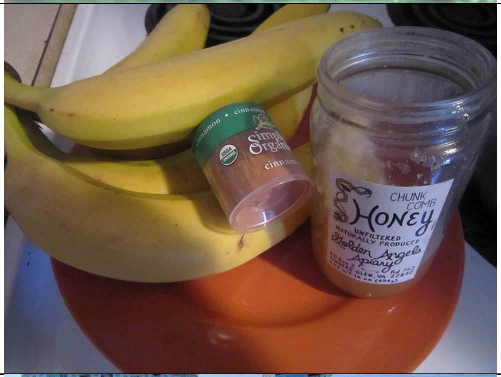
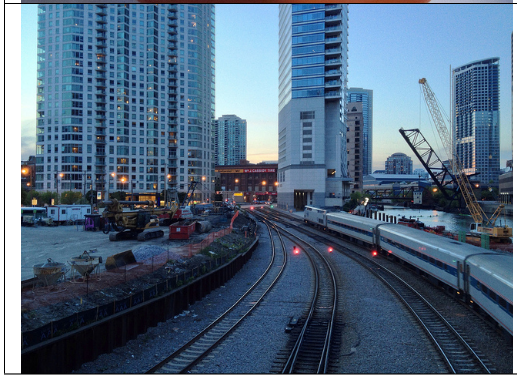
examine cases where DDN-ILP's modifications result in inaccuracies (rows have grey background), contrasting with Q2L's correct predictions. These results demonstrate the effectiveness of DDN-ILP in improving the predicted labels relative to the baseline.

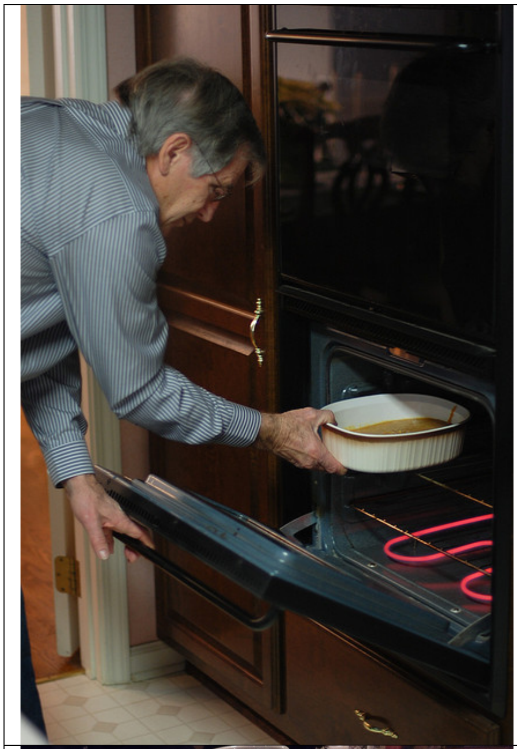
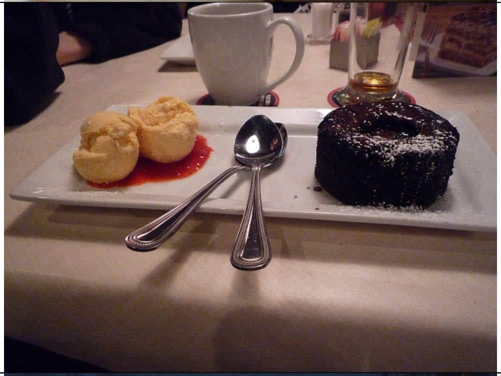
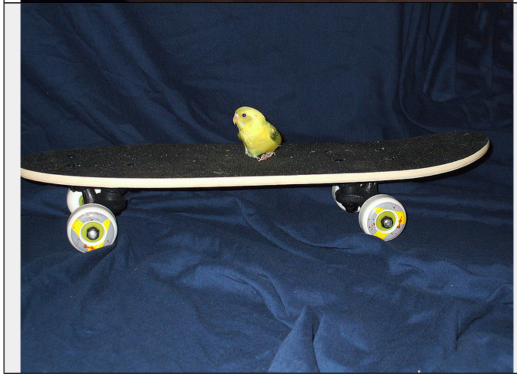
Table ST6: Comparison of labels predicted by Q2L (Liu et al., 2021) and our DDN-ILP scheme on the MS-COCO dataset. Labels inside **[]** represent the difference between the predictions of the two methods, assuming that a threshold of 0.5 is used (i.e., every label whose probability > 0.5 is considered a predicted label). Due to the MPE focus in DDN-ILP, only label configurations are generated, omitting corresponding probabilities.

