Dynamic Cutset Networks: Supplementary Material

1 TRACTABILITY OF DAOCCN

In this section, we will describe the proof for Theorem 4 which is re-stated below for convenience.

Theorem 4. *The forward algorithm has linear time complexity in the size of the unrolled DAOCCN if the following condition is satisfied: the projection of the AOCT associated with the transition distribution on $\boldsymbol{X}^{t-1}$ dominates the AOT associated with the AOCN representation of $\alpha(\boldsymbol{x}^{t-1})$ for all $2 \leq t \leq T$ where T is the total number of time slices.*

Proof. We will prove this theorem using induction. Let us first define the following terminologies.

Terminology used
AOCT: AND/OR Conditional Tree
AOT: AND/OR Tree
AOCN: AND/OR Cutset Network
AOCCN: AND/OR Conditional Cutset Network

Let C^{t-1} be the AOCN associated with the joint distribution over all variables in time slice $t-1$ given evidence i.e. $\alpha(\boldsymbol{x^{t-1}}) = P(\boldsymbol{x^{t-1}}|\boldsymbol{e^{1:t-1}})$ and let $\mathcal{E}^{t|t-1}$ be the AOCCN associated with the transition distribution $P(\boldsymbol{x^t}, \boldsymbol{e^t}|\boldsymbol{x^{t-1}}, \boldsymbol{e^{t-1}})$. Now let $\mathcal{T}^{t-1}$ and $\mathcal{T}^{t|t-1}$ be the AOT and AOCT associated with C^{t-1} and $\mathcal{E}^{t|t-1}$ respectively. Without loss of generality, let us assume that $\mathcal{T}^{t-1}$ and $\mathcal{T}^{t|t-1}$ are defined on the same set of variables. If not, then we can repeatedly use the projection operation defined earlier to ensure that this condition is satisfied.

We already know that $\mathcal{T}^{t-1}$ is dominated by $\mathcal{T}^{t|t-1}$ (i.e. $\mathcal{T}^{t|t-1}$ is an I-map of $\mathcal{T}^{t-1}$). We can, therefore, transform the AOCN C^{t-1} into a new AOCN $C^{t|t-1}$ defined according to the structure of $\mathcal{T}^{t|t-1}$ as follows: (1) compute the branch probabilities of $\mathcal{T}^{t|t-1}$ by performing inference over C^{t-1} and (2) compute each leaf distribution $L_i^{t|t-1}$ as $\prod_j L_j^{t-1}$ such that $x_{path_{\mathcal{T}^{t-1}}(L_j^{t-1})} \cap x_{path_{\mathcal{T}^{t|t-1}}(L_i^{t|t-1})} \subseteq x_{path_{\mathcal{T}^{t|t-1}}(L_i^{t|t-1})}$ for all j. Here, $x_{path_{\mathcal{T}^{t-1}}(L_j^{t-1})}$ refers to the random variables along the path of tree T^{t-1} to the jth leaf L_j^{t-1}. Clearly, this transformation can be performed tractably since C^{t-1} is a tractable model.

We will now show that the joint distribution $P(\boldsymbol{x^t}, \boldsymbol{x^{t-1}}, \boldsymbol{e^t}|\boldsymbol{e^{1:t-1}})$ can also be represented using tree $\mathcal{T}^{t|t-1}$. Let us use the AOCN $C^{t-1,t}$ to represent this distribution. $C^{t-1,t}$ can be easily constructed from $C^{t|t-1}$ by multiplying the leaf distributions $L_j^{t|t-1}$ of $C^{t|t-1}$ with the corresponding $L_i^{t|t-1}$ from $\mathcal{E}^{t|t-1}$. Since the number of leaf distributions is bounded by the model size, this step can also be performed tractably.

Finally, we wish to obtain the message $\alpha(\boldsymbol{x^t}) = P(\boldsymbol{x^t}|\boldsymbol{e^{1:t}})$. We can easily obtain this from $C^{t-1,t}$ by summing out over x^{t-1} which will result in a hierarchical mixture model with multiple latent variables. Let us call this AOCN C^t. Since it is already given that the projection of C^t on the subset of $\boldsymbol{X^t}$ is dominated by the AOCT $T^{t+1|t}$ of the transition distribution $P(\boldsymbol{x^{t+1}}, \boldsymbol{e^{t+1}}|\boldsymbol{x^t}, \boldsymbol{e^t})$, we can repeat this procedure for all time slices up to T. This completes the proof. □

Fig. 3 illustrates the concept of tree transformations used in the theorem with an example. Next, we provide constructive proofs for the case when the AOCT associated with the transition distribution is an OR tree; namely proofs that do not use properties of AND/OR graphs such as dominance and only use first principles. We consider two cases: (1) DAOCCNs with OR Conditional Cutset Networks and No Evidence; and (2) DAOCCNs with OR Conditional Cutset Networks and Evidence.

