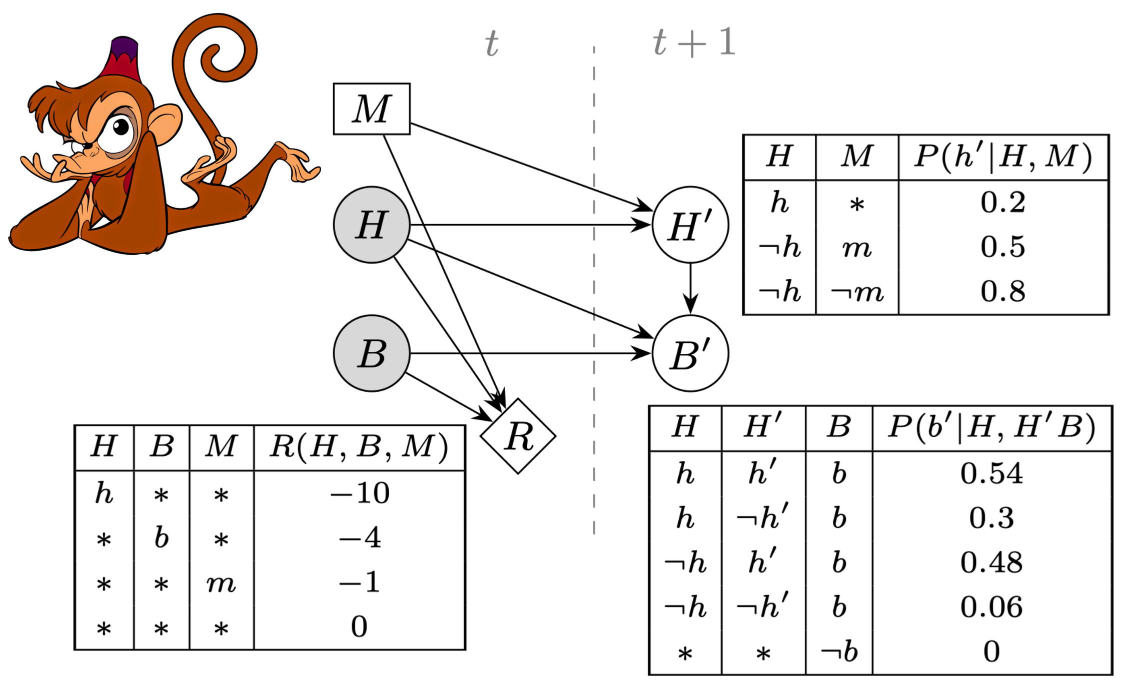
Inference and Learning in Dynamic Decision Networks Using Knowledge Compilation

Gabriele Venturato, Vincent Derkinderen, Pedro Zuidberg Dos Martires, Luc De Raedt

I. Problem Setting

Bayesian network + decisions + time = Markov decision process (MDP)



H	M	$P(h' \mid H, M)$
h	$*$	0.2
$\neg h$	m	0.5
$\neg h$	$\neg m$	0.8

H	B	M	$R(H, B, M)$
h	$*$	$*$	-10
$*$	b	$*$	-4
$*$	$*$	m	-1
$*$	$*$	$*$	0

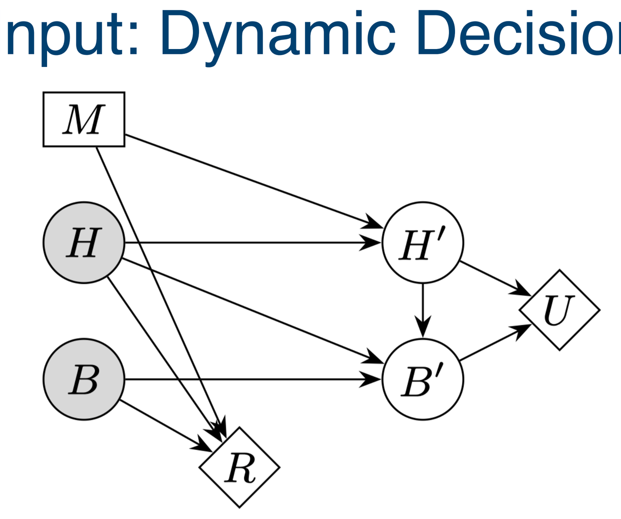
H	H'	B	$P(b' \mid H, H'B)$
h	h'	b	0.54
h	$\neg h'$	b	0.3
$\neg h$	h'	b	0.48
$\neg h$	$\neg h'$	b	0.06
$*$	$*$	$\neg b$	0

- Factored representation
- Exact inference
- (Intra-state) structure
- Discrete time steps
- Time-slicing
- Maximise expected utility

GOAL: **Exploit structure in the MDP** to represent it as a probabilistic circuit for both inference (i.e. planning) and gradient-based parameter learning.

