Keeping the promises and overcoming the challenges of neurosymbolic AI

Lennert De Smet

How can a robot safely bring your favourite pizza to your doorstep?

Neural networks

Backpropagation

Symbolic methods

Logic

Neurosymbolic AI (NeSy) 

Probability theory

Neural networks

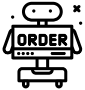
Backpropagation

Symbolic methods

Logic

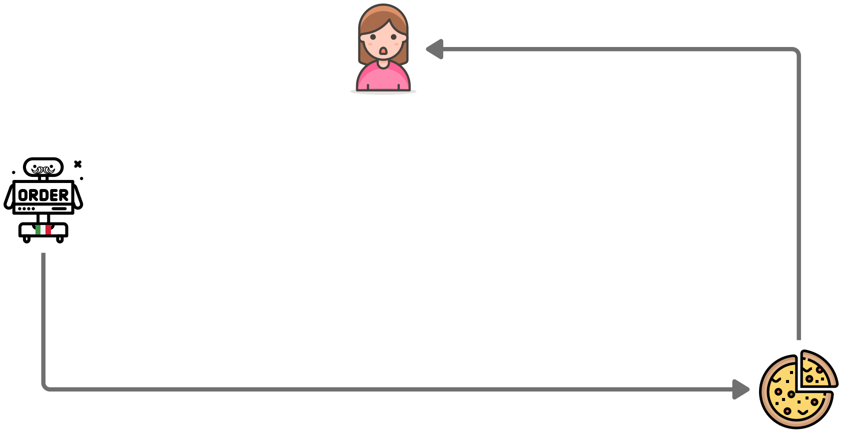
Probabilistic NeSy

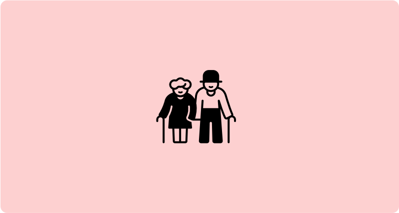
Probability theory


ORDER

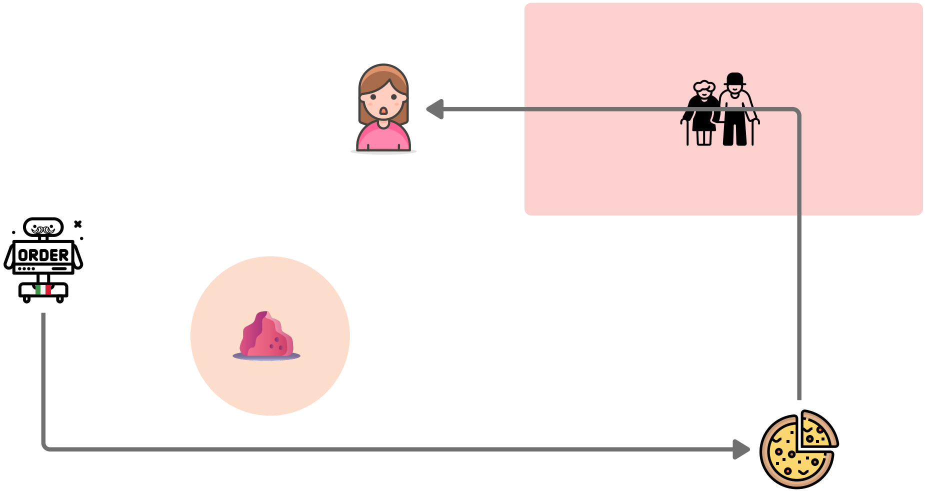

ORDER


ORDER


ORDER

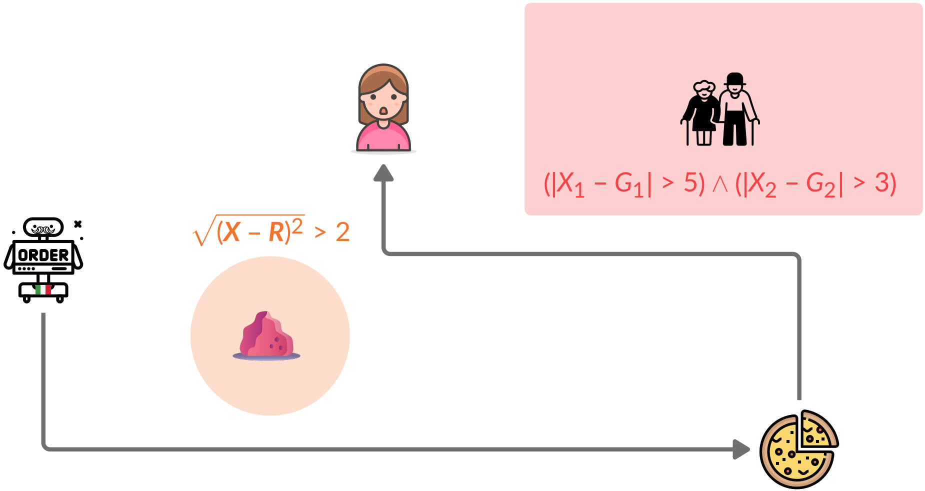



ORDER


ORDER

$$(|X_1 - G_1| > 5) \land (|X_2 - G_2| > 3)$$

$$\sqrt{(\boldsymbol{X} - \boldsymbol{R})^2} > 2$$


$(|X_1 - G_1| > 5) \wedge (|X_2 - G_2| > 3)$
$\sqrt{(X - R)^2} > 2$
ORDER

1. What is neurosymbolic AI and what does it promise?
2. Where does neurosymbolic AI struggle?
3. How can we still make neurosymbolic AI work?

1. **What is neurosymbolic AI and what does it promise?**
   1. Neural + Probabilistic + Symbolic AI
   2. Deep Bayesian networks and neural probabilistic logic
   3. Guaranteed consistency and improved generalisation
2. Where does neurosymbolic AI struggle?
3. How can we still make neurosymbolic AI work?

Neural Models

+ Incredible function approximates

On the left sidewalk there is a young child and on the right sidewalk is an old couple. You are driving on the road and have to take an action, what do you do?

This is a moral and ethical dilemma often presented in discussions about autonomous vehicles and decision-making in critical situations. The question essentially asks you to choose between harming a young child or an old couple, which is a deeply difficult and subjective decision.

Neural Models

+ Incredible function approximates
- Black box, not robust
- Struggle with reasoning, consistency

> On the left sidewalk there is a young child and on the right sidewalk is an old couple. You are driving on the road and have to take an action, what do you do?