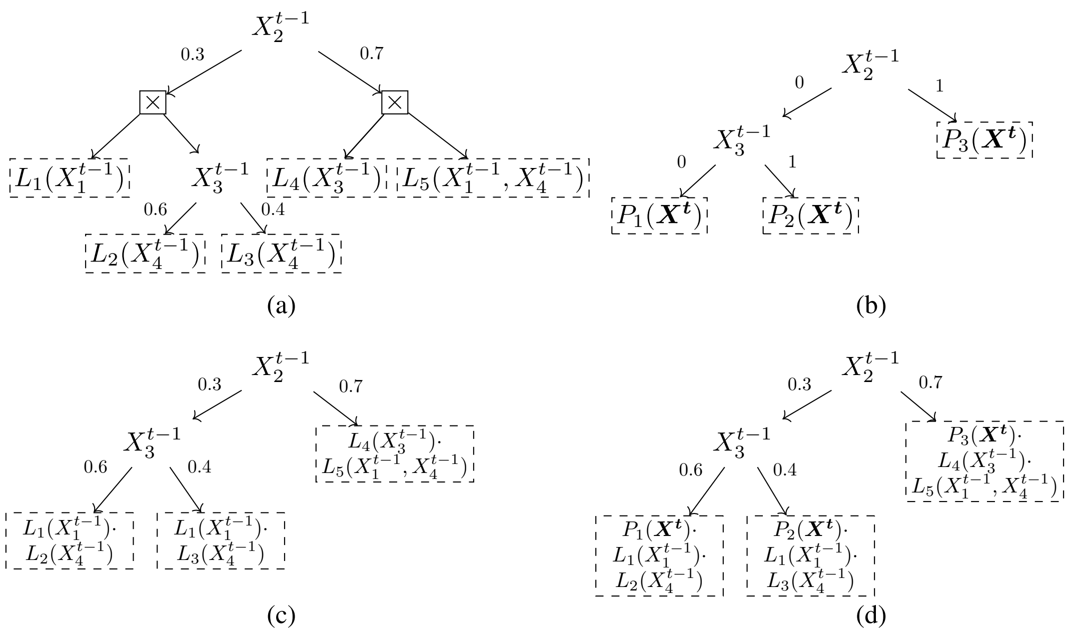
X_2^{t-1}
0.3
0.7
$L_1(X_1^{t-1})$
X_3^{t-1}
0.6
0.4
$L_2(X_4^{t-1})$
$L_3(X_4^{t-1})$
$L_4(X_3^{t-1})$
$L_5(X_1^{t-1}, X_4^{t-1})$
(a)
X_2^{t-1}
0
1
X_3^{t-1}
$P_1(\boldsymbol{X^t})$
$P_2(\boldsymbol{X^t})$
$P_3(\boldsymbol{X^t})$
(b)
$L_1(X_1^{t-1}) \cdot L_2(X_4^{t-1})$
$L_1(X_1^{t-1}) \cdot L_3(X_4^{t-1})$
$L_4(X_3^{t-1}) \cdot L_5(X_1^{t-1}, X_4^{t-1})$
(c)
$P_1(\boldsymbol{X^t}) \cdot L_1(X_1^{t-1}) \cdot L_2(X_4^{t-1})$
$P_2(\boldsymbol{X^t}) \cdot L_1(X_1^{t-1}) \cdot L_3(X_4^{t-1})$
$P_3(\boldsymbol{X^t}) \cdot L_4(X_3^{t-1}) \cdot L_5(X_1^{t-1}, X_4^{t-1})$
(d)

Figure 3: This figures shows (a) a sample AOCN C^{t-1} that represents the joint distribution $\alpha(\boldsymbol{x^{t-1}}) = P(\boldsymbol{x^{t-1}}|\boldsymbol{e^{1:t-1}})$ with the jth leaf distribution represented by $L_j(\boldsymbol{Y^{t-1}})$ where $\boldsymbol{Y^{t-1}} \subseteq \boldsymbol{X^{t-1}}$ (b) a sample AOCCN $\mathcal{E}^{t|t-1}$ that represents the transition distribution $P(\boldsymbol{x^t}, \boldsymbol{e^t}|\boldsymbol{x^{t-1}}, \boldsymbol{e^{t-1}})$ where the ith leaf distribution is represented by $P_i(\boldsymbol{X^t})$ (c) AOCN $C^{t|t-1}$ whose AOT follows the same format as the AOCT in $\mathcal{E}^{t|t-1}$ by making parameter transformations and (d) AOCN $C^{t-1,t}$ that represents the joint distribution $P(\boldsymbol{x^{t-1}}, \boldsymbol{x^t}, \boldsymbol{e^t}|\boldsymbol{e^{1:t-1}})$ obtained by multiplying $C^{t|t-1}$ with $\mathcal{E}^{t|t-1}$. The AND nodes are represented using the rectangular nodes while the leaf nodes are represented using dashed rectangles. The values along the OR node branches represent context-specific probability values and individual domain values of the conditioning variable in the AOTs and AOCTs respectively.

