An Introduction to Abductive Learning

Integrating Machine Learning and Logical Reasoning

(Press ? for help, n and p for next and previous slide)

Wang-Zhou Dai

School of Intelligence Science & Technology,
Nanjing University

November 28th, 2025

https://daiwz.net

Learning and Reasoning

Data-Driven AI

Image Net (1.2M images)

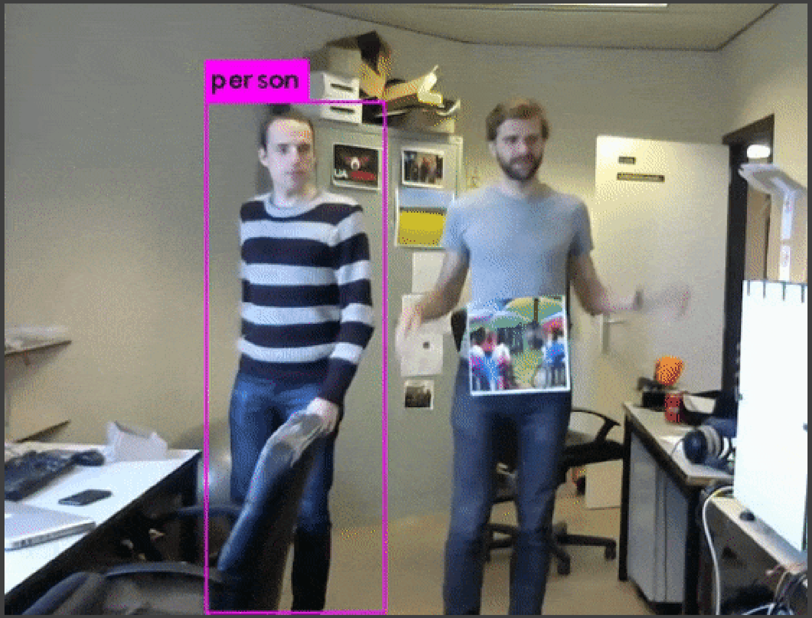
GPT-5 (≥20 trillion tokens)

Data-Driven AI

The more **data**, the better performance!

Heavy Learning / Light Reasoning

Lack of robustness and soundness.



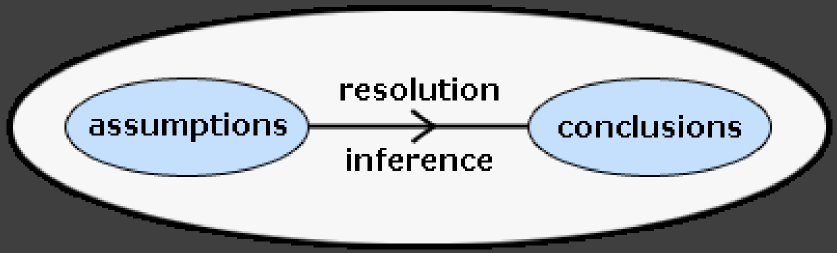
Question: How many words are in your response to this prompt?

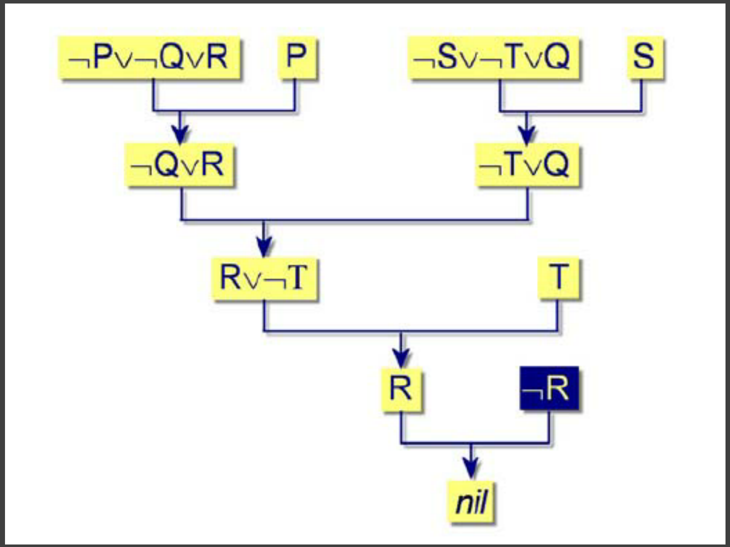
GPT4o: There are 12 words in my response to this prompt.
DeepSeek-V3: This response contains 10 words. (Note the count includes every individual word in this message.)

o3-mini-high (18s): Determining an exact word count is a bit of a self-referential trap. If I include a count in my answer, that very act would change the total number of words. In other words, any fixed number I offer would instantly become inaccurate as soon as it’s stated.
DeepSeek-R1 (232s): There are 15 words in this response.

Knowledge-Driven AI

- **Problem solving and Knowledge Engineering/Induction**
  - Automated Theorem Proving, 1955
  - General Problem Solver, 1957
  - Expert Systems, 1960s
  - Logic Programming, 1970s
  - Symbolic Learning, 1980s



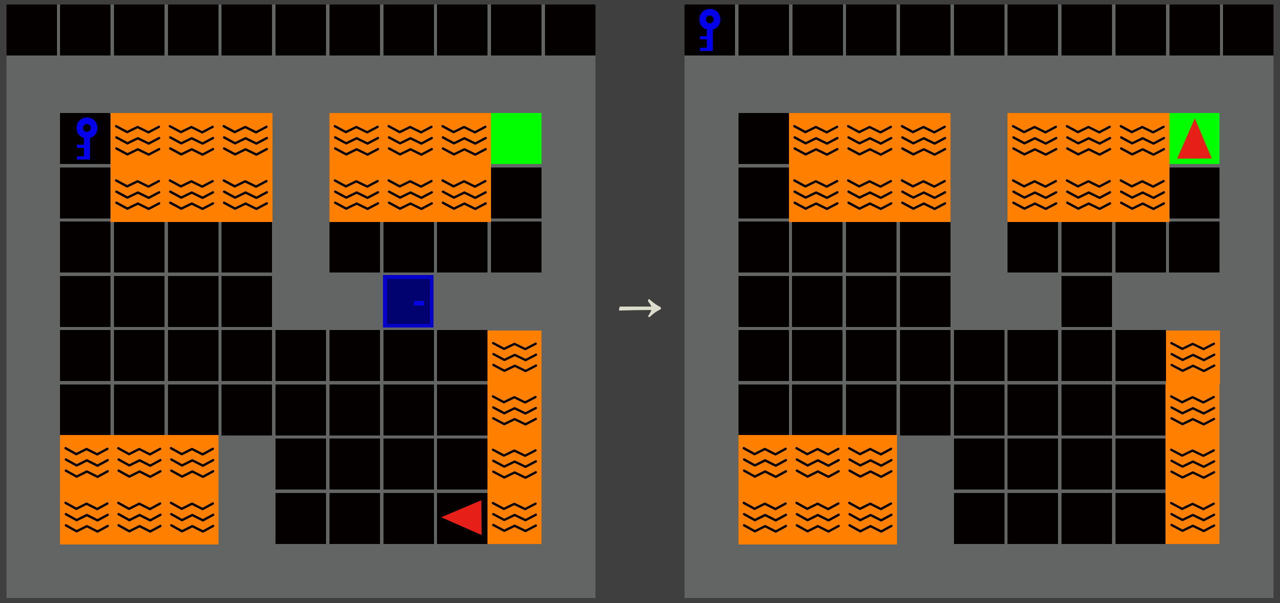


Knowledge-Driven AI

The more **knowledge**, the better performance!