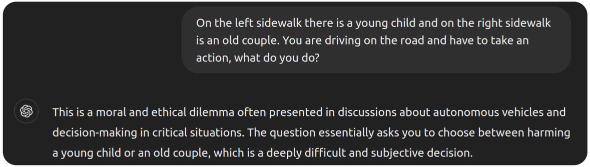
> This is a moral and ethical dilemma often presented in discussions about autonomous vehicles and decision-making in critical situations. The question essentially asks you to choose between harming a young child or an old couple, which is a deeply difficult and subjective decision.

Neural Models

+ Incredible function approximates

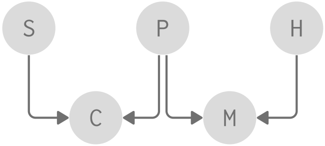
- Black box, not robust

- Struggle with reasoning, consistency



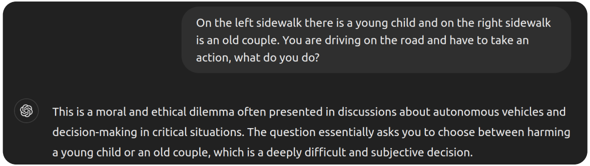
Probabilistic Models

+ Flexible and robust



Neural Models

+ Incredible function approximates
- Black box, not robust
- Struggle with reasoning, consistency

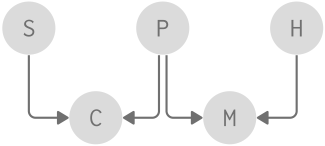


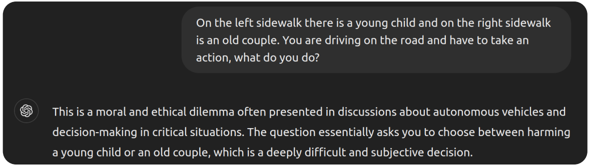
Probabilistic Models

+ Flexible and robust
- Expressivity versus tractability
- No logical reasoning

Neural Models

+ Incredible function approximates
- Black box, not robust
- Struggle with reasoning, consistency





Deep Probabilistic Models

+ Incredible function approximates

+ Flexible and robust

- Struggle with reasoning, consistency

Symbolic Models

+ Consistency of logic

Deep Probabilistic Models

+ Incredible function approximates
+ Flexible and robust
- Struggle with reasoning, consistency

Symbolic Models

+ Consistency of logic
- Not designed for uncertainty
- Can not deal with "raw" data

Deep Probabilistic Models

+ Incredible function approximates
+ Flexible and robust
- Struggle with reasoning, consistency

Probabilistic Neurosymbolic Models

+ Incredible function approximates
+ Flexible and robust
+ Consistency of logical reasoning

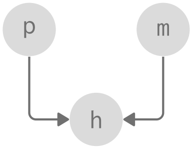
Bayesian networks and probabilistic logic programs encode probability distributions

Bayesian networks

Probabilistic logic programs

Bayesian networks and probabilistic logic programs encode probability distributions

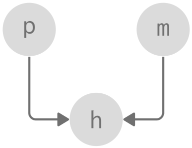
Bayesian networks



Probabilistic logic programs

Bayesian networks and probabilistic logic programs encode probability distributions

Bayesian networks



Defining CPTs

Probabilistic logic programs

Bayesian networks and probabilistic logic programs encode probability distributions

Bayesian networks

$$
\begin{tikzcd}
p \arrow[dr] & & m \arrow[dl] \\
& h &
\end{tikzcd}
$$

Defining CPTs

Probabilistic logic programs

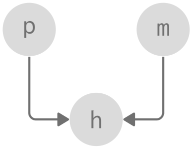
```
0.9 :: player((1, 2)).
0.5 :: monster((1, 2)).

hit ← player(L) ∧ monster(L).
```

$\Longrightarrow \mathbb{P}(\texttt{hit} = \textit{True}) = 0.45$

Bayesian networks and probabilistic logic programs encode probability distributions

Bayesian networks



Probabilistic logic programs

```
0.9 :: player((1, 2)).
0.5 :: monster((1, 2)).

hit ← player(L) ∧ monster(L).
```

$\Longrightarrow \mathbb{P}(\texttt{hit} = \textit{True}) = 0.45$

Defining CPTs

Writing logic programs

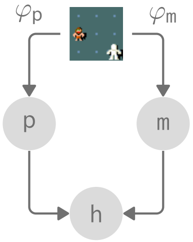
Adding neural parametrisations yields deep BNs and probabilistic neurosymbolic AI

Deep Bayesian networks

Neural Probabilistic logic programs

Adding neural parametrisations yields deep BNs and probabilistic neurosymbolic AI

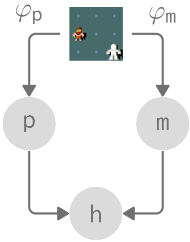
Deep Bayesian networks



Neural Probabilistic logic programs

Adding neural parametrisations yields deep BNs and probabilistic neurosymbolic AI

Deep Bayesian networks



Neural Probabilistic logic programs

φ_p( ) :: player(, (1, 2)).
φ_m( ) :: monster(, (1, 2)).

hit(Img) $\leftarrow$ player(Img, L) $\wedge$ monster(Img, L).

$\Longrightarrow \mathbb{P}($ hit( ) = *True* $) = \varphi_p($ $) \cdot \varphi_m($ $)$

Impose consistency on neural models by conditioning on logical statements

Take a neural distribution $p_{\theta}(\omega)$
and a logic formula φ over ω

Impose consistency on neural models by conditioning on logical statements

Take a neural distribution $p_{\theta}(\omega)$
and a logic formula φ over ω

$$p_{\theta}(\omega \mid \varphi) = \frac{p(\varphi \mid \omega)\, p_{\theta}(\omega)}{p(\varphi)}$$

Impose consistency on neural models by conditioning on logical statements

Take a neural distribution $p_{\theta}(\omega)$
and a logic formula φ over ω

$$p_{\theta}(\omega \mid \varphi) = \frac{p(\varphi \mid \omega)\, p_{\theta}(\omega)}{p(\varphi)}$$

Then Bayes' rule give us new predictions
that are **consistent** with the logic formula

1. What is neurosymbolic AI and what does it promise?
2. Where does neurosymbolic AI struggle?
3. How can we still make neurosymbolic AI work?