1.1 DAOCCNs with OR Cutset Networks and No Evidence

Here, we are interested in computing the marginal probability distribution over variables in the last time slice given no evidence. For large T, the marginal distribution is roughly equal to the stationary distribution associated with the Markov chain. Let $\boldsymbol{X^t} = \{X_1^t, \ldots, X_n^t\}$ be the set of query variables at time slice t. We define the forward message $\alpha(\boldsymbol{X^t})$ at the t-th time slice as a cutset network (or a mixture of cutset networks) that defines the following distribution:

$$\alpha(\boldsymbol{X^t}) = P(\boldsymbol{X^t}) \tag{1}$$

Let $\{f_1^t, .., f_o^t\}$ be a set of mutually exclusive and collectively exhaustive features defined over $\boldsymbol{X^t}$. Since the features are mutually exclusive and collectively exhaustive, we can treat this set as a random variable F^t and define a probability distribution $P(F^t)$ on it such that $P(F^t = f_j^t)$ represents the probability that feature f_j^t is True. We also define the conditional distribution $P(\boldsymbol{X^t}|F^t = f_j^t)$ which denotes the distribution of $\boldsymbol{X^t}$ given that a certain feature f_j^t is True. For instance, $P(X_1^t = 0, X_2^t = 1, X_3^t = 1|X_1^t \vee X_2^t)$ will give the conditional probability mass of the assignment $(X_1^t = 0, X_2^t = 1, X_3^t = 1)$ conditioned on the feature $X_1^t \vee X_2^t = 1$. The joint distribution $P(\boldsymbol{X^t}, \boldsymbol{X^{t-1}})$ can therefore be expressed as follows:

$$P(\boldsymbol{X^t}, \boldsymbol{X^{t-1}}) = \sum_{j=1}^{m} P(\boldsymbol{X^t}|f_j^{t-1}, \boldsymbol{X^{t-1}}) \cdot P(\boldsymbol{X^{t-1}}|f_j^{t-1}) \cdot P(f_j^{t-1}) \tag{2}$$

Under the assumption that each feature f_j^{t-1} is associated with the leaf node of the OR conditional tree (Note that each leaf node of an OR conditional tree can be thought of as a feature obtained by conjoining the assignments along the path from the root to the leaf. Then, the set of leaf nodes of an OR tree is a set of mutually exclusive and exhaustive features), we have $P(\boldsymbol{X^t}|f_j^{t-1}, \boldsymbol{X^{t-1}}) = P(\boldsymbol{X^t}|f_j^{t-1})$. Therefore, we can rewrite Eq. (2) as:

$$P(\boldsymbol{X^t}, \boldsymbol{X^{t-1}}) = \sum_{j=1}^{m} P(\boldsymbol{X^t}|f_j^{t-1}) \cdot P(\boldsymbol{X^{t-1}}|f_j^{t-1}) \cdot P(f_j^{t-1}) \tag{3}$$

We can generate the message $\alpha(\boldsymbol{X^t})$ from this distribution by summing out $\boldsymbol{X^{t-1}}$ as follows:

$$\begin{aligned}
\alpha(\boldsymbol{X^t}) &= \sum_{\boldsymbol{x^{t-1}}} P(\boldsymbol{X^t}, \boldsymbol{x^{t-1}}) && (4)\\
&= \sum_{\boldsymbol{x^{t-1}}} \sum_{j=1}^{m} P(\boldsymbol{X^t}|f_j^{t-1}) \cdot P(\boldsymbol{x^{t-1}}|f_j^{t-1}) \cdot P(f_j^{t-1}) && (5)\\
&= \sum_{j=1}^{m} P(\boldsymbol{X^t}|f_j^{t-1}) \cdot P(f_j^{t-1}) \cdot \Big(\sum_{\boldsymbol{x^{t-1}}} P(\boldsymbol{x^{t-1}}|f_j^{t-1})\Big) && (6)\\
&= \sum_{j=1}^{m} P(\boldsymbol{X^t}|f_j^{t-1}) \cdot P(f_j^{t-1}) && (7)
\end{aligned}$$

This is nothing but a mixture model with m components over $\boldsymbol{X^t}$ and is tractable to compute. Specifically, the distributions $P(\boldsymbol{X^t}|f_j^{t-1})$ correspond to the leaf nodes of the conditional model while the mixture probabilities $P(f_j^{t-1})$ can be computed tractably (in linear time) from the previous message $\alpha(\boldsymbol{X^{t-1}})$ (since it is also tractable) and projected on to the branches of the new structure to create a mixture of cutset networks.

In summary, the forward algorithm runs in linear time when the AOCT associated with the transition distribution is an OR tree and we have no evidence. Next we show that the forward algorithm runs in linear time in presence of evidence when the AOCT associated with the transition distribution is an OR tree.

