Supplementary Material

A Queries with Negated Variables

Section 4.2 mentions that although the complement of a box is not a box, queries involving negated variables can be calculated exactly with Inclusion-Exclusion, demonstrated in Table 5. While there are many more interesting and efficient approaches, we simply use the formula for calculating the volume of the union of hyperrectangles (a standard Inclusion-Exclusion formula).

This is equivalent since the intersection of complements of boxes is the complement of the union of boxes. We first intersect all of the non-negated variables into one conjunction box, T. We then calculate the volume of the union of T with all of the boxes representing complements of negated variables $F = \neg f_1, \neg f_2, \neg f_3, ...$, $v_1 = (T \cup f_1 \cup f_2 \cup f_3 ...) = 1 - P(\neg T, \neg f_1, \neg f_2, \neg f_3, ...)$, and the volume of just the negated variables' boxes, $v_2 = (f_1 \cup f_2 \cup f_3 ...) = 1 - P(\neg f_1, \neg f_2, \neg f_3, ...)$. The probability of the query is $v_1 - v_2 = P(F) - P(\neg T, F) = (P(T, F) + P(\neg T, F)) - P(\neg T, F) = P(T, F)$, which was the original query.

P(deer | ...)	
P(deer)	0.12
¬white	0.13
animal	0.50
¬white,animal	0.54
¬white,animal,herbivore	0.73
¬white, animal, herbivore, ¬rabbit	0.80
¬white, animal, ¬herbivore,¬rabbit	0.00

Table 5: Negated variables: queries on the toy data with negated variables, calculated with Inclusion-Exclusion.

B Properties of the Box Lattice

In this section, we cover some technical details about the box lattice model and its properties especially as compared to the order embedding model.

B.1 Non-Distributivity

A lattice is called *distributive* if the following identity holds for all members x, y, z:

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

Claim. *Order embeddings form a distributive lattice.*

Proof. This is a standard results on vector lattices shown in e.g. (Zaanen, 1997)

A non-distributive lattice is a strictly more general object, capable of modeling more objects since it does not necessarily need to fulfill the above identity for all triples x, y, z.

Claim. *The box lattice is non-distributive.*

Proof. Consider the box lattice in 1-dimension. Let $x = [0, 0.3]$, $y = [0.2, 0.6]$, and $z = [0.5, 1.0]$. Then $x \wedge (y \vee z) = [0.2, 0.3]$, but $(x \wedge y) \vee (x \wedge z) = [0, 0.6] \vee \perp = [0, 0.6]$.

This proves that the box lattice is a strict generalization of order embeddings, and not equivalent to order embeddings of any dimensionality. Additionally, our choice of an example containing disjoint elements hints at the importance of non-distributivity for our goal of modeling disjoint events.

B.2 Pseudocomplemented

A lattice is called *pseudocomplemented* if for every element x there exists a unique greatest element in the lattice x^* that is disjoint from x and $x \wedge x^* = \bot$. The box lattice is almost always pseudocomplemented, aside from symmetry concerns (for example, a perfectly centered cube in the 2-dimensional box lattice of side length < 1 has 4 possible equally large pseudocomplements. However any such symmetries can always be infinitesimally perturbed without breaking order structure so the box lattice is pseudocomplemented in a measure-theoretic sense. However, these pseudocomplements can be arbitrarily bad approximations of the true complement set of a box, with the worst case scenario coming from large, nearly-centered cubes.

C Asymmetrizing Score Matrices

C.1 Probabilistic Models

Assume we have a pairwise CPD between Bernoulli variables, and also have access to the unary marginals for each Bernoulli, and further that no unary marginals are exactly identical. If they are exactly identical, we can generate random independent Bernoulli parameters and their JPD, and take a small convex combination with that to infinitesimally perturb the statistics, so this proof is valid everywhere but on a set of measure 0 which we can approximate arbitrarily well.

Claim. *If all unary marginals are distinct, taking the elements of the pairwise CPD, removing the diagonal, and deleting an entry if $P(A|B) < P(B|A)$, that is if $A_{ij} < A_{ji}$, will result in a weighted adjacency matrix for an acyclic directed graph*

Proof. Order the variables $x_1 ... x_n$ so that $p(x_i) < p(x_j)$ if $i < j$. Now an entry of the CPD $p(x_i|x_j) = p(x_i, x_j)/p(x_j) = C_{ij}$ is less than $C_{ji} = p(x_i, x_j)/p(x_i)$ if $p(x_i) < p(x_j)$. So with the variables so ordered, if we use the CPD to create an adjacency matrix with an edge $C_{ij} = 1$ if and only if $p(x_i) < p(x_j)$, it will be upper triangular with 0 on the diagonal. This is a nilpotent matrix which means it is the adjacency matrix of an acyclic graph. This can be easily seen since the entries of A^k are the set of K-hop neighbors, and if this set eventually becomes empty, as in a nilpotent matrix, we have no cycles.