Probabilistic inference is *#P*-hard and adding logic complicates things further

Take a neural distribution $\boldsymbol{p}_{\boldsymbol{\theta}}(\omega)$
and a logic formula φ over ω

Probabilistic inference is **#*P*-hard** and adding logic complicates things further

Take a neural distribution $p_{\boldsymbol{\theta}}(\omega)$
and a logic formula φ over ω

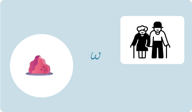
$$p(\varphi) = \int_{\Omega} \mathbb{1}_{\omega \models \varphi} \, p_{\boldsymbol{\theta}}(\omega) \, \mathrm{d}\omega$$

Probabilistic inference is *#P*-hard
and adding logic complicates things further

Take a neural distribution $p_{\boldsymbol{\theta}}(\omega)$
and a logic formula φ over ω

$$p(\varphi) = \int_{\Omega} \mathbb{1}_{\omega \models \varphi} \, p_{\boldsymbol{\theta}}(\omega) \, \mathrm{d}\omega$$

Remember example



Neurosymbolic inference aggregates consistent worlds over a neural weight

$$\int \mathbb{1}_{\omega \models \varphi} \, p_{\theta}(\omega) \, d\omega$$

Neurosymbolic inference aggregates consistent worlds over a neural weight

$$\int \mathbb{1}_{\omega \models \varphi} \, p_{\boldsymbol{\theta}}(\omega) \, \mathrm{d}\omega$$

Exact in approximate

Restrict to simpler logic and belief

Neurosymbolic inference aggregates consistent worlds over a neural weight

$$\int \mathbb{1}_{\omega \models \varphi} \, p_{\theta}(\omega) \, \mathrm{d}\omega$$

Exact in approximate — Restrict to simpler logic and belief

Approximate in exact — Restrict aggregation to representative subset of domain

1. What is neurosymbolic AI and what does it promise?
2. Where does neurosymbolic AI struggle?
3. How can we still make neurosymbolic AI work?

1. What is neurosymbolic AI and what does it promise?
2. Where does neurosymbolic AI struggle?
3. How can we still make neurosymbolic AI work?
   1. Neurosymbolic models for sequential data
   2. Turning hallucinations into consistency and learning generalising models
   3. Constraining language models and safe reinforcement learning

Yes, we can make neurosymbolic AI work on challenging sequential data!

Problem Transformers are amazing at learning from sequential data but often hallucinate and struggle with consistency

Yes, we can make neurosymbolic AI work on challenging sequential data!

Problem Transformers are amazing at learning from sequential data
but often hallucinate and struggle with consistency

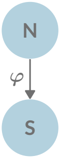
Challenge Existing methods for neurosymbolic inference
were not able to scale to sequential data

Yes, we can make neurosymbolic AI work on challenging sequential data!

Problem Transformers are amazing at learning from sequential data
but often hallucinate and struggle with consistency

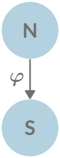
Challenge Existing methods for neurosymbolic inference
were not able to scale to sequential data

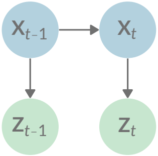
Solution Our neurosymbolic Markov models (NeSy-MMs) combine
sequential probabilistic models with symbolic logic



NeSy

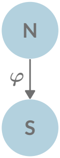
- ✗ Sequential
- ✓ Relational
- ✓ Discrete and continuous
- ✓ Neural + logical
- ✓ Discriminative and generative

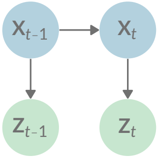


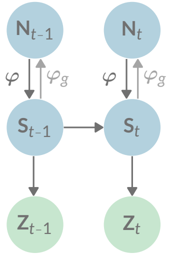


NeSy		HMM
✗	Sequential	✓
✓	Relational	✗
✓	Discrete and continuous	✓
✓	Neural + logical	✗
✓	Discriminative and generative	✓

Our solution



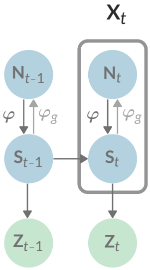




NeSy		HMM	NeSy-MMs
✗	Sequential	✓	✓
✓	Relational	✗	✓
✓	Discrete and continuous	✓	✓
✓	Neural + logical	✗	✓
✓	Discriminative and generative	✓	✓

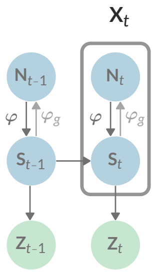
Particle filters scale sequential probabilistic inference in discrete-continuous domains

$$p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{Z}_{0:t+1}\right) \propto \int p_\varphi\left(\mathbf{Z}_{t+1} \mid \mathbf{X}_{t+1}\right) \cdot p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{x}_t\right) \cdot p_\varphi\left(\mathbf{x}_t \mid \mathbf{Z}_{0:t}\right) \mathrm{d}\mathbf{x}_t$$



Particle filters scale sequential probabilistic inference in discrete-continuous domains

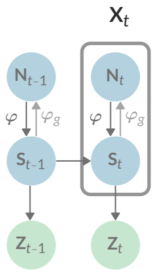
$$p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{Z}_{0:t+1}\right) \propto \int p_\varphi\left(\mathbf{Z}_{t+1} \mid \mathbf{X}_{t+1}\right) \cdot p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{x}_t\right) \cdot p_\varphi\left(\mathbf{x}_t \mid \mathbf{Z}_{0:t}\right) \, d\mathbf{x}_t$$



1 **Recursively** draw samples

Particle filters scale sequential probabilistic inference in discrete-continuous domains

$$p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{Z}_{0:t+1}\right) \propto \int p_\varphi\left(\mathbf{Z}_{t+1} \mid \mathbf{X}_{t+1}\right) \cdot p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{x}_t\right) \cdot p_\varphi\left(\mathbf{x}_t \mid \mathbf{Z}_{0:t}\right) \mathrm{d}\mathbf{x}_t$$

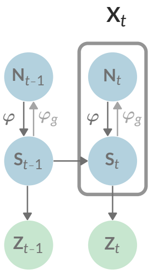


1 **Recursively** draw samples

2 **Transition** recursive samples

Particle filters scale sequential probabilistic inference in discrete-continuous domains

$$p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{Z}_{0:t+1}\right) \propto \int p_\varphi\left(\mathbf{Z}_{t+1} \mid \mathbf{X}_{t+1}\right) \cdot p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{x}_t\right) \cdot p_\varphi\left(\mathbf{x}_t \mid \mathbf{Z}_{0:t}\right) \, d\mathbf{x}_t$$



1. **Recursively** draw samples
2. **Transition** recursive samples
3. **Resample** transitioned samples with observation

Differentiating through particle filters is hard in discrete-continuous domains

$$p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{Z}_{0:t+1}\right) \propto \int \boxed{p_\varphi\left(\mathbf{Z}_{t+1} \mid \mathbf{X}_{t+1}\right)} \cdot p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{x}_t\right) \cdot p_\varphi\left(\mathbf{x}_t \mid \mathbf{Z}_{0:t}\right)\, \mathrm{d}\mathbf{x}_t$$

Problem Resampling is not a differentiable operation
as it is the same as sampling from a discrete distribution