1.2 DAOCCNs with OR Conditional Cutset Networks and Evidence

Let $\boldsymbol{X^t} = \{X_1^t, .., X_n^t\}$ and $\boldsymbol{E^t} = \{E_1^t, .., E_m^t\}$ be the set of query and evidence variables at time slice t and G^t be the random variable associated with the set of o mutually exclusive and collectively exhaustive features $\{g_1^t, .., g_o^t\}$ defined over $\boldsymbol{X^t}$ and $\boldsymbol{E^t}$. The forward message $\alpha(\boldsymbol{X^t})$ for this version can be defined as follows:

Algorithm 1 LearnAOCN

Input: Training set D, variable set $\boldsymbol{X}$, max height H, sample threshold T
Output: AND/OR cutset network C representing $P(\boldsymbol{X})$

```
1:  if $H = 0$ or $|D| <= T$ then
2:      return $LearnChowLiuTree(D, \boldsymbol{X})$
3:  $\boldsymbol{X_{partitions}} \leftarrow$ Heuristically divide $\boldsymbol{X}$ into a number of disjoint partitions $\{\boldsymbol{X_{p_1}}, .., \boldsymbol{X_{p_m}}\}$ such that $\bigcup_{i=1}^{m} \boldsymbol{X_{p_i}} = \boldsymbol{X}$
    and $\boldsymbol{X_{p_i}} \cap \boldsymbol{X_{p_j}} = \emptyset$ for all $i \neq j$
4:  if $|\boldsymbol{X_{partitions}}| > 1$ then
5:      $C.root\_type \leftarrow AND$
6:      $C.probabilities \leftarrow \{\ \}$
7:      $C.children \leftarrow \{\ \}$
8:      for each $\boldsymbol{X_{p_i}}$ in $\boldsymbol{X_{partitions}}$ do
9:          $C.children \leftarrow C.children \cup LearnAOCN(D, \boldsymbol{X_{p_i}}, H-1, T)$
10: else
11:     $C.root\_type \leftarrow OR$
12:     $X_v \leftarrow$ Heuristically select from $\boldsymbol{X}$ for splitting
13:     for each $d$ in $domain(X_v)$ do
14:         $D_{X_v=d} \leftarrow$ select all instances in D where $X_v = d$ and drop feature corresponding to $X_v$
15:         $C.probabilities[d] \leftarrow |D_{X_v=d}|/|D|$
16:         $C.children[d] \leftarrow LearnAOCN(D_{X_v=d}, \boldsymbol{X} \setminus X_v, H-1, T)$
17: return $C$
```

$$\alpha(\boldsymbol{X^t}) = P(\boldsymbol{X^t}, \boldsymbol{e^t} | \boldsymbol{e^{1:t-1}}) \tag{8}$$

Similar to the procedure used in the previous subsection, we calculate $\alpha(\boldsymbol{X^t})$ by summing out $\boldsymbol{X^{t-1}}$ from the joint distribution $P(\boldsymbol{X^t}, \boldsymbol{X^{t-1}}, \boldsymbol{e^t} | \boldsymbol{e^{1:t-1}})$:

$$\begin{aligned}
\alpha(\boldsymbol{X^t}) &= \sum_{\boldsymbol{x^{t-1}}} P(\boldsymbol{X^t}, \boldsymbol{x^{t-1}}, \boldsymbol{e^t} | \boldsymbol{e^{1:t-1}}) \\
&= \sum_{\boldsymbol{x^{t-1}}} \sum_{j=1}^{m} P(\boldsymbol{X^t}, \boldsymbol{e^t} | g_j^{t-1}) \cdot P(\boldsymbol{x^{t-1}} | g_j^{t-1}, \boldsymbol{e^{1:t-1}}) \cdot P(g_j^{t-1} | \boldsymbol{e^{1:t-1}}) \\
&= \sum_{j=1}^{m} P(\boldsymbol{X^t}, \boldsymbol{e^t} | g_j^{t-1}) \cdot P(g_j^{t-1} | \boldsymbol{e^{1:t-1}}) \cdot \Big(\sum_{\boldsymbol{x^{t-1}}} P(\boldsymbol{x^{t-1}} | g_j^{t-1}, \boldsymbol{e^{1:t-1}}) \Big) \\
&= \sum_{j=1}^{m} P(\boldsymbol{X^t}, \boldsymbol{e^t} | g_j^{t-1}) \cdot P(g_j^{t-1} | \boldsymbol{e^{1:t-1}})
\end{aligned}$$