II. Dynamic Decision Circuits (DDCs)

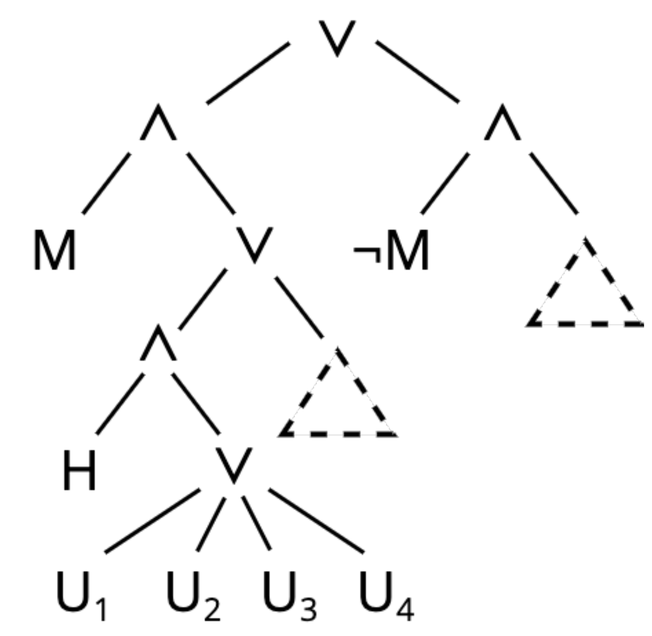
Input: Dynamic Decision Network (DDN)



H'	B'	$U(H', B')$
h'	b'	u_1
h'	$\neg b'$	u_2
$\neg h'$	b'	u_3
$\neg h'$	$\neg b'$	u_4

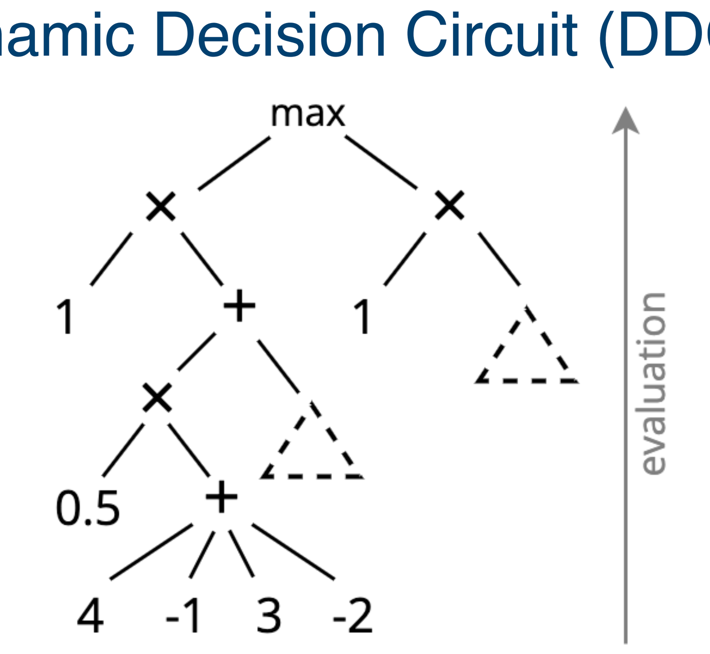
Encoding + Knowledge Compilation

Logic Theory



Labelling

Dynamic Decision Circuit (DDC)



Algebraic Model Counting computes

$$U(s) = \max_{a \in \mathbf{A}} \left\{ R(s, a) + \gamma \sum_{s' \in \mathbf{S}} P(s' \mid s, a) U(s') \right\}$$

III. mapl-cirup

repeat
$U \leftarrow 0$
foreach state s **in** S **do** (explicit)
$U'[s] \leftarrow \max_a \left[R(s, a) + \sum_{s'} P(s' \mid s, a) \gamma U[s'] \right]$ (DDC)
$\delta \leftarrow ||U' - U||$
$U \leftarrow U'$
until $\delta < \epsilon$