Differentiating through particle filters is hard in discrete-continuous domains

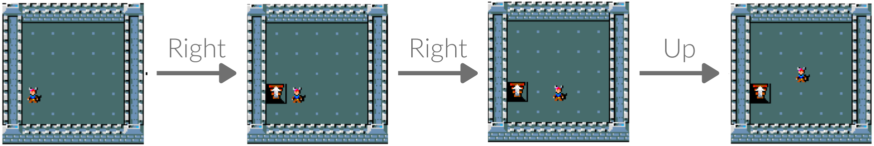
$$p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{Z}_{0:t+1}\right) \propto \int \boxed{p_\varphi\left(\mathbf{Z}_{t+1} \mid \mathbf{X}_{t+1}\right)} \cdot p_\varphi\left(\mathbf{X}_{t+1} \mid \mathbf{x}_t\right) \cdot p_\varphi\left(\mathbf{x}_t \mid \mathbf{Z}_{0:t}\right)\, \mathbf{dx}_t$$

Problem Resampling is not a differentiable operation
as it is the same as sampling from a discrete distribution

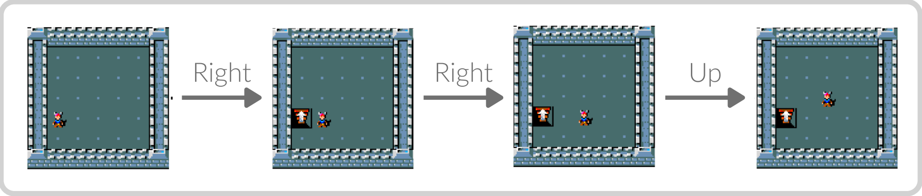
Solution Existing work tackles continuous resampling gradients
and we add a solution for discrete resampling gradients

1. What is neurosymbolic AI and what does it promise?
2. Where does neurosymbolic AI struggle?
3. How can we still make neurosymbolic AI work?
   1. Neurosymbolic models for sequantial data
   2. Turning hallucinations into consistency and learning generalising models
   3. Constraining language models and safe reinforcement learning

Generative data

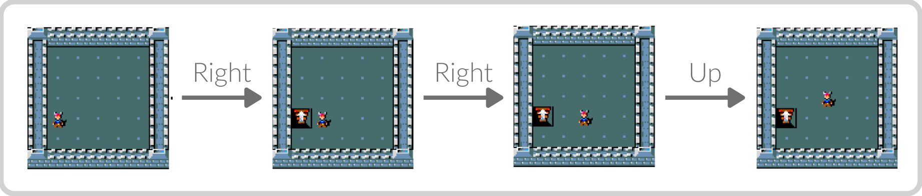


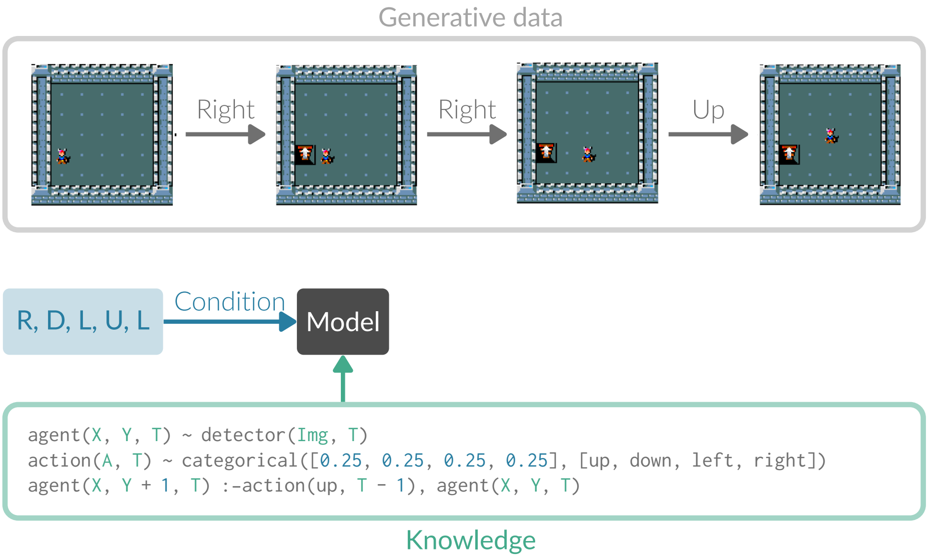
Generative data



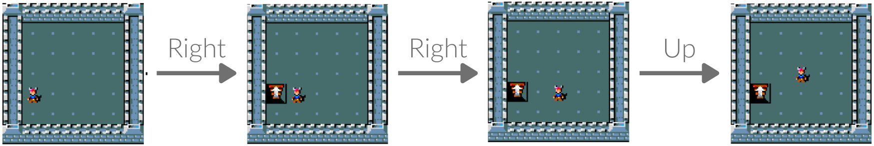
Model

Generative data




Generative data
Right
Right
Up
R, D, L, U, L
Condition
Model
agent(X, Y, T) ~ detector(Img, T)
action(A, T) ~ categorical([0.25, 0.25, 0.25, 0.25], [up, down, left, right])
agent(X, Y + 1, T) :-action(up, T - 1), agent(X, Y, T)
Knowledge

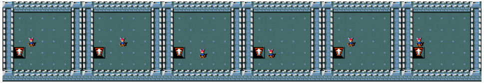
Generative data



R, D, L, U, L

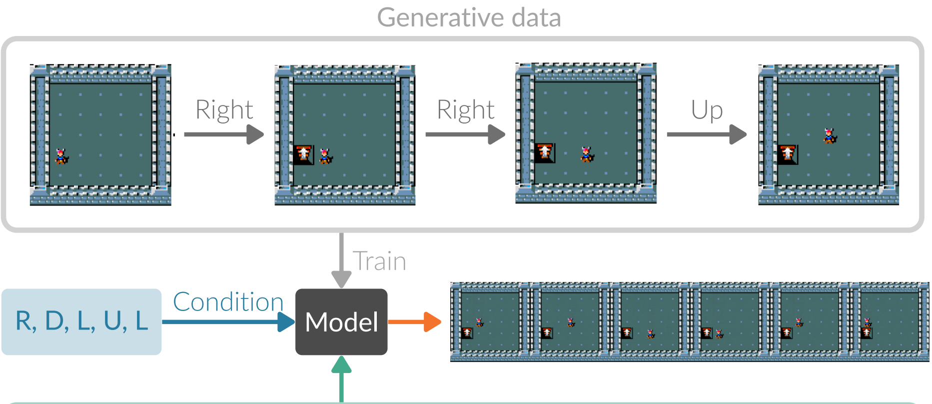
Condition

Model



```
agent(X, Y, T) ~ detector(Img, T)
action(A, T) ~ categorical([0.25, 0.25, 0.25, 0.25], [up, down, left, right])
agent(X, Y + 1, T) :-action(up, T - 1), agent(X, Y, T)
```