Since the labeling of our vertices is arbitrary, this means that our adjacency matrix created by the proposed asymmetrizing procedure is always acyclic since it is similar to an upper triangular matrix with 0s on the diagonal.

This holds as long as the unary marginals can always be ordered (which they can be except on a set of measure zero, and in practice on it seems to work even if you ignore this constraint.

C.2 KL Divergences and Gaussian Embeddings

Assume the same setup as section C.1, but the scores in the matrix come from (possibly thresholded if $A_{ij} - A_{ji} < c$) pairwise divergences between Gaussian embeddings.

Claim. *There exist graphs produced by the above procedure that do not lead to directed acyclic graphs if thresholded by deleting entries when $A_{ij} < A_{ji}$:*

Proof. Consider the following set of 5 2-dimensional Gaussians with diagonal covariance:

$$
\begin{aligned}
G_1 &= \mathcal{N}(x_1; [-5, -3], \operatorname{diag}([3, 7])) \\
G_2 &= \mathcal{N}(x_2; [-3, 5], \operatorname{diag}[(7, 4])) \\
G_3 &= \mathcal{N}(x_3; [-5, -6], \operatorname{diag}([8, 1])) \\
G_4 &= \mathcal{N}(x_4; [-7, 6], \operatorname{diag}([5, 5])) \\
G_5 &= \mathcal{N}(x_5; [9, 3], \operatorname{diag}([5, 9]))
\end{aligned}
$$

Applying asymmetrization and even pruning at a threshold of $c = 1$ (which is non-nilpotent and does affect edges) produces a cycle between nodes 5, 1, and 3. There are certain repeated numbers in the

parameters, but this is not the cause of the issue. They are whole numbers for ease of exposition, they were randomly generated and many more examples can be created with arbitrary floating point numbers.

C.3 Order Embeddings

We simulated many millions of random sets of order embedding parameters, and created pairwise graphs using the order embedding energy function, and were never able to find a cycle in the resulting asymmetrized graphs. We conjecture that this is because the order embedding energy is essentially a Lagrangian relaxation term penalizing the violation of a true partial order relation, but have not proven it.

Conjecture. *Sets of Order Embeddings can be consistently asymmetrized into directed acyclic graphs according to the procedure in section C.1.*

D Model Parameters

D.1 WordNet Parameters

Since the WordNet data has binary $0, 1$ links instead of calibrated probabilities, and the negative links are found from random negative sampling, we constrain the delta embedding to not update for negative samples during optimization. We found this was effective in preventing random negative samples from decreasing the volume of the boxes and creating artificially disjoint pairs.

The WordNet parameters that achieved best performance on the development set (whose train set performance we reported) are:

```
batch size: 800
dimension: 50
edge loss weight: 1.0
unary loss weight: 9.0
learning rate: 0.001
minimum dimension delta size: 1e-6
dimension-max regularization weight: 0.005
optimizer: Adam
```

For WordNet training with additional soft CPD edges, we use the same parameters. We also perform pruning on the generated CPD file. We only include $\langle t_1, t_2 \rangle$ pairs with probability ≥ 0.6 and the reverse pair $\langle t_2, t_1 \rangle \leq 0.4$ probability.

We tune the batch size of the model between 800 and 40000 because bigger batch size facilitates faster training. We also sweep over 1.0 to 9.0 for edge loss weight and 9.0 to 1.0 for the unary loss weight. The learning rate we tune in $\lambda \in \{0.001, 0.0001\}$. The minimum dimension delta size we tune in $\in \{0.01, 0.001, 0.0001, 0.00001, 0.000001\}$. The dimension-max regularization encourages the upper bound of box to be close 1.0 with an L1 penalty to prevent collapse. We perform parameter search in $\{0.0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$.

D.2 Flickr Parameters

The Flickr parameters that achieved best performance on the development set (whose train set performance we reported) are:

```
batch size: 512
dropout: 0.5
unary loss weight: 8.0
edge loss weight: 2.0
learning rate: 0.0001
minimum dimension delta size: 1e-6
optimizer: Adam
```

The LSTM parameters are initialized with Glorot initialization (Glorot and Bengio, 2010), as are the weight and bias parameters for the feedforward networks to produce the box minimums. The network to produce the Δ embedding is initialized from a uniform distribution from $[15.0, 15.50]$. We clip to zero for min embeddings (apply a ReLU), and apply a softplus to enforce the positivity and minimum dimension size constraints on the Δ embeddings.

We also sweep over 1.0 to 9.0 for edge loss weight and 9.0 to 1.0 for the unary loss weight. The learning rate $\lambda \in \{0.001, 0.0001\}$. We tried Glorot initialization with the Δ network as well, but since we wanted a high degree of overlap at the beginning of training, we simply swept over different uniform initialization ranges in $[5.0, 5.5]$, $[10.0, 10.5]$ and $[15.0, 15.5]$.