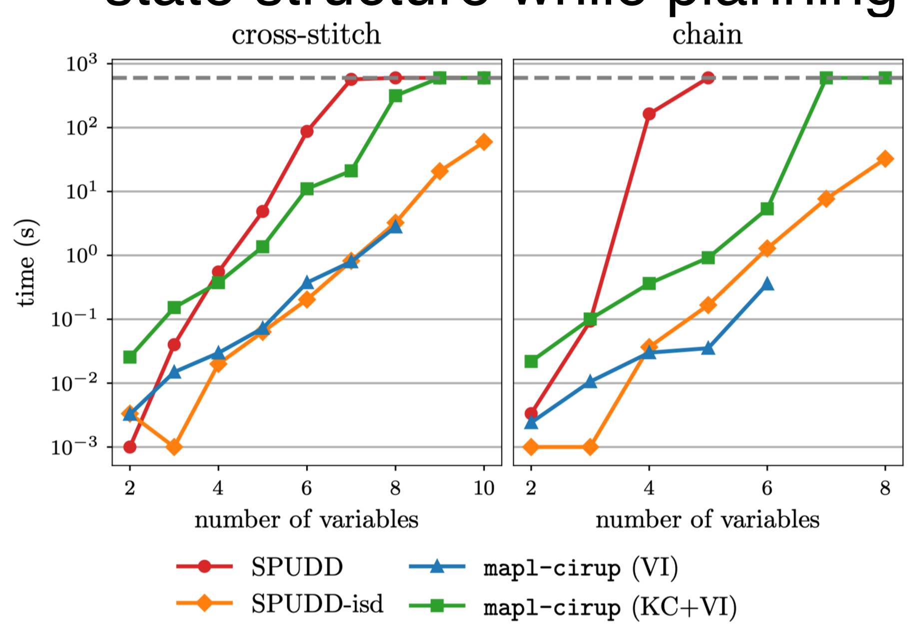
DDCs can be easily integrated in a value iteration algorithm

planning is reduced to inference in DDCs

IV. Experiments

Inference

we are able to exploit intra-state structure while planning

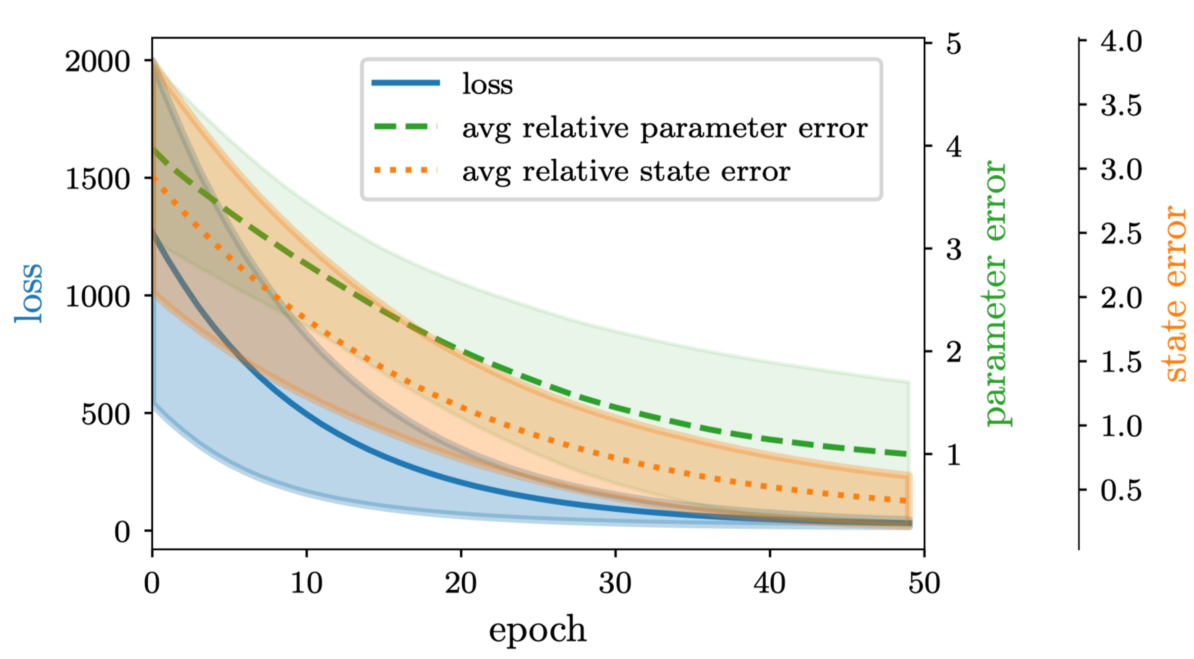


we get more compact circuits than the state of the art

model	$\lvert\mathbf{X}\rvert$	SPUDD $\lvert\Delta\rvert$	SPUDD VI [s]	mapl-cirup $\lvert\Delta\rvert$	mapl-cirup KC [s]	mapl-cirup VI [s]
monkey	2	11 664	< 0.01	163	0.01	0.005
elevator	4	5 794	< 0.01	277	0.02	0.003
coffee	6	142 519	0.03	2542	0.6	0.054
factory	7	38 163	0.01	2932	0.93	0.105

$\lvert\mathbf{X}\rvert$: problem size; SPUDD $\lvert\Delta\rvert$: circuit size; mapl-cirup $\lvert\Delta\rvert$: circuit size; KC [s]: compilation time

Learning (new in this setting!)



we can learn reward parameters from trajectories: $\tau = \langle s_0, a_{0:k}, r_{0:k} \rangle$

V. Contribution

We introduce **dynamic decision circuits**: compact probabilistic circuits that can **exploit structure** for exact MDP planning, and enable **gradient-based parameter learning** for the first time in this setting.

gabriele.venturato@kuleuven.be
@GabVenturato