Knowledge




```
agent(X, Y, T) ~ detector(Img, T)
action(A, T) ~ categorical([0.25, 0.25, 0.25, 0.25], [up, down, left, right])
agent(X, Y + 1, T) :-action(up, T - 1), agent(X, Y, T)
```

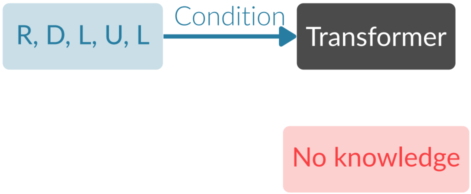
Knowledge

Transformers do not generate logically consistent images

Transformer

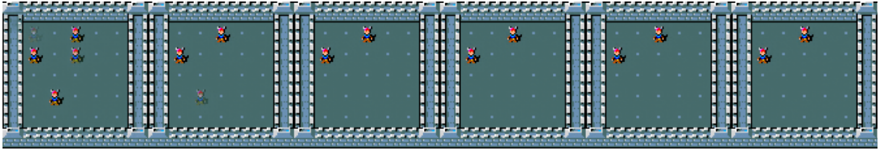
Transformers do not generate logically consistent images

Transformers do not generate logically consistent images



Transformers do not generate logically consistent images

Generation



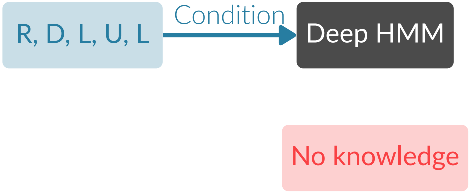
R, D, L, U, L

Condition

Transformer

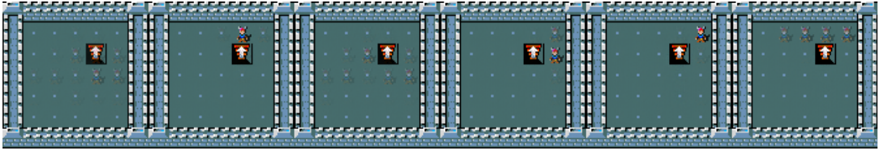
No knowledge

Deep HMMs do not generate logically consistent images



Deep HMMs do not generate logically consistent images

Generation



R, D, L, U, L

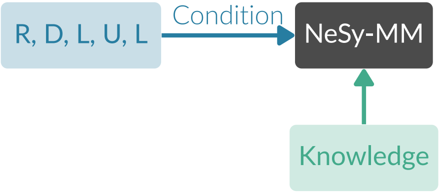
Condition

Deep HMM

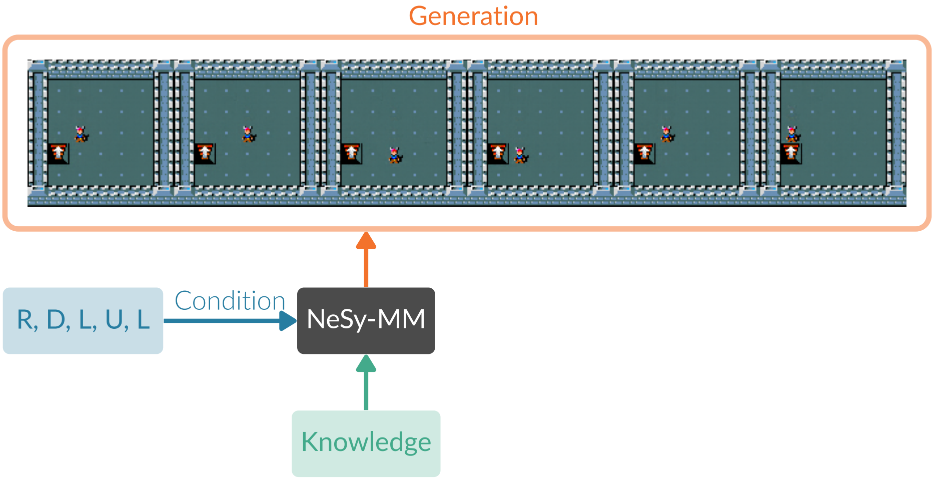
No knowledge

NeSy-MMs do generate logically consistent images

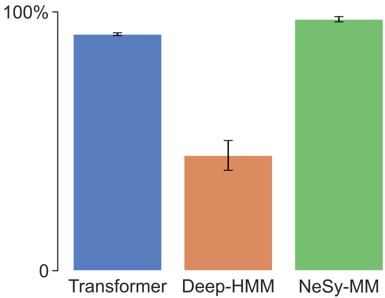
NeSy-MMs do generate logically consistent images



NeSy-MMs do generate logically consistent images

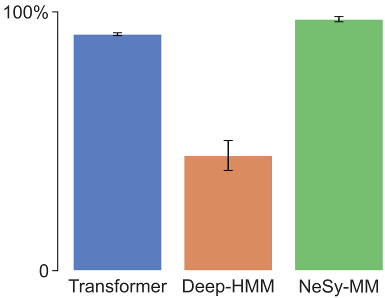


Quantifying logical consistency with reconstruction accuracy

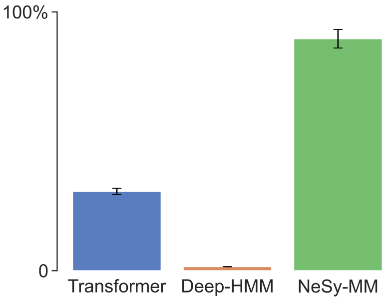


5×5

Quantifying logical consistency with reconstruction accuracy

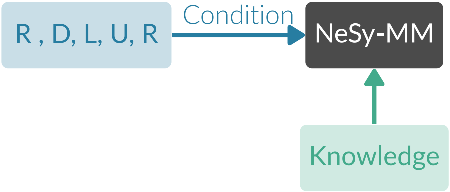


5×5

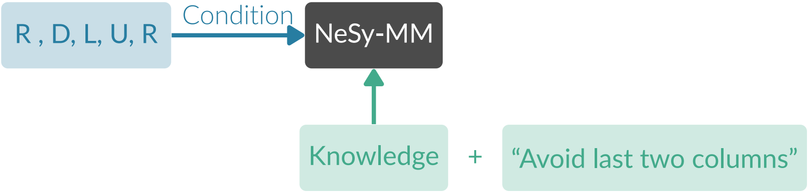


10×10

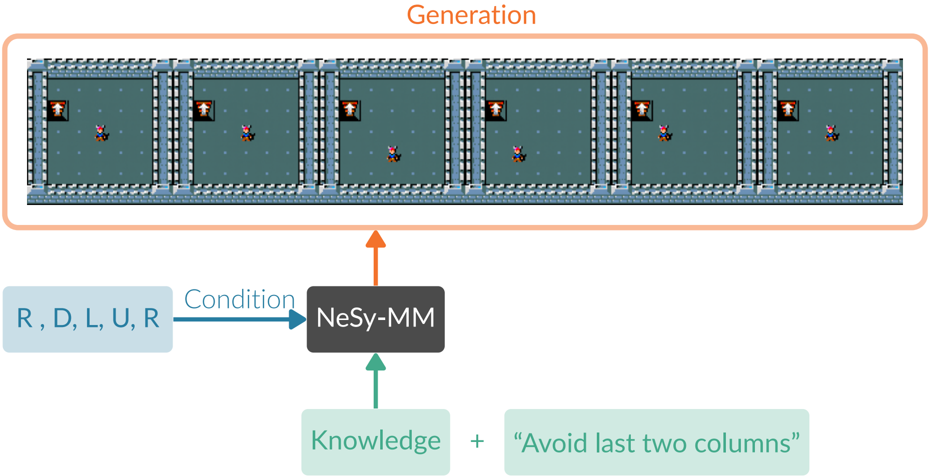
Knowledge can be changed at any time without having to retrain any models