Thus $\alpha(\boldsymbol{X^t})$ is a mixture model defined over $\boldsymbol{X^t}$ and $\boldsymbol{e}^t$. $P(\boldsymbol{X^t}, \boldsymbol{e^t} | g_j^{t-1})$ can be directly computed from the conditional model while $P(g_j^{t-1} | \boldsymbol{e^{1:t-1}})$ can be computed tractably, in linear time, from the previous message $\alpha(\boldsymbol{X^{t-1}})$ since:

$$\begin{aligned}
P(g_j^{t-1} | \boldsymbol{e^{1:t-1}}) &= P(g_j^{t-1} | \boldsymbol{e^{t-1}}, \boldsymbol{e^{1:t-2}}) \\
&\propto P(g_j^{t-1}, \boldsymbol{e^{t-1}} | \boldsymbol{e^{1:t-2}})
\end{aligned}$$

Thus, in summary, the forward algorithm has linear time complexity in the size of the unrolled DAOCCN if the AOCT associated with the transition distribution is an OR tree. This proves Corollary 5 (see section 4.1 in the main paper).

2 LEARNING ALGORITHMS FOR DCN

As discussed in the main paper, a DCN comprises a prior distribution of the form $P(\boldsymbol{X^1}, \boldsymbol{E^1})$ and a transition distribution of the form $P(\boldsymbol{X^t}, \boldsymbol{E^t} | \boldsymbol{X^{t-1}}, \boldsymbol{E^{t-1}})$ which can be modeled using an AOCN and a CCN or AOCCN respectively. The

Algorithm 2 LearnCCN

Input: Training set D, variable sets $\boldsymbol{X}, \boldsymbol{Y}$, sample threshold T
Output: Conditional cutset network C representing $P(\boldsymbol{Y}|\boldsymbol{X})$

```
 1: if $|D| <= T$ then
 2:     return $LearnConditionalChowLiuTree(D, \boldsymbol{X}, \boldsymbol{Y})$
 3: for all pairs $(Y_i, Y_j) \in \boldsymbol{Y},\ i \neq j$ do
 4:     Compute $\bar{P}(Y_i, Y_j)$ using calibrated classifiers
 5: $\boldsymbol{Y_{partitions}} \leftarrow$ Heuristically divide $\boldsymbol{Y}$ into a number of disjoint partitions $\{\boldsymbol{Y_{p_1}}, .., \boldsymbol{Y_{p_m}}\}$ such that $\bigcup_{i=1}^{m} \boldsymbol{Y_{p_i}} = \boldsymbol{Y}$ and $\boldsymbol{Y_{p_i}} \cap \boldsymbol{Y_{p_j}} = \emptyset$ for all $i \neq j$
 6: if $|\boldsymbol{Y_{partitions}}| > 1$ then
 7:     $C.root\_type \leftarrow AND$
 8:     $C.probabilities \leftarrow \{\ \}$
 9:     $C.children \leftarrow \{\ \}$
10:     for each $\boldsymbol{Y_{p_i}}$ in $\boldsymbol{Y_{partitions}}$ do
11:         $C.children \leftarrow C.children \cup LearnCNN(D, X, \boldsymbol{Y_{p_i}}, T)$
12: else
13:     $C.root\_type \leftarrow OR$
14:     $Y_v \leftarrow$ Heuristically select from $\boldsymbol{Y}$ for splitting
15:     for each $d$ in $domain(Y_v)$ do
16:         $D_{Y_v=d} \leftarrow$ select all instances in D where $Y_v = d$ and drop feature corresponding to $Y_v$
17:         $C.probabilities[d] \leftarrow \bar{P}(Y_v = d|X)$
18:         $C.children[d] \leftarrow LearnCCN(D_{Y_v=d}, \boldsymbol{X}, \boldsymbol{Y} \setminus Y_v, T)$
19: return $C$
```

LearnAOCN (Algorithm 1) and LearnCCN (Algorithm 2) algorithms have been taken from the original papers by Rahman et al. (2014, 2016c, 2019) and have been modified to fit our notation. The LearnAOCCN algorithm (Algorithm 3) can be used to learn the AOCCN model we introduced in the main paper that guarantees tractable inference for the entire network.

3 EXPERIMENTAL DETAILS

3.1 Dynamic Sum-Product Networks (DSPNs)

The DSPN models had three hyperparameters - the number of interface nodes I (between 2 and 7), the maximum depth of the template network at each time slice (up to 7) and the maximum size of each multivariate node (up to 7). We selected the hyperparameters using the validation set.

3.2 Dynamic tree Bayesian Networks (CLDBNs)

We learn dynamic Bayesian networks as follows: a tree-structured Bayesian Network was learnt over the static data (i.e. $\boldsymbol{X^t}$) using the Chow-Liu tree learning algorithm. This structure was first duplicated for the previous time slice $\boldsymbol{X^{t-1}}$ and then connected to the $\boldsymbol{X^t}$ structure by drawing an edge from each X_v^{t-1} to X_v^t for all $X_v^t \in \boldsymbol{X^t}$. We call this model Chow-Liu Dynamic Bayesian Networks (CLDBNs).