Heavy Reasoning / Light Learning

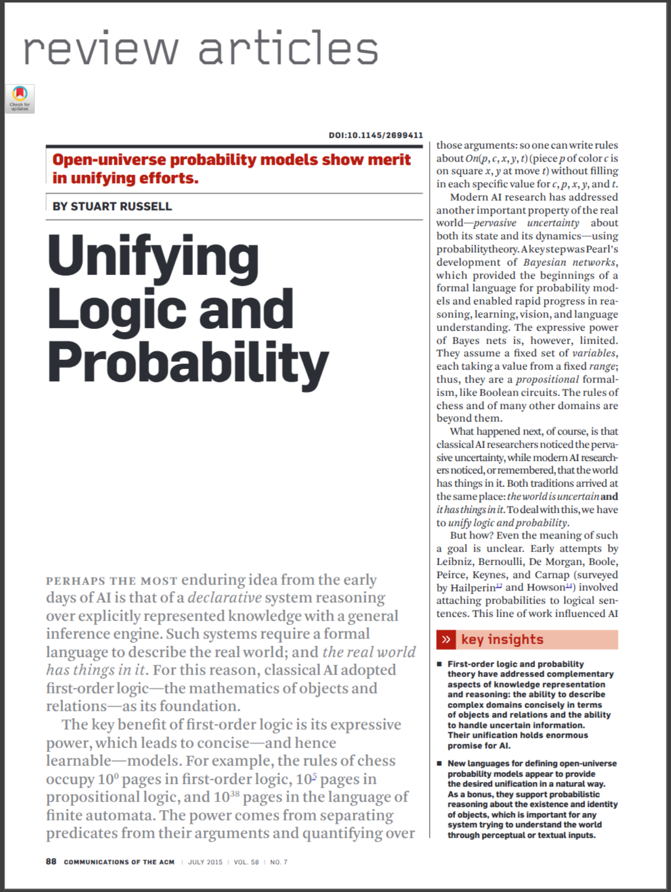
Cannot solve problems involving sub-symbolic inputs:



The representation gap!

> … these approaches miss an important consequence of uncertainty in a world of things: *uncertainty about what things are in the world*. Real objects seldom wear unique identifiers or preannounce their existence like the cast of a play. In areas such as vision, language understanding, … , the existence of objects *must be inferred from raw data* (pixels, strings, and so on) that contain no explicit object references.

[Russell, 2015]



review articles

DOI:10.1145/2699411

Open-universe probability models show merit in unifying efforts.

BY STUART RUSSELL

Unifying Logic and Probability

PERHAPS THE MOST enduring idea from the early days of AI is that of a *declarative* system reasoning over explicitly represented knowledge with a general inference engine. Such systems require a formal language to describe the real world; and *the real world has things in it*. For this reason, classical AI adopted first-order logic—the mathematics of objects and relations—as its foundation.

The key benefit of first-order logic is its expressive power, which leads to concise—and hence learnable—models. For example, the rules of chess occupy 10^0 pages in first-order logic, 10^5 pages in propositional logic, and 10^{38} pages in the language of finite automata. The power comes from separating predicates from their arguments and quantifying over those arguments: so one can write rules about $On(p, c, x, y, t)$ (piece p of color c is on square x, y at move t) without filling in each specific value for c, p, x, y, and t.

Modern AI research has addressed another important property of the real world—*pervasive uncertainty* about both its state and its dynamics—using probability theory. A key step was Pearl's development of *Bayesian networks*, which provided the beginnings of a formal language for probability models and enabled rapid progress in reasoning, learning, vision, and language understanding. The expressive power of Bayes nets is, however, limited. They assume a fixed set of *variables*, each taking a value from a fixed *range*; thus, they are a *propositional* formalism, like Boolean circuits. The rules of chess and of many other domains are beyond them.

What happened next, of course, is that classical AI researchers noticed the pervasive uncertainty, while modern AI researchers noticed, or remembered, that the world has things in it. Both traditions arrived at the same place: *the world is uncertain* **and** *it has things in it*. To deal with this, we have to *unify logic and probability*.

But how? Even the meaning of such a goal is unclear. Early attempts by Leibniz, Bernoulli, De Morgan, Boole, Peirce, Keynes, and Carnap (surveyed by Hailperin[12] and Howson[14]) involved attaching probabilities to logical sentences. This line of work influenced AI

» key insights

- **First-order logic and probability theory have addressed complementary aspects of knowledge representation and reasoning: the ability to describe complex domains concisely in terms of objects and relations and the ability to handle uncertain information. Their unification holds enormous promise for AI.**
- **New languages for defining open-universe probability models appear to provide the desired unification in a natural way. As a bonus, they support probabilistic reasoning about the existence and identity of objects, which is important for any system trying to understand the world through perceptual or textual inputs.**

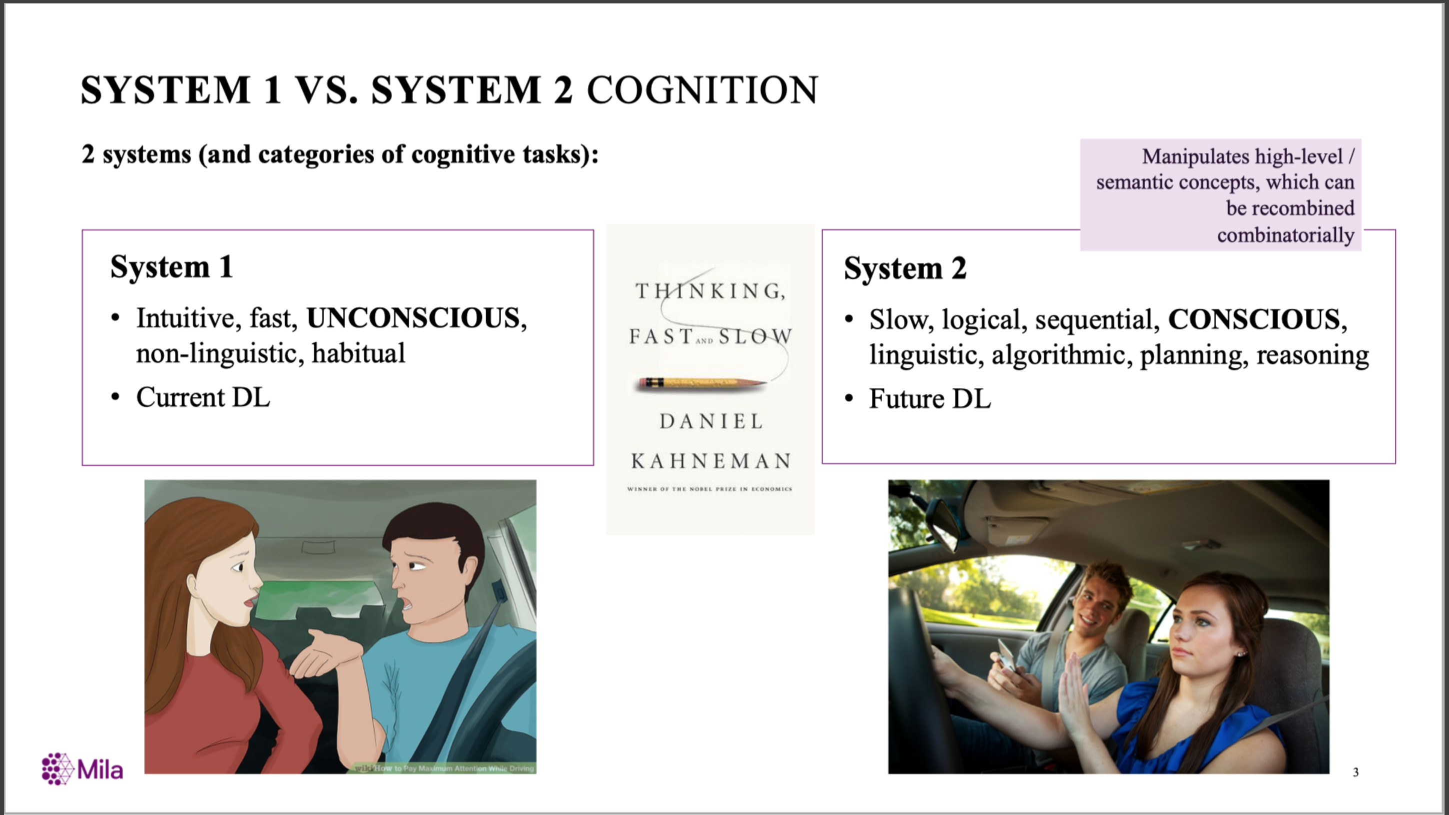
88 COMMUNICATIONS OF THE ACM | JULY 2015 | VOL. 58 | NO. 7

Comparison

Representation	Statistical / Neural	Symbolic
Data	Sensory / Raw / Vector	Term / Program
Hypotheses	First-order / functions	First & higher-order relations
Explainability	Difficult	Possible
Knowledge transfer	Difficult	More difficult
Noise Tolerance	Easy	Difficult
Inference Speed	Fast	Slow
Examples for Learning	Many	Few
Prior Knowledge for Learning	Few	Many

Combining The Two Systems

Yoshua Bengio: From System 1 Deep Learning to System 2 Deep Learning.