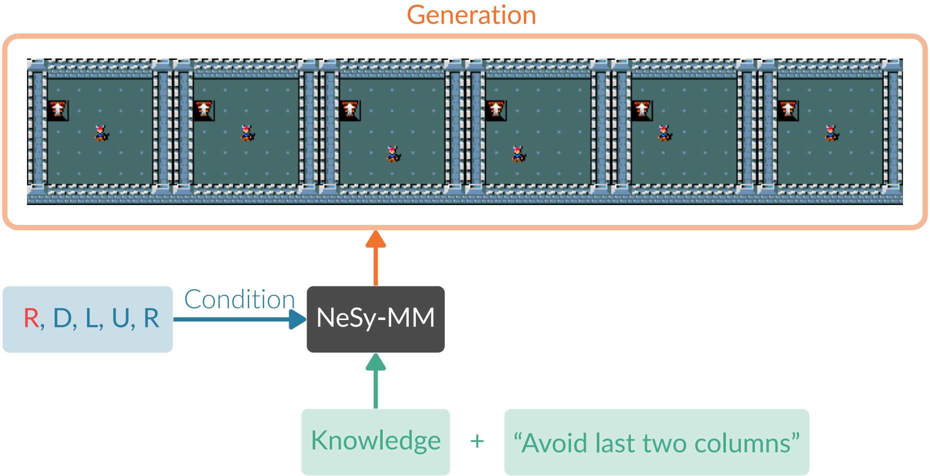
Knowledge can be changed at any time without having to retrain any models