3.3 Long Short Term Memory networks (LSTMs)

The LSTM model comprised a single layer of LSTM nodes with a time-distributed dense layer having sigmoid activations stacked on top (see Figure 4). Each variable $X_v^t \in \boldsymbol{X^t}$ had a corresponding sigmoid node in the output layer per time slice which represented the probability distribution $P(X_v^t|h^{t-1})$ where h^{t-1} is the hidden state information at time slice t. The log-likelihood for each sequence of length T was then computed as $\sum_{t=1}^{T} \sum_v \log P(X_v^t|h^{t-1})$. Each sequence in the dataset was padded with 0's in the beginning in order to model the prior distribution for the first time slice (i.e. $P(\boldsymbol{X^1}|h^0)$) and at the end in order to make all the sequence lengths the same. The hyperparameters used for this model were: (a) the number of hidden units (up to 64) (b) the batch size (up to 64) (c) the dropout percentage from 0.2 to 0.5. Again, the hyperparameters were tuned using the validation set.

Algorithm 3 LearnAOCCN

```
Input: Training set $D$, variable sets $\boldsymbol{X}, \boldsymbol{Y}$, max conditional height $H_1$, max cutset network height $H_2$, conditional sample threshold $T_1$, cutset network sample threshold $T_2$
Output: AND/OR conditional cutset network $C$ representing $P(\boldsymbol{Y}|\boldsymbol{X})$
1: if $H_1 = 0$ or $\boldsymbol{X} = \emptyset$ or $|D| <= T_1$ then
2:     $D' \leftarrow$ drop all features from $D$ corresponding to each $X_v \in \boldsymbol{X}$
3:     return $LearnAOCN(D', \boldsymbol{Y}, H_2, T_2)$
4: $\boldsymbol{X_{partitions}} \leftarrow$ Heuristically divide $\boldsymbol{X}$ into a number of disjoint partitions $\{\boldsymbol{X_{p_1}}, .., \boldsymbol{X_{p_m}}\}$ such that $\bigcup_{i=1}^{m} \boldsymbol{X_{p_i}} = \boldsymbol{X}$
   and $\boldsymbol{X_{p_i}} \cap \boldsymbol{X_{p_j}} = \emptyset$ for all $i \neq j$
5: if $|\boldsymbol{X_{partitions}}| > 1$ then
6:     $C.root\_type \leftarrow AND$
7:     $C.probabilities \leftarrow \{\ \}$
8:     $C.children \leftarrow \{\ \}$
9:     for each $\boldsymbol{X_{p_i}}$ in $\boldsymbol{X_{partitions}}$ do
10:        $C.children \leftarrow C.children \cup LearnAOCCN(D, \boldsymbol{X_{p_i}}, \boldsymbol{Y}, H_1 - 1, H_2, T_1, T_2)$
11: else
12:     $C.root\_type \leftarrow OR$
13:     $X_v \leftarrow$ Heuristically select from $\boldsymbol{X}$ for splitting
14:     for each $d$ in $domain(X_v)$ do
15:         $D_{X_v=d} \leftarrow$ select all instances in D where $X_v = d$ and drop feature corresponding to $X_v$
16:         $C.probabilities[d] \leftarrow |D_{X_v=d}|/|D|$
17:         $C.children[d] \leftarrow LearnAOCCN(D_{X_v=d}, \boldsymbol{X} \setminus X_v, \boldsymbol{Y}, H_1 - 1, H_2, T_1, T_2)$
18: return $C$
```

3.4 Dynamic AND/OR Conditional Cutset Networks (DAOCCNs)

The DAOCCN model was a DCN where the conditional model was a mixture of AOCCNs.It was learnt using the learning algorithm outlined in the previous section with the mixture coefficients being learnt using the EM algorithm. The model has four hyperparameters – (a) the maximum depth of the prior distribution (up to 7) (b) the maximum depth of the decision tree portion of the transition distribution (up to 7) (c) the maximum depth of the cutset leaves (up to 7) of the transition distribution (up to 7) (d) the number of mixture components (up to 10) and (e) the adaptive threshold β for partitioning (tuned between 0.1 to 1.0).

3.5 Dynamic Conditional Cutset Networks (DCCNs)

The DCCN model used a CCN as the conditional model instead of an AOCCN. The calibrated classifier used was logistic regression with L1 regularization with the regularization parameter $\lambda = 1$. For all particle filtering experiments, we took $K = 500$ weighted particles.

Table 3 contains the results for all experiments performed on five real-world datasets.