SYSTEM 1 VS. SYSTEM 2 COGNITION

2 systems (and categories of cognitive tasks):

Manipulates high-level / semantic concepts, which can be recombined combinatorially

System 1

- Intuitive, fast, **UNCONSCIOUS**, non-linguistic, habitual
- Current DL

System 2

- Slow, logical, sequential, **CONSCIOUS**, linguistic, algorithmic, planning, reasoning
- Future DL

THINKING, FAST AND SLOW
DANIEL KAHNEMAN
WINNER OF THE NOBEL PRIZE IN ECONOMICS
wikiHow to Pay Maximum Attention While Driving
Mila

3

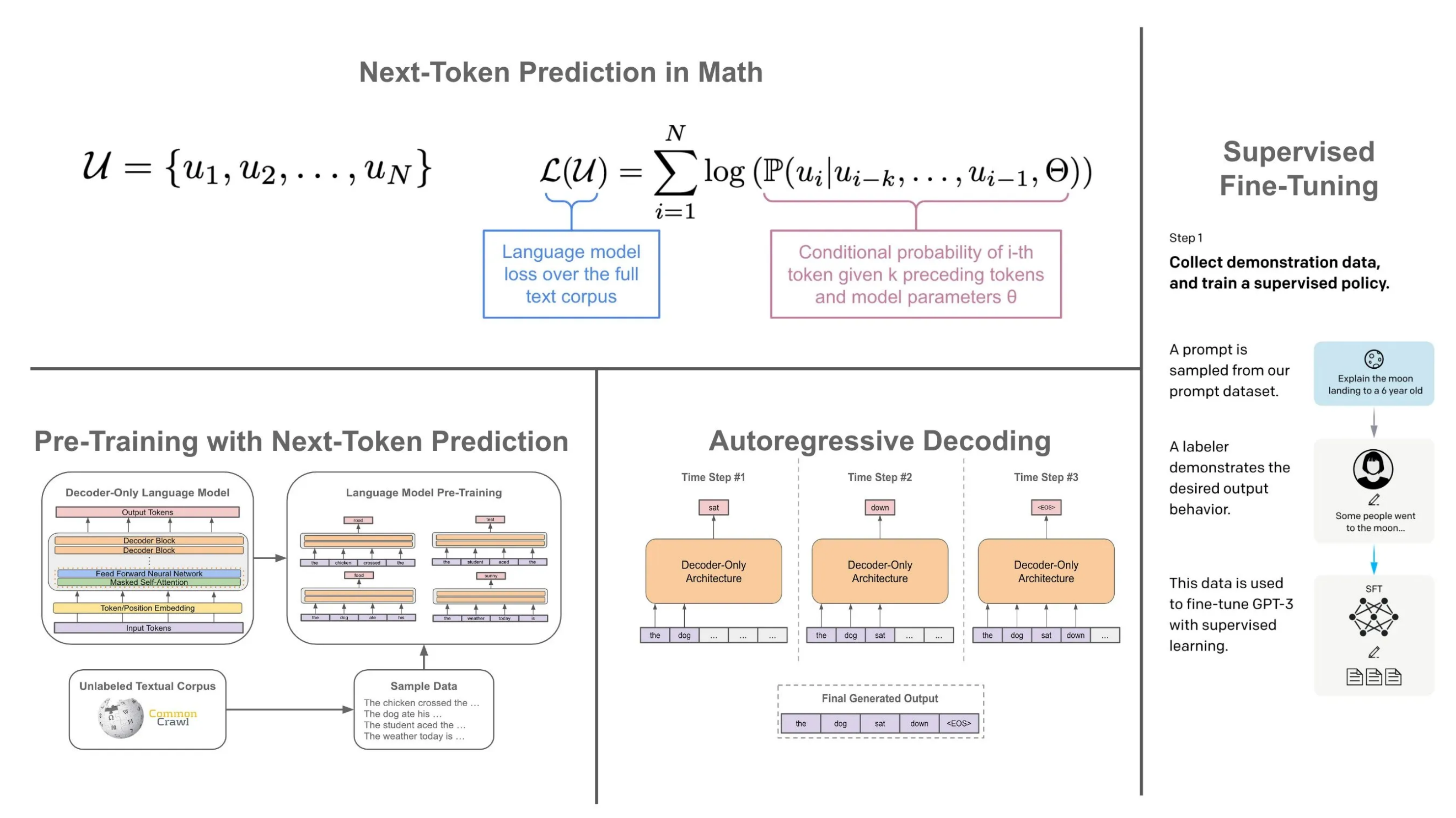
(NeurIPS'2019 Keynote)

Combining The Two Systems

1. **[End2end]** Approximate logic inference with continuous functions (fuzzy operators, semantic loss, etc.):
   - e.g., Logic Tensor Network [Badreddine et. al., 2021], Neural Theorem Prover [Rocktäschel et. al., 2017], Neural Logic Machines [Dong et. al., 2019], ∂ILP [Evans et. al., 2018] etc.
2. **[Hybrid]** Probabilistic models
   - e.g., DeepProbLog [Manhaeve et. al., 2018], DeepStochLog [Winters et. al., 2021], etc.
3. **[Hybrid]** Pure (classic) logic
   - e.g., Abductive Learning (ABL) [Dai et. al., 2019], NeurASP [Yang et. al., 2020], NEUROLOG Tsamoura et. al., 2021, etc.
4. **[End2end]** Reasoning LLMs
   - e.g., CoT, ToT, etc.
   - e.g., Open AI o series, DeepSeek-R1, etc.
5. **[Hybrid]** LLM Agents (workflow, tool using, etc.)

Yet another end2end model

“Next-token prediction”



Next-Token Prediction in Math

$$\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$$

$$\underbrace{\mathcal{L}(\mathcal{U})}_{\text{Language model loss over the full text corpus}} = \sum_{i=1}^{N} \log \big(\underbrace{\mathbb{P}(u_i | u_{i-k}, \ldots, u_{i-1}, \Theta)}_{\text{Conditional probability of i-th token given k preceding tokens and model parameters θ}}\big)$$

Pre-Training with Next-Token Prediction

The chicken crossed the …
The dog ate his …
The student aced the …
The weather today is …

Autoregressive Decoding

Supervised Fine-Tuning

Step 1

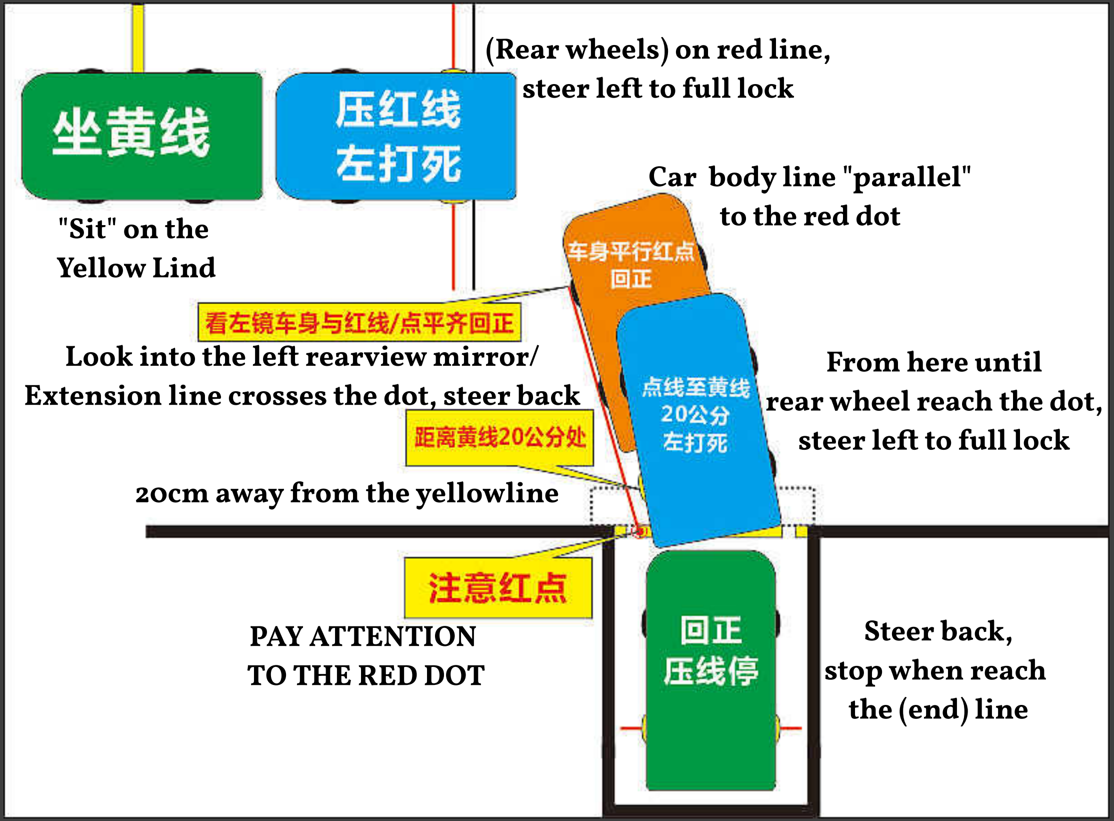
Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Formal Reasoning is useful in Learning