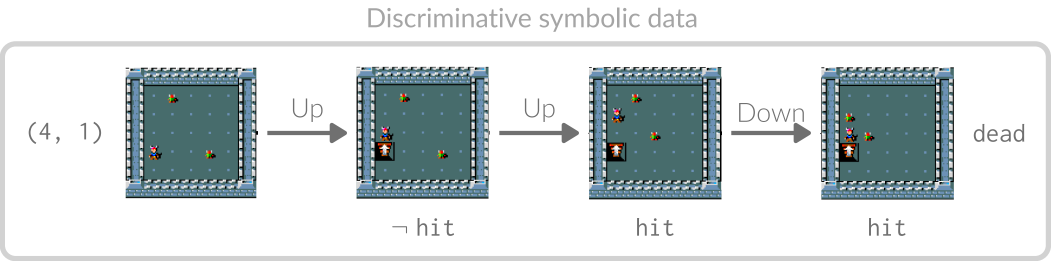


Knowledge can be changed at any time without having to retrain any models

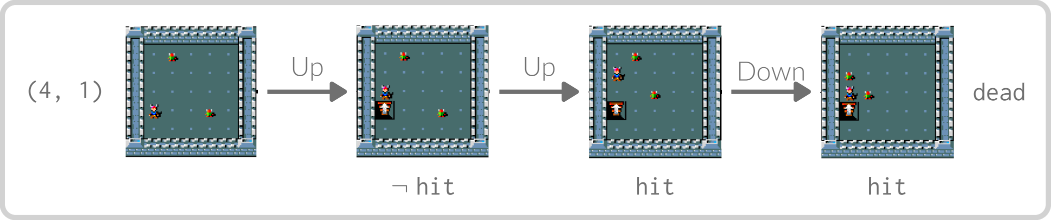


Knowledge can be changed at any time without having to retrain any models



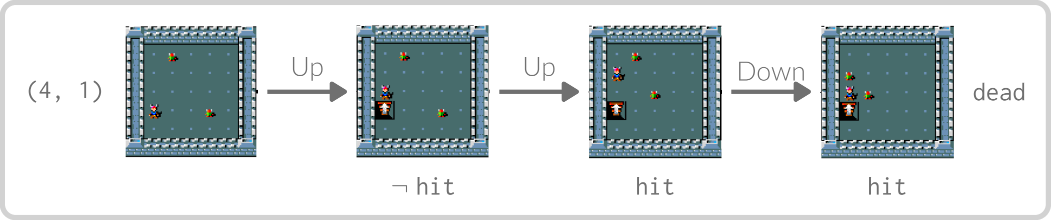

Discriminative symbolic data
(4, 1)
Up
Up
Down
dead
¬ hit
hit
hit

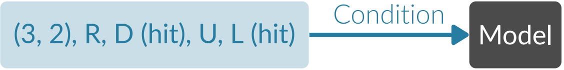
Discriminative symbolic data

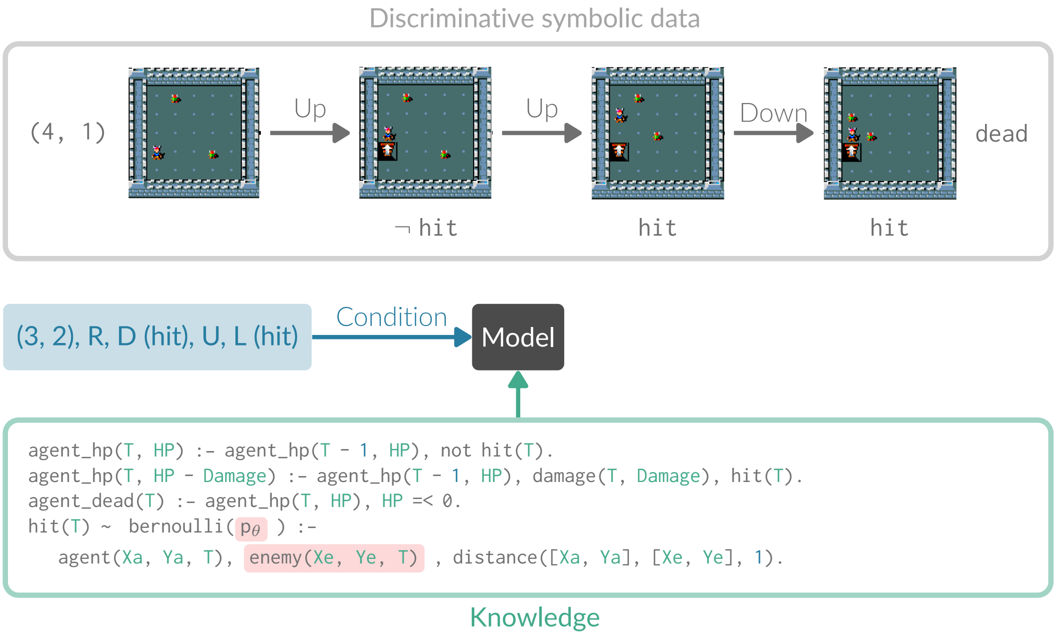


Model

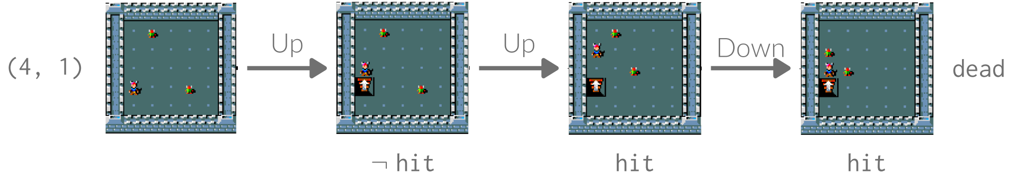
Discriminative symbolic data






Discriminative symbolic data
(4, 1)
Up
Up
Down
dead
¬ hit
hit
hit
(3, 2), R, D (hit), U, L (hit)
Condition
Model
agent_hp(T, HP) :- agent_hp(T - 1, HP), not hit(T).
agent_hp(T, HP - Damage) :- agent_hp(T - 1, HP), damage(T, Damage), hit(T).
agent_dead(T) :- agent_hp(T, HP), HP =< 0.
hit(T) ~ bernoulli(pθ) :-
agent(Xa, Ya, T), enemy(Xe, Ye, T) , distance([Xa, Ya], [Xe, Ye], 1).
Knowledge

Discriminative symbolic data

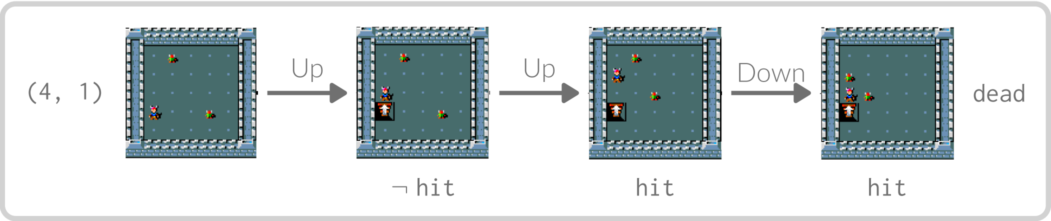


(3, 2), R, D (hit), U, L (hit) —Condition→ **Model** —Predict→ dead/alive

```
agent_hp(T, HP) :- agent_hp(T - 1, HP), not hit(T).
agent_hp(T, HP - Damage) :- agent_hp(T - 1, HP), damage(T, Damage), hit(T).
agent_dead(T) :- agent_hp(T, HP), HP =< 0.
hit(T) ~ bernoulli( pθ ) :-
   agent(Xa, Ya, T), enemy(Xe, Ye, T) , distance([Xa, Ya], [Xe, Ye], 1).
```

Knowledge

Discriminative symbolic data



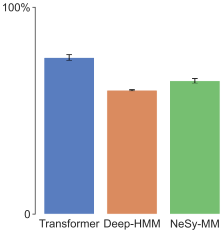
Train

(3, 2), R, D (hit), U, L (hit) —Condition→ **Model** —Predict→ dead/alive

```
agent_hp(T, HP) :- agent_hp(T - 1, HP), not hit(T).
agent_hp(T, HP - Damage) :- agent_hp(T - 1, HP), damage(T, Damage), hit(T).
agent_dead(T) :- agent_hp(T, HP), HP =< 0.
hit(T) ~ bernoulli( pθ ) :-
   agent(Xa, Ya, T), enemy(Xe, Ye, T) , distance([Xa, Ya], [Xe, Ye], 1).
```

Knowledge

Which model learns a generalisable representation?

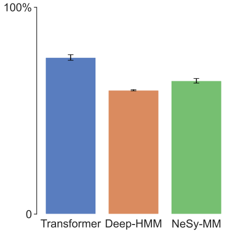


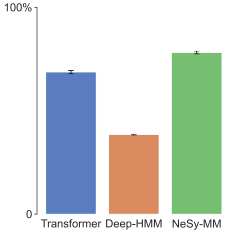
10×10

10 steps

1 enemy

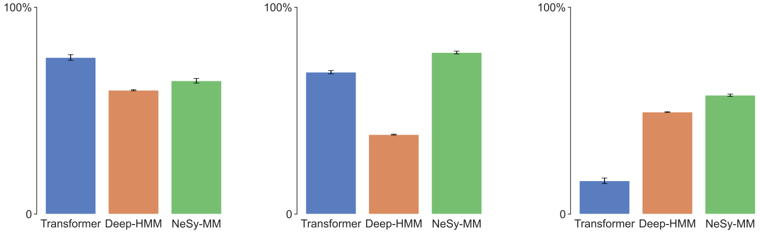
Which model learns a generalisable representation?





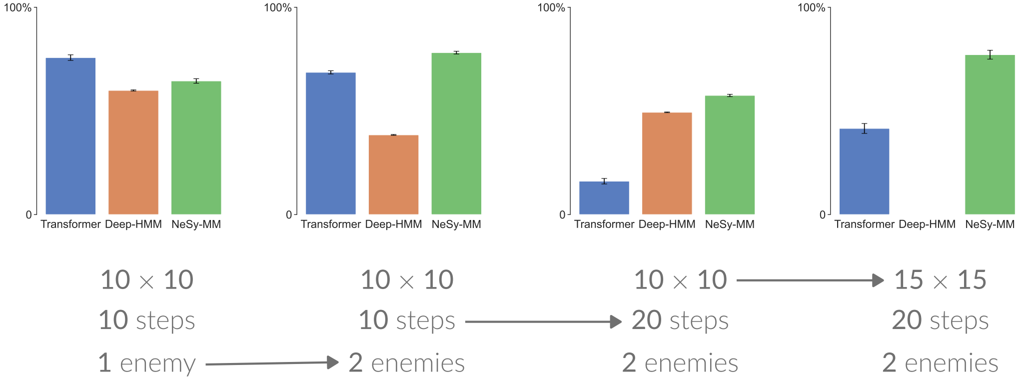
10×10	10×10
10 steps	10 steps
1 enemy →	2 enemies

Which model learns a generalisable representation?