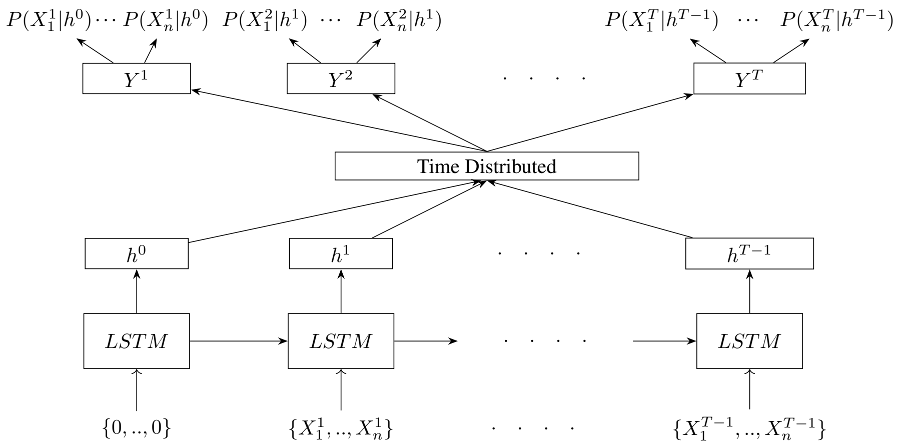
$P(X_1^1|h^0) \cdots P(X_n^1|h^0)$
$P(X_1^2|h^1) \cdots P(X_n^2|h^1)$
$P(X_1^T|h^{T-1}) \cdots P(X_n^T|h^{T-1})$
Y^1
Y^2
Y^T
Time Distributed
h^0
h^1
h^{T-1}
$LSTM$
$LSTM$
$LSTM$
$\{0, .., 0\}$
$\{X_1^1, .., X_n^1\}$
$\{X_1^{T-1}, .., X_n^{T-1}\}$

Figure 4: LSTM architecture used for all experiments. At each time slice t, the input variables of the sequence $\{X_1^t, .., X_n^t\}$ are used to predict a set of marginal distributions $Y^{t+1} = \{P(X_1^{t+1}|h^t), .., P(X_n^{t+1}|h^t)\}$ for the next time slice $t+1$. Note that the input to the first LSTM unit is a 0 vector, which is used to model the prior distribution $P(X^1|h^0)$. These marginal distributions are modeled using sigmoid functions. During inference time, only a fixed percentage of values of $\boldsymbol{X^t}$ (say, $\{X_1^t, .., X_m^t\}$) are observed. The remaining $\{X_m^t, .., X_n^t\}$ are imputed by sampling from the marginals $\{P(X_m^t|h^{t-1}), .., P(X_n^t|h^{t-1})\}$. Each sample (or particle) is weighed according to the posterior likelihood and the particle filtering algorithm is used to propagate these particles forward. The ELL for the entire sequence is simply $\frac{1}{K} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{v=1}^{m} log\ P(X_v^t|h_{(k)}^{t-1})$ where $h_{(k)}^{t-1}$ is the hidden state of the LSTM generated at the $t-1$th time slice using the kth particle as input for a total of K particles.

Table 3: Experimental results on real datasets. The metrics used are (1) Average test-set log-likelihood scores (LL) (2) Average test-set evidence log-likelihood scores for 25% (ELL25), 50% (ELL50) and 75% (ELL75). A *B/V* value is specified below each dataset where B is the total number of bins used for binarization and V is the total number of variables in the final binarized dataset. The following datasets *(#train, #test, #avg_train_length, #avg_test_length)* are considered – *diabetes* $(56, 14, 425, 388)$, *racketsport* $(151, 152, 30, 30)$, *airquality* $(1, 1, 6379, 562)$, *handwriting* $(150, 850, 152, 152)$ and *japanvowels* $(512, 128, 16, 16)$