(Rear wheels) on red line, steer left to full lock

"Sit" on the Yellow Lind

Car body line "parallel" to the red dot

看左镜车身与红线/点平齐回正

Look into the left rearview mirror/ Extension line crosses the dot, steer back

From here until rear wheel reach the dot, steer left to full lock

距离黄线20公分处

20cm away from the yellowline

注意红点

PAY ATTENTION TO THE RED DOT

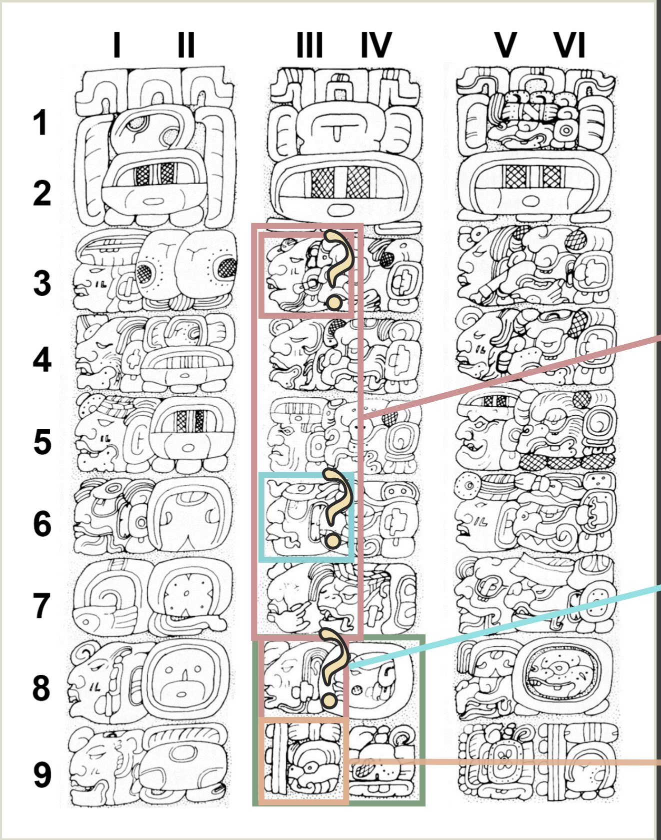
Steer back, stop when reach the (end) line

Knowledge turning into "muscle memory"

credit: bilibili.com

The Interface: Abductive Reasoning

For example ...



Three Mayan Calandar Systems

Creation + LC = Tz,Hb

Long Count (玛雅长历) date:

?. 18 . 5 . ? . 0

Tzolk'in (玛雅神历) date:

? Ahau

Haab'(玛雅太阳历) date:

13 Mac

The Interface: Groundings

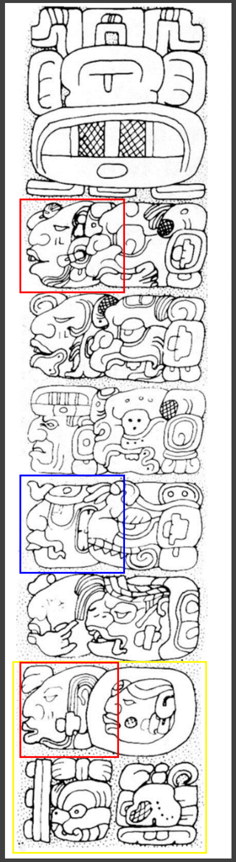
Image → `Numbers` → *Equation*

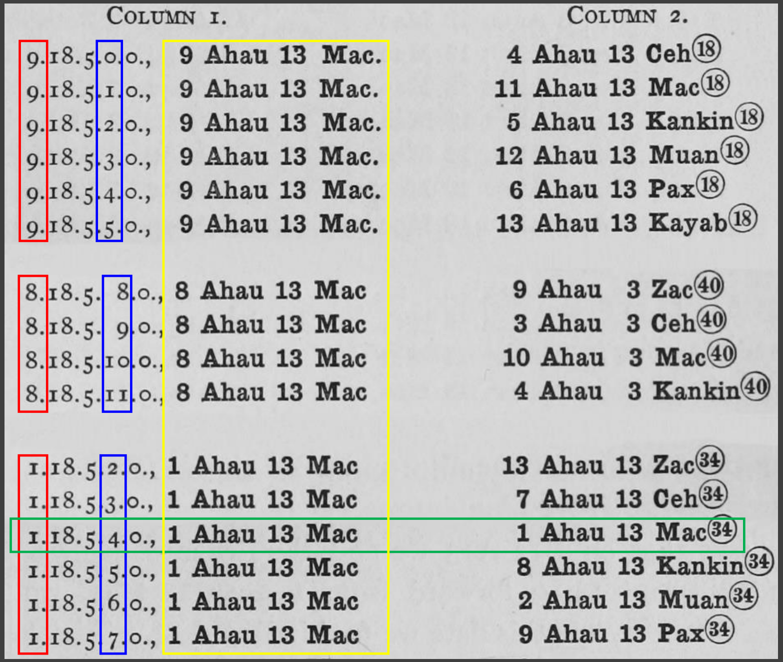
The Interface: Groundings

Image → `Numbers` → *Equation*

Perception → `Groundings` → *Reasoning*

Cracking the glyphs

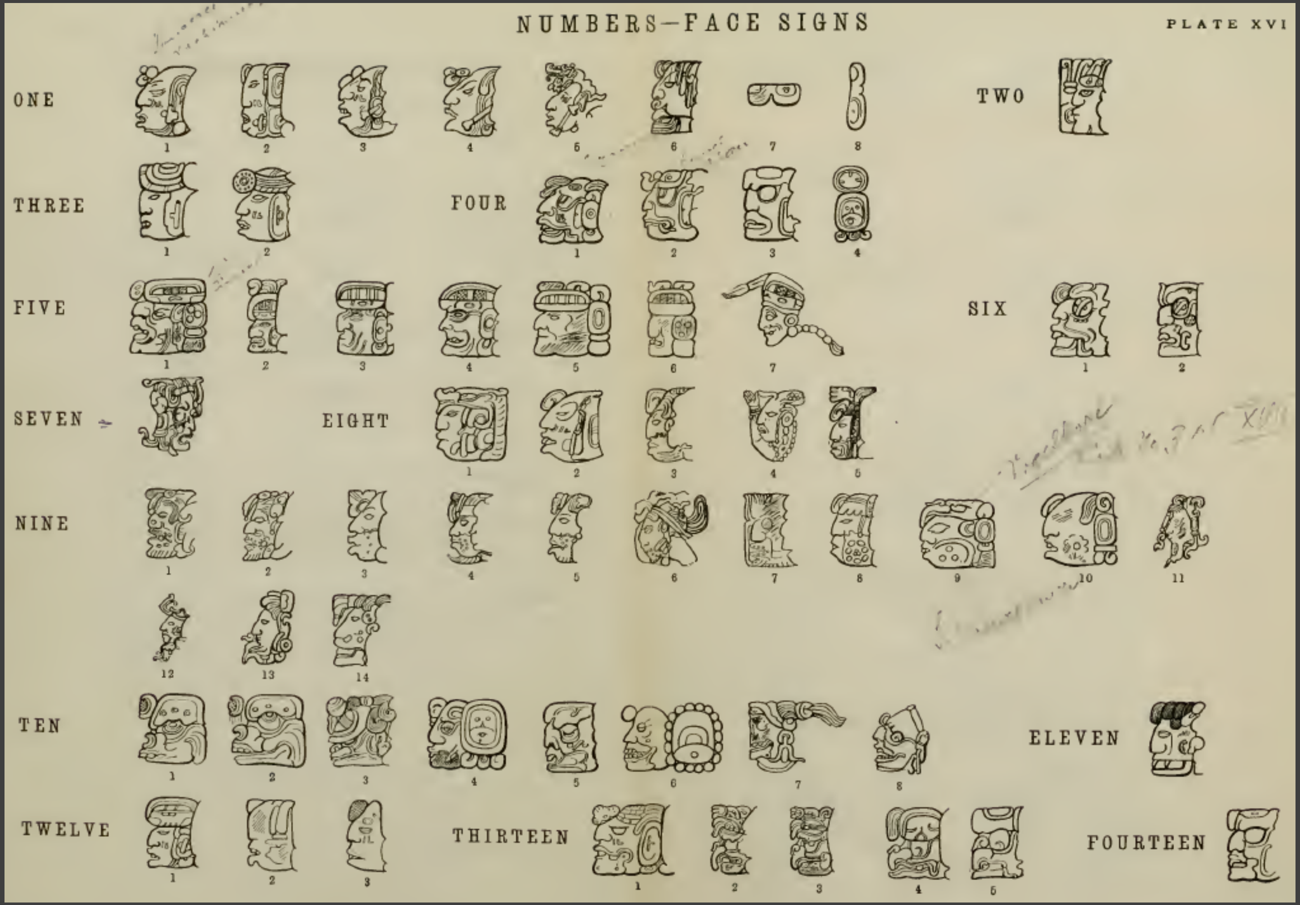




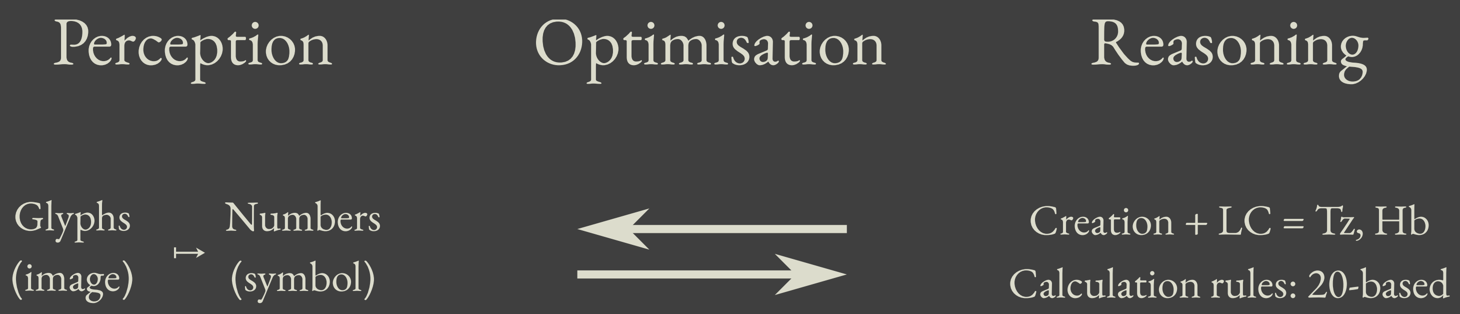
Column I.		Column 2.
9.18.5.0.0.,	9 Ahau 13 Mac.	4 Ahau 13 Ceh⑱
9.18.5.1.0.,	9 Ahau 13 Mac.	11 Ahau 13 Mac⑱
9.18.5.2.0.,	9 Ahau 13 Mac.	5 Ahau 13 Kankin⑱
9.18.5.3.0.,	9 Ahau 13 Mac.	12 Ahau 13 Muan⑱
9.18.5.4.0.,	9 Ahau 13 Mac.	6 Ahau 13 Pax⑱
9.18.5.5.0.,	9 Ahau 13 Mac.	13 Ahau 13 Kayab⑱
8.18.5. 8.0.,	8 Ahau 13 Mac	9 Ahau 3 Zac㊵
8.18.5. 9.0.,	8 Ahau 13 Mac	3 Ahau 3 Ceh㊵
8.18.5.10.0.,	8 Ahau 13 Mac	10 Ahau 3 Mac㊵
8.18.5.11.0.,	8 Ahau 13 Mac	4 Ahau 3 Kankin㊵
1.18.5.2.0.,	1 Ahau 13 Mac	13 Ahau 13 Zac㉞
1.18.5.3.0.,	1 Ahau 13 Mac	7 Ahau 13 Ceh㉞
1.18.5.4.0.,	1 Ahau 13 Mac	1 Ahau 13 Mac㉞
1.18.5.5.0.,	1 Ahau 13 Mac	8 Ahau 13 Kankin㉞
1.18.5.6.0.,	1 Ahau 13 Mac	2 Ahau 13 Muan㉞
1.18.5.7.0.,	1 Ahau 13 Mac	9 Ahau 13 Pax㉞

[Bowditch, 1901]

The cracked ground mapping



Abductive Reasoning



Perception

Optimisation

Reasoning

Glyphs (image) ↦ Numbers (symbol)

Creation + LC = Tz, Hb

Calculation rules: 20-based

"Equations should be correct"

> Starts from **incomplete** observations, proceeds to the likeliest possible **explanation** for them.

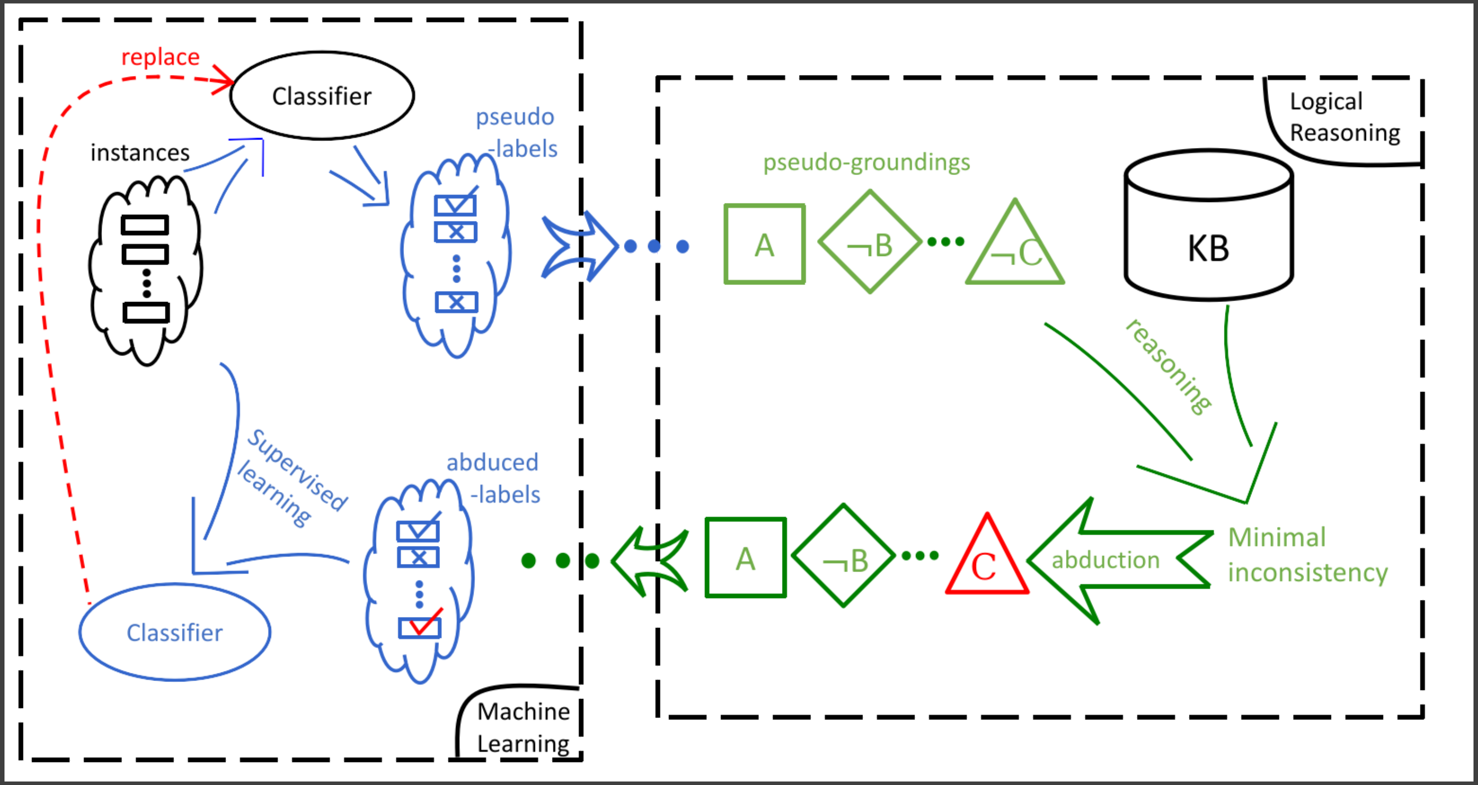
Burglary Network

(due to J. Pearl)

$$\begin{aligned} Alarm &\leftarrow Burglary \\ Alarm &\leftarrow Earthquake \\ JohnCalls &\leftarrow Alarm \\ MaryCalls &\leftarrow Alarm \end{aligned}$$

- **Observation**: $JohnCalls, \neg MaryCalls$
- **Explaination**:?