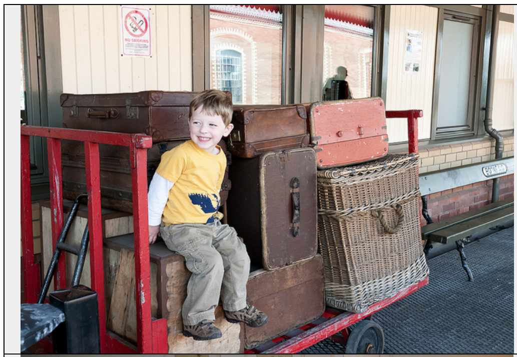
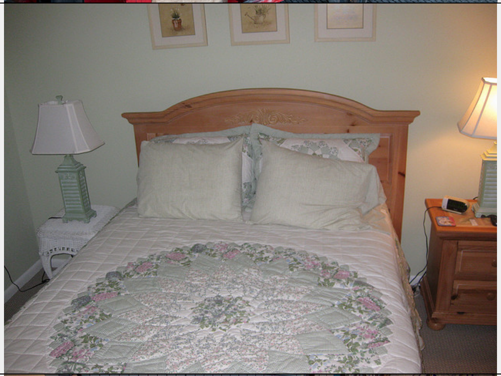
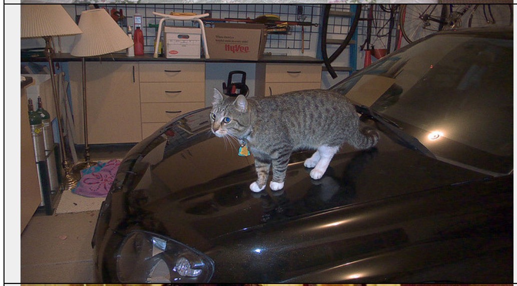
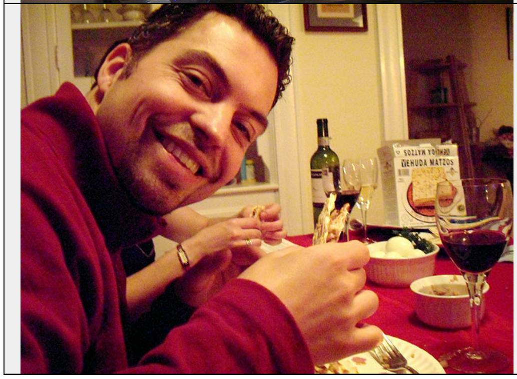
Image	**Ground Truth**	**Q2L**	**DDN**
	person, car, truck, traffic light, skateboard	person (1.00), car (1.00), skateboard (1.00), **[truck (0.33), traffic light (0.41)]**	person, car, truck, traffic light, skateboard
	knife, spoon, microwave, oven, sink, refrigerator	microwave (1.00), oven (1.00), sink (1.00), refrigerator (1.00), knife (0.98), **[spoon (0.38)]**	knife, spoon, microwave, oven, sink, refrigerator

	person, skis, snowboard	person (1.00), skis (1.00), **[snowboard (0.20)]**	person, skis, snowboard
	person, truck, suitcase	truck (1.00), suitcase (0.98), **[person (0.24)]**	person, truck, suitcase
 The Washington Humane Society washhumane.org Meet your new best friend here!	car, bus, truck, cat, dog	car (0.99), truck (0.99), cat (0.96), **[bus (0.37), dog (0.15)]**	car, bus, truck, cat, dog
	fork, sandwich, hot dog, dining table	fork (1.00), hot dog (1.00), dining table (0.90), **[sandwich (0.28)]**	fork, sandwich, hot dog, dining table

	person, bottle, wine glass, chair, dining table	person (1.00), bottle (1.00), wine glass (1.00), dining table (0.99), **[chair (0.25)]**	person, bottle, wine glass, chair, dining table
	person, sports ball, baseball glove,	person (1.00), baseball glove (1.00), **[sports ball (0.27)]**	person, sports ball, baseball glove,
	person, teddy bear	person (1.00), teddy bear (1.00), **[handbag (0.58), clock (0.96)]**	person, teddy bear
	person, cup, chair, dining table, cell phone,	person (1.00), cup (1.00), chair (1.00), dining table (1.00), cell phone (1.00), **[clock (0.92), potted plant (0.85)]**	person, cup, chair, dining table, cell phone,

	cup, banana, apple	cup (1.00), banana (1.00), apple (1.00), **[spoon (0.90), fork (0.55)]**	cup, banana, apple
	cup, banana	banana (1.00), **[bottle (0.90), oven (0.66)]**, cup (0.54)	cup, banana
	train	train (1.00), **[traffic light (0.90), car (0.56), , truck (0.51)]**	train

	person, bowl, oven	person (1.00), oven (1.00) , bowl (0.99), **[chair (0.85), spoon (0.59)]**	person, bowl, oven
	person, cup, spoon, cake, dining table	cup (1.00), spoon (1.00), cake (1.00), dining table (0.95), person (0.91), **[donut (0.89), fork (0.65)]**	person, cup, spoon, cake, dining table
	bird, skateboard, couch	bird (1.00), skateboard (1.00), couch (0.73)	bird, skateboard, **[couch]**

	person, bench, suitcase	person (1.00), suitcase (1.00), bench (0.57)	person, suitcase, **[bench]**
	bed, clock	bed (1.00), clock (0.51)	bed, **[clock]**
	bicycle, car, cat, bottle	bicycle (1.00), car (1.00), cat (1.00), bottle (0.77)	bicycle, car, cat, **[bottle]**
	person, bottle, wine glass, fork, bowl, dining table, sandwich	person (1.00), bottle (1.00), wine glass (1.00), fork (1.00), bowl (1.00), dining table (0.99), sandwich (0.52)	person, bottle, wine glass, fork, bowl, dining table, **[sandwich]**