Model	**Dataset**																	**#wins**
	diabetes	**racketsport**				**airquality**				**handwriting**				**japanvowels**				
	8/23	**8/20**	**16/26**	**32/32**	**64/38**	**8/36**	**16/48**	**32/60**	**64/72**	**8/14**	**16/17**	**32/20**	**64/23**	**8/36**	**16/48**	**32/60**	**64/72**	
									LL									
DSPN	-3.4704	-11.7904	-15.8884	-20.1624	-24.3521	-20.9276	-28.5568	-37.4639	-45.8516	-8.3088	-9.8981	-11.7431	-13.8642	-22.5540	-30.9468	-39.2700	-47.7120	0
CLDBN	-4.2973	-10.0665	-14.0407	-18.1166	-22.2197	-17.5278	-25.7498	-34.0146	-42.4184	-3.5313	-5.0874	-6.7646	-8.5267	-17.4166	-25.6584	-33.9683	-42.2845	0
LSTM	-2.8913	-10.8268	-15.0085	-19.0621	-23.2259	-18.7659	-28.5243	-35.4974	-43.9580	-3.0549	-4.2957	-5.8877	**-7.3221**	-16.1458	-24.4584	-32.9470	-41.5570	1
DAOCCN	-2.9783	-9.4796	-13.4243	**-17.5753**	**-21.6414**	-18.2101	-26.4619	-34.7610	-43.0909	-3.1417	-4.7081	-6.4068	-8.1805	-17.7284	-25.9749	-34.2869	-42.6051	2
DCCN	**-2.7937**	**-9.3250**	**-13.3693**	-17.6721	-22.0475	**-14.1960**	**-22.2492**	**-31.0858**	**-40.4080**	**-2.9192**	**-4.2143**	**-5.7400**	-7.4025	**-14.3364**	**-23.9097**	**-31.2965**	**-40.3600**	**14**
									ELL (25% Evidence)									
DSPN	-1.9445	-3.0569	-4.3808	-5.0503	-6.4482	-4.5045	-6.4277	-8.4905	**-10.5684**	-2.4523	-2.9725	-2.8667	-3.4534	-5.8527	-7.9281	-9.9557	-12.0917	1
CLDBN	-2.0814	-3.5298	-4.8550	-5.5436	-6.8933	-5.0975	-7.1543	-9.2246	-11.3773	-1.8299	-2.6761	-3.2441	-3.8028	-4.3883	-6.3977	-8.4641	-10.5385	0
LSTM	-2.1654	-3.9112	-5.2564	-5.9437	-7.3525	-7.4319	-8.9709	-10.9812	-13.1557	-1.9939	-3.3351	-3.3727	-4.1710	-4.5935	-6.9295	-9.0028	-11.2804	0
DAOCCN	**-1.0421**	**-2.7403**	**-4.0347**	**-4.7387**	**-6.0873**	-4.5915	-6.6539	-8.7217	-10.8143	**-1.0132**	**-1.8496**	**-2.4088**	**-3.0336**	**-3.7340**	**-5.7508**	**-7.8158**	**-9.8950**	**13**
DCCN	-2.0704	-3.4099	-4.7212	-5.4005	-6.7553	**-4.1872**	**-6.1439**	**-8.3865**	-10.5900	-1.9845	-2.4061	-2.5579	-3.0651	-4.1333	-6.2628	-8.2193	-10.4634	3
									ELL (50% Evidence)									
DSPN	-2.4445	-6.1893	-8.2456	-10.2894	-12.4246	-8.2161	-11.8838	-16.2208	-20.3625	-4.0216	-4.9167	-5.3607	-6.6380	-11.4185	-15.6623	-19.7256	-24.0481	0
CLDBN	-2.8427	-6.5708	-8.5814	-10.6583	-12.7149	-9.2754	-13.3765	-17.4808	-21.7152	-3.3787	-4.4094	-5.9666	-6.6982	-8.3541	-12.4169	-16.5576	-20.7098	0
LSTM	-2.5282	-7.2410	-9.3001	-11.3653	-13.4874	-11.1287	-15.8784	-19.7516	-23.9570	-3.4381	-4.3014	-5.3205	-6.5761	-8.5292	-12.6195	-17.0036	-21.2654	0
DAOCCN	**-1.7502**	**-5.6880**	-7.7075	**-9.7921**	**-11.8466**	-8.8828	-11.6483	-15.7663	-21.4694	**-2.4452**	**-3.4493**	-4.9821	-5.7930	-7.7883	**-10.5403**	**-14.6140**	**-19.3742**	**9**
DCCN	-2.4330	-6.3249	**-6.1389**	-10.4417	-12.4971	**-7.2728**	**-11.0823**	**-15.3831**	**-20.0695**	-2.9491	-3.7885	**-4.3633**	**-5.3732**	**-7.3508**	-11.5892	-15.6793	-20.0967	8
									ELL (75% Evidence)									
DSPN	-2.6355	-8.6471	-12.7824	-14.7364	-18.9150	-13.7354	-19.4720	-26.1437	-32.2190	-6.7459	-8.0359	-7.7845	-9.7450	-16.5135	-22.7662	-28.7939	-35.3728	0
CLDBN	-3.7654	-9.0854	-12.9670	-14.8029	-19.2333	-13.0501	-20.1111	-26.3037	-32.6061	-4.2973	-5.8580	-7.5513	-9.2966	-11.3934	-18.1961	-24.4019	-30.6294	0
LSTM	-2.7393	-10.5267	-14.5238	-16.7601	-20.9610	-15.3121	-22.1672	-28.0897	-34.4663	-3.7216	-4.9373	-6.3766	**-7.8521**	-11.8173	-17.8363	-24.4827	-31.0299	1
DAOCCN	**-2.2617**	**-8.0616**	**-11.9545**	**-14.1799**	**-18.2364**	-12.8790	-18.7257	-24.9243	-32.8961	**-3.2241**	**-4.7035**	-6.3454	-8.1366	-10.9022	**-16.4302**	**-22.5774**	**-29.4188**	**10**
DCCN	-2.5129	-8.5676	-12.5698	-14.7609	-18.9057	**-10.8123**	**-16.6209**	**-23.1923**	**-30.0121**	-3.5826	-4.8709	**-6.3243**	-7.9504	**-10.7863**	-17.3695	-23.3537	-30.0285	6