The Abductive Learning (ABL) Framework



Zhi-Hua Zhou. Abductive learning: Towards bridging machine learning and logical reasoning. In: Science China Information Sciences, 2019, 62: 076101.

Abductive Learning (ABL)

Training examples: $\langle x, y \rangle$.

`sum([` `,` `,` `], 15).`

1. **Machine learning** (e.g., neural net):

$$z = f(x;\theta) = \text{Sigmoid}(P_\theta(z|x))$$

   - Learns a perception model mapping **raw data** $(x) \rightarrow$ **logic facts** (z);

2. **Logical Reasoning** (e.g., logic program):

$$B \cup z \models y$$

   - **Inference (Deduction)**: $x \xrightarrow{f} z \xrightarrow{B} y$
   - **Learning (Abduction)**: $y \xrightarrow{B} z \xrightarrow{x} f$;

3. **Optimisation**: maximises the *consistency* of z and f w.r.t. $\langle x, y \rangle$ and B.

Some related papers

1. The Abductive Learning Framework [NeurIPS 2019] [NeSy:SOTA 2022]
2. Joint optimisation for better abduction:
   - » Exploiting similarity of symbols' raw representations [NeurIPS 2021]
   - » Abduction using ground knowledge base [IJCAI 2021]
   - » Learning to prune the symbolic search space [AAAI 2025]
3. Incomplete knowledge base:
   - » Combining second-order abduction with first-order abduction [IJCAI 2021]
   - » Utilizing external knowledge graph [IJCAI 2023]
   - » Discovering undefined concepts [AAAI 2023]
4. Learnability:
   - » Abduction with formal knowledge helps even when there is no pre-train [AAAI 2024]
5. Applications:
   - » Law documents analysis [ICDM 2020]
   - » Chinese historical manuscript recognition [AAAI 2024]

daiwz.net/publications

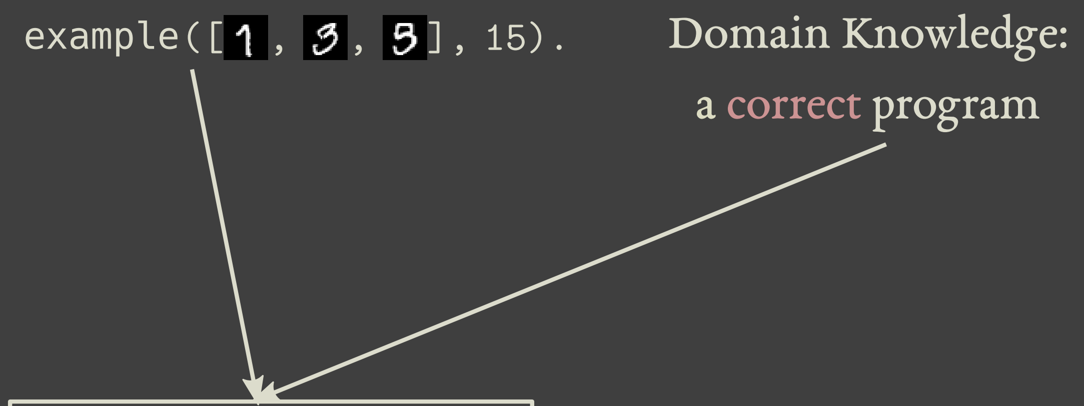
Joint Optimisation

Challenges

1. Abduction is non-deterministic.
2. The under-trained perception model has low accuracy

The Abductive Optimisation

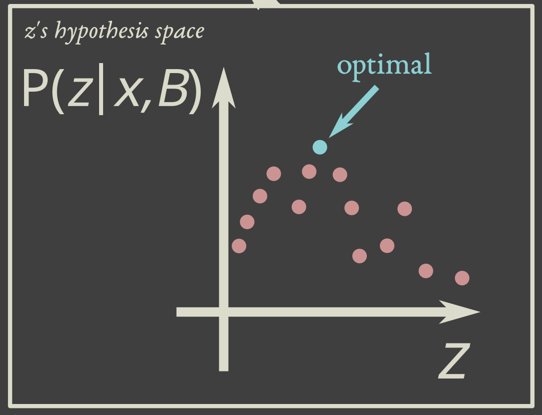
Observing evidence $\langle x, y\rangle$, and maximises the *consistency* of $H \cup z$ and f w.r.t. the evidence and background knowledge:



`example([1, 3, 5], 15).`

Domain Knowledge:

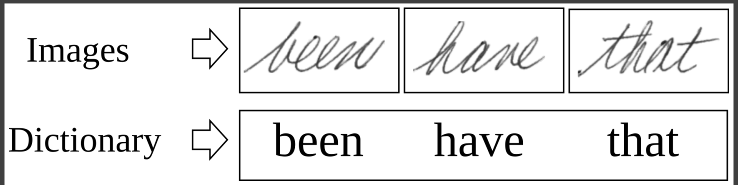
a correct program



Optimisation in this unknown distribution:
- exact inference [Manhaeve et al., 2018]
- evolutionary algorithm [Dai et al., 2019]
- Reinforcement learning [Dong et al., 2019]
- MCMC sampling [Li et al., 2020]

Example: Abduction with Ground KB

Example



Task: Optical Character Recognition

- Ligatures
- Limited labelled examples (~10%)

Background Knowledge: Dictionary

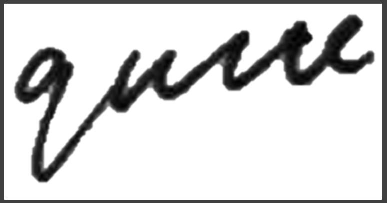
```
Vocab size:  115,320

['sure', 'he', 'during', 'of', 'booty', 'gastronomy', 'boy',
 'The', 'and', 'in', ...]
```

Cai, et al. Abductive Learning with Ground Knowledge Base, 2021.

Abduction by Looking Up in Dictionary

Grounded Abductive Learning

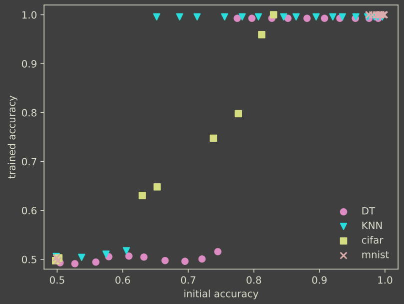


1. CRNN output： qute
2. Not in the dictionary (ground KB)
3. **Abduction**: close (Hamming dist.) guesses are *cute, quite, quit, quirk*, ……
4. Return the most possible one according to NN score: *quite*

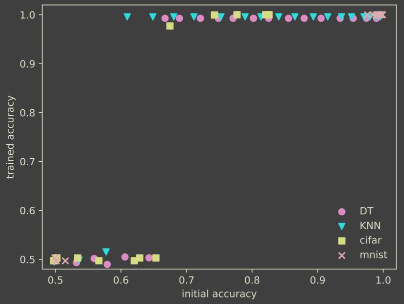
Cai, et al. Abductive Learning with Ground Knowledge Base, 2021.

Successful Abduction Needs Good Pre-train

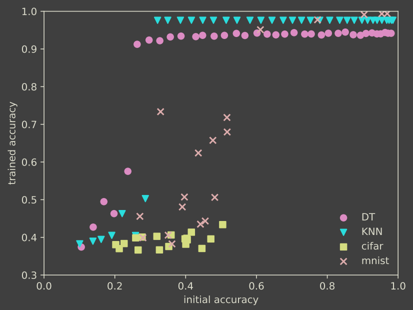
A weak perception model may get the abduction trapped in local optima.