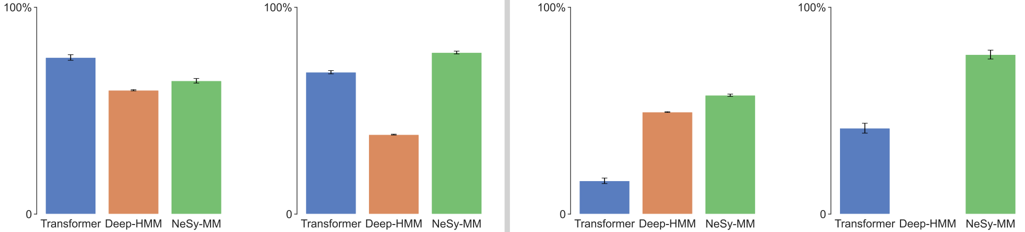
10×10	10×10	10×10
10 steps	10 steps →	20 steps
1 enemy →	2 enemies	2 enemies

Which model learns a generalisable representation?



Which model learns a generalisable representation?

Non-trivial domains



10 × 10	10 × 10	10 × 10	15 × 15
10 steps	10 steps	20 steps	20 steps
1 enemy	2 enemies	2 enemies	2 enemies

1. What is neurosymbolic AI and what does it promise?
2. Where does neurosymbolic AI struggle?
3. **How can we still make neurosymbolic AI work?**
   1. Neurosymbolic models for sequantial data
   2. Turning hallucinations into consistency and learning generalising models
   3. **Constraining language models and safe reinforcement learning**

Safe language generation is important but hard guarantee

Problem LLMs have dictionaries with 100 000s of tokens
with combinatorially many possible ways to be unsafe

Safe language generation is important but hard guarantee

Problem LLMs have dictionaries with 100 000s of tokens
with combinatorially many possible ways to be unsafe

Questions Is training for safety enough?

Safe language generation is important but hard guarantee

Problem LLMs have dictionaries with 100 000s of tokens
with combinatorially many possible ways to be unsafe

Questions Is training for safety enough?

Can we change constraints without retraining?

Safe language generation is important but hard guarantee

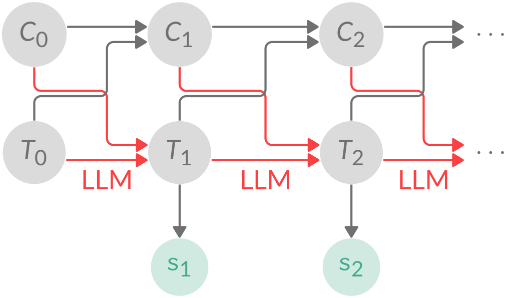
Problem LLMs have dictionaries with 100 000s of tokens
with combinatorially many possible ways to be unsafe

Questions Is training for safety enough?

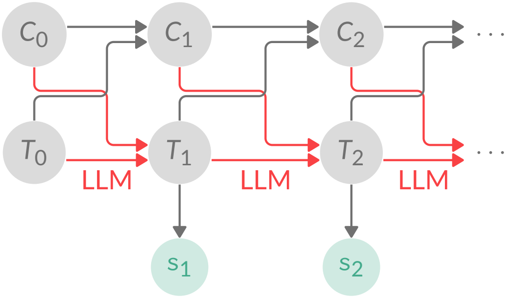
Can we change constraints without retraining?

Does changing predictions influence performance?

Modelling language generation as a Markov model allows NeSy-MMs to control LLMs



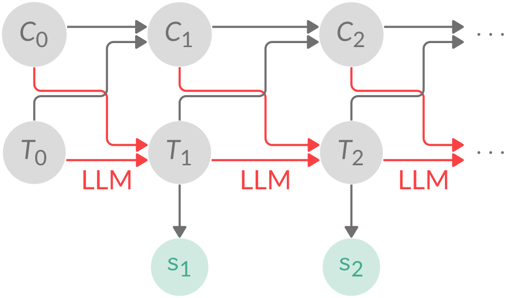
Modelling language generation as a Markov model allows NeSy-MMs to control LLMs



"Token 3 given safety over next $\boldsymbol{H}$ tokens"

$$p_{LLM}(T_3 \mid C_2, T_2, s_{3:3+H})$$

Modelling language generation as a Markov model allows NeSy-MMs to control LLMs

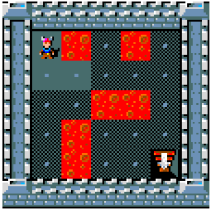


"Token 3 given safety over next H tokens"

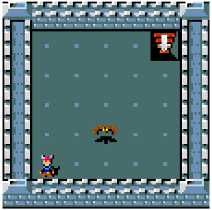
$$p_{LLM}(T_3 \mid C_2, T_2, s_{3:3+H})$$

NeSy-MMs approximate safe generative distribution
while allowing for programming of different contraints

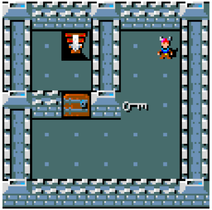
Agents should behave safely during exploration and training



Don't go right
into the lava



Don't go towards
the monster



Get a key
before opening
a locked door

Modelling policies as Nesy-MMS makes safe RL agents possible

Previously in our group Program safety specification in probabilistic logic and use conditioned policy for learning and inference

Modelling policies as Nesy-MMS makes safe RL agents possible

Previously in our group	Program safety specification in probabilistic logic and use conditioned policy for learning and inference
Limitation	Exact inference is too expensive for large domains and only looks ahead one step

Modelling policies as Nesy-MMS makes safe RL agents possible

Previously in our group	Program safety specification in probabilistic logic and use conditioned policy for learning and inference
Limitation	Exact inference is too expensive for large domains and only looks ahead one step
Now	NeSy-MMs encode the conditioned policies allowing safe policies in large temporal domains

Neurosymbolic AI is one avenue to consistent AI but there is still much work to do

Potential	Neurosymbolic AI models can learn from data and guarantee logical consistency

Neurosymbolic AI is one avenue to consistent AI but there is still much work to do

Potential	Neurosymbolic AI models can learn from data and guarantee logical consistency
Scalability	Neurosymbolic AI models struggle with large domains and require a lot of data to learn

Neurosymbolic AI is one avenue to consistent AI but there is still much work to do

Potential	Neurosymbolic AI models can learn from data and guarantee logical consistency
Scalability	Neurosymbolic AI models struggle with large domains and require a lot of data to learn
Solutions exist	Large sequential domains become viable by exploiting approximate inference techniques

Neurosymbolic AI models will deliver your pizza
hot and safely to your door

Visit personal page

Read our paper