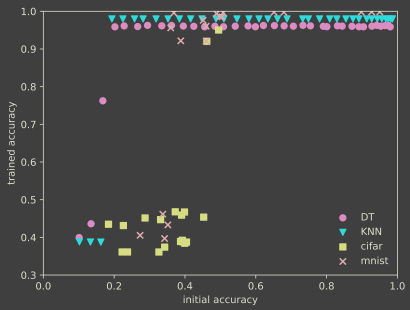
(a) Hamming w/o confidence.



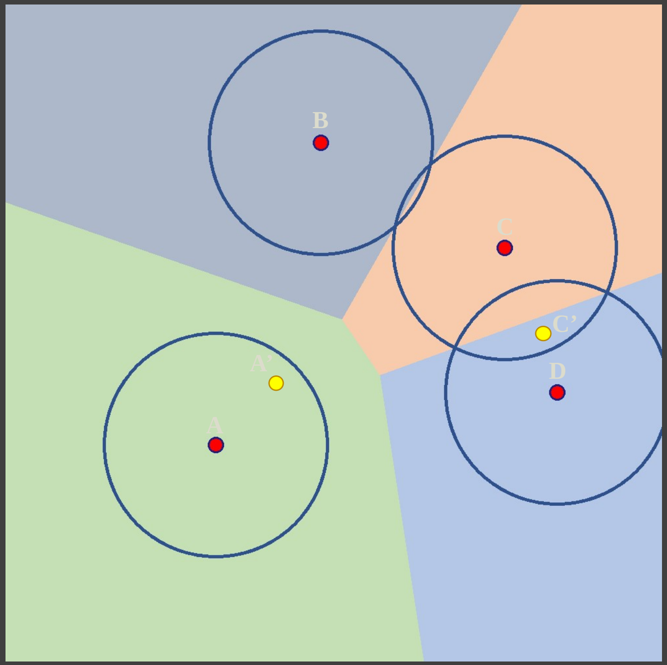
(b) Hamming w/ confidence.



(c) Equation w/o confidence.



(d) Equation w/ confidence.



Question

What kind of background knowledge is **helpful** for training f?

Location Matrix Q

Target concept: `X3 is X1 /\ X2`: $(\tau = \{[0,0,0],[0,1,0],[1,0,0],[1,1,1]\})$

- 1 appears 7 times, 2 of which are in the 1st and 2nd position, 3 times in the 3rd;
- 0 appears 5 times, 2 of which are in the 1st and 2nd position, only 1 time in the 3rd;

$$Q = \begin{pmatrix} 2/7 & 2/7 & 3/7 \\ 2/5 & 2/5 & 1/5 \end{pmatrix}$$

Target concept: `0 is X1 /\ X2`: $(\tau = \{[0,0],[0,1],[1,0]\})$

- 1 appears 2 times, either in the 1st or the 2nd position;
- 0 appears 4 times, 2 of which are in the 1st and 2nd position;

$$Q = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

Some knowledge bases are extremely useful!

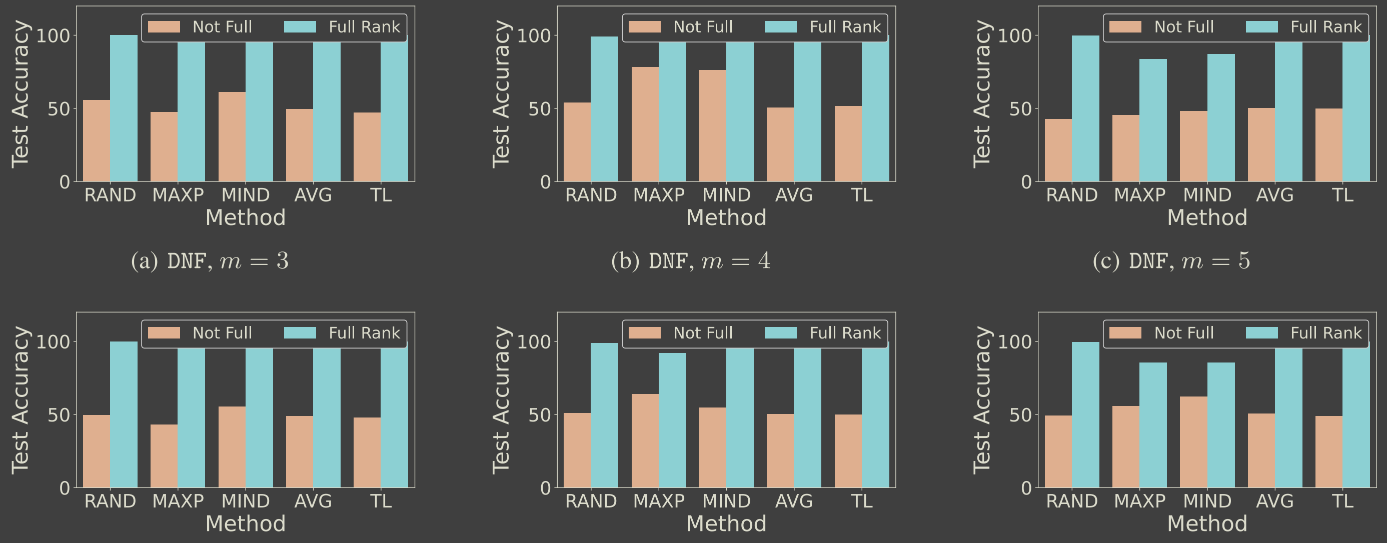
Theorem *If the probability matrix Q has full row rank and the cross-entropy loss is used, then the perception minimiser $f_L^* = \arg\min_h \mathcal{R}_L(h)$ recovers the true minimiser $f^* = \arg\min_h \mathcal{R}(h)$*, i.e., $f_L^* = f^*$.

Remark:

- The ground truth symbols of raw inputs can be reliably recovered if the probability matrix Q has **full row rank**
- We also derived a **tight bound** for this kind of knowledge bases.

L. Tao, et.al., Deciphering Raw Data in Neuro-Symbolic Learning with Provable Guarantees. AAAI'24.

Experiments



(a) DNF, $m = 3$

(b) DNF, $m = 4$

(c) DNF, $m = 5$

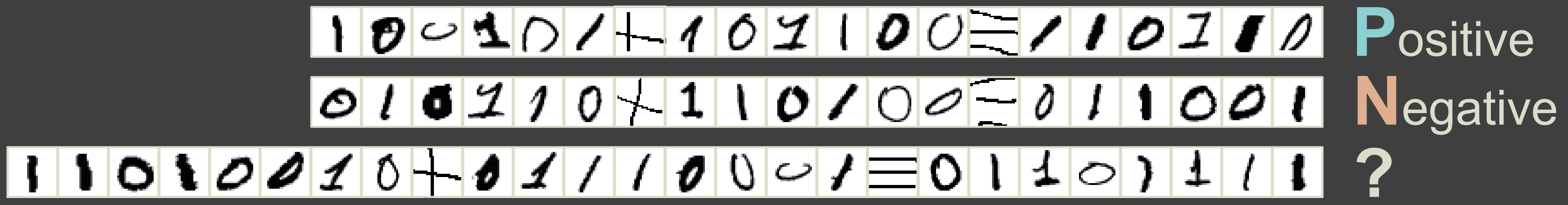
(d) CNF, $m = 3$

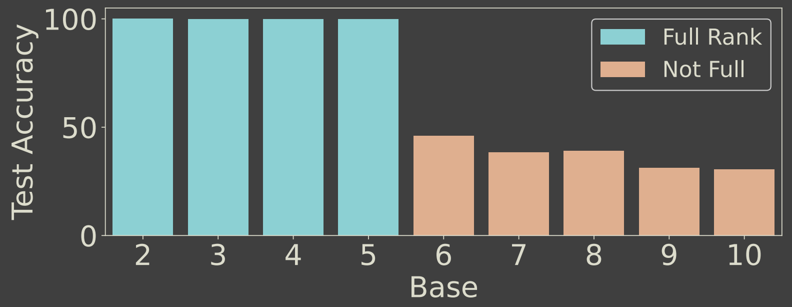
(e) CNF, $m = 4$

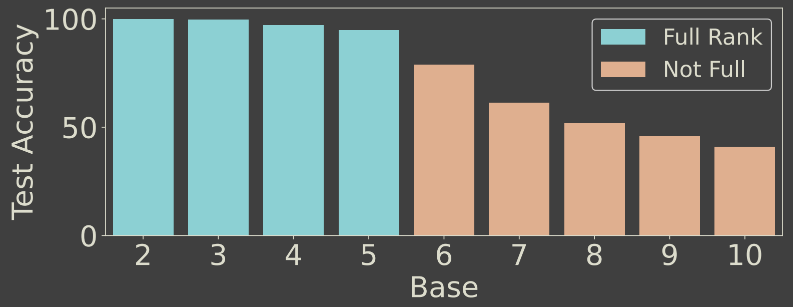
(f) CNF, $m = 5$

- Randomly generated CNFs/DNFs as target concepts with different Q matrices
- Abduction by `RAND` (random choice), `MAXP` (maximal perception probability), `MIND` (minimal Hamming distance), `AVG` (average/expectation over symbols)
- `TL`: Minimising the loss function derived from our bound

Experiments







- *Handwritten Equation Deciphering* task with different number-based addition tasks
- Abduction with `MAXP` (left) and `TL` (right) strategy
- Cold start: NO pre-training for the perception model

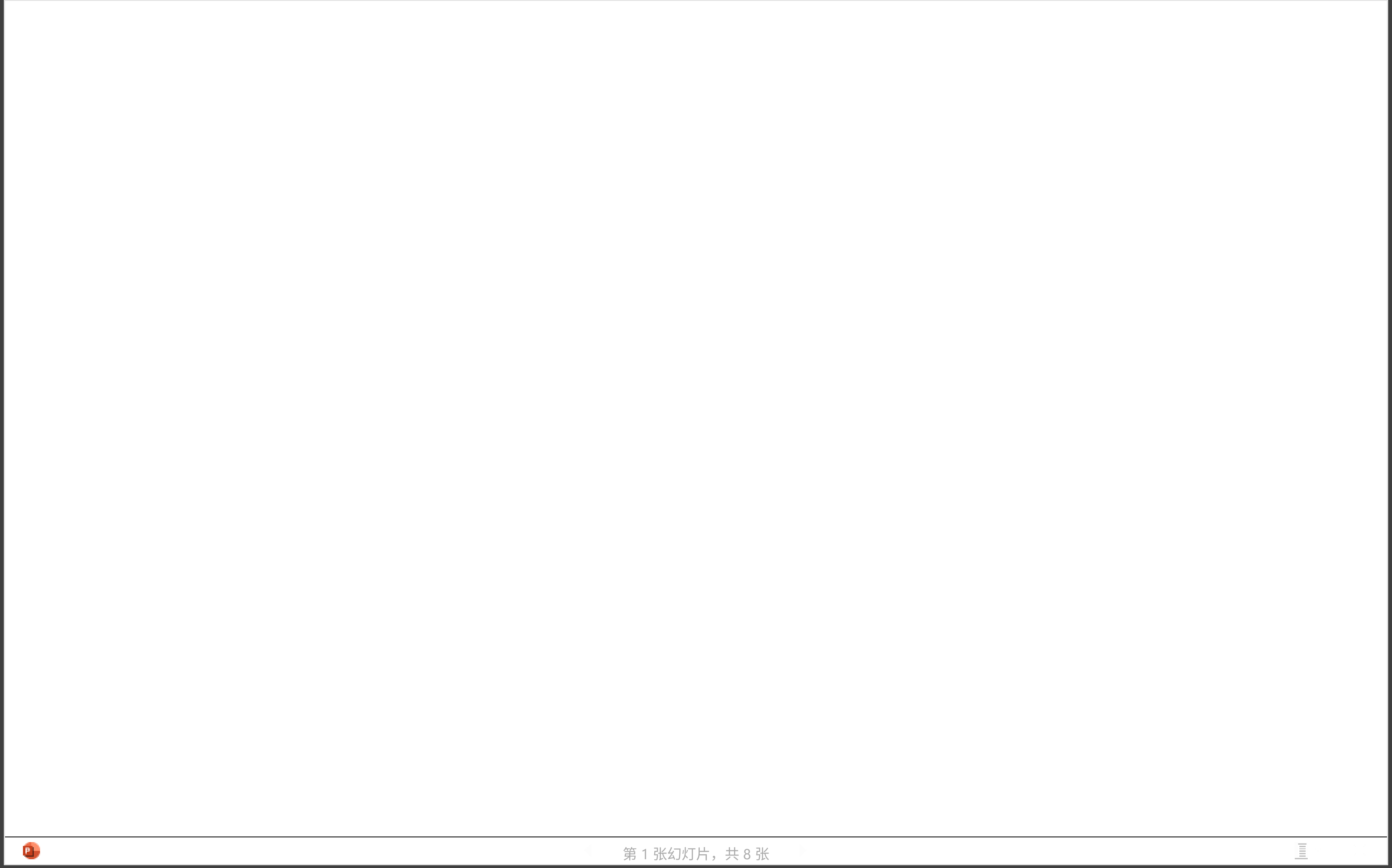
What about weaker knowledge bases?

Can **machine learning model** help?

Generate logic constraints

第 1 张幻灯片，共 2 张

Learning to "reflect"

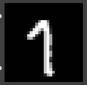
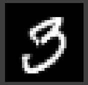
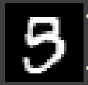


Learning to "reflect"

The reflection layer learns which parts of the problem (sub-problems) are too hard and require formal verification

When Knowledge is Incomplete

A simple example

`sum([``, ``, ``], 15).`

1. `sum([img_1, img_2, img_3, ...])`
2. `sum([SUM, img_3, ...]), SUM=add(img_1,img_2)`
3. `sum([SUM, ...]), SUM=add(SUM, img_3)`
4. This procedure can be executed repeatedly, until only one element left
5. `return SUM`

If the function / program is unknown

$$\begin{aligned} 15 &= f([img_{1,1}, img_{1,2}, \ldots]) \\ 21 &= f([img_{2,1}, img_{2,2}, \ldots]) \\ 8 &= f([img_{3,1}, img_{3,2}, \ldots]) \\ &\ldots \end{aligned}$$

Logical Abduction

Starts from **incomplete** observations, proceeds to the likeliest possible **explanation** for them.

A **program / abstraction** is also an **explanation**, but in higher-order.

W.-Z. Dai and S. H. Muggleton, Abductive Knowledge Induction From Raw Data, IJCAI-21

Abduction in Second-order Logic

$$\exists P\, Q\, R\, \forall X\, Y\, Z,\ P(X,Y) \leftarrow Q(X,Z), R(Z,Y)$$

- P, Q, R are 2nd-order variables (existentially quantified)
  - i.e., the groudings of them are **dyadic relations**, e.g., state-transitions like `add_first_two_number(List1,List2)` or functions like `square(X,Y)`
- X, Y, Z are 1st-order variables (universally quantified)
  - the groundings of them are the constants, e.g., `[1,2,3,4]` in `add_first_two_number([1,2,3,4],[3,3,4])`

$$\forall L_1\, L_2\, L_3,\ f(L_1,L_2) \leftarrow add\_first\_two(L_1,L_3), add\_first\_two(L_3,L_2)$$

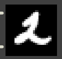
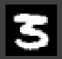
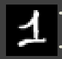
Supply it with a meta-interpreter...

```
prove([], Prog, Prog, [], Prob, Prob).
prove([Atom|As], Prog1, Prog1, Abds, Prob1, Prob2) :-
    deduce(Atom),
    prove(As, Prog1, Prog2, Abds, Prob1, Prob2).
%%%%%%% Use abduction to prove the current example %%%%%%%
prove([Atom|As], Prog1, Prog1, Abds, Prob1, Prob2) :-
    call_abducible(Atom, Abd, Prob),
    Prob3 is Prob1 * Prob,
    get_max_prob(Max), Prob3 > Max,
    set_max_prob(Prob3),
    prove(As, Prog1, Prog1, [Abd|Abds], Prob3, Prob2).
%%%%%%% Meta Interpretive program induction %%%%%%%
prove([Atom|As], Prog1, Prog2, Abds, Prob1, Prob2) :-
    meta-rule(Name, MetaSub,(Atom :- Body), Order),
    Order,
    substitue(metasub(Name, MetaSub), Prog1, Prog3),
    prove(Body, Prog3, Prog4),
    prove(As, Prog4, Prog2, Abds, Prob1, Prob2)
```

W.-Z. Dai and S. H. Muggleton, Abductive Knowledge Induction From Raw Data, IJCAI-21

Then it can do this

Given an example:

`f([``,` `,` `], 6).`

Denoting the unknown labels of images in the list as `z1, z2, z3,...`

(1) Abduce *f* with composite functions (2nd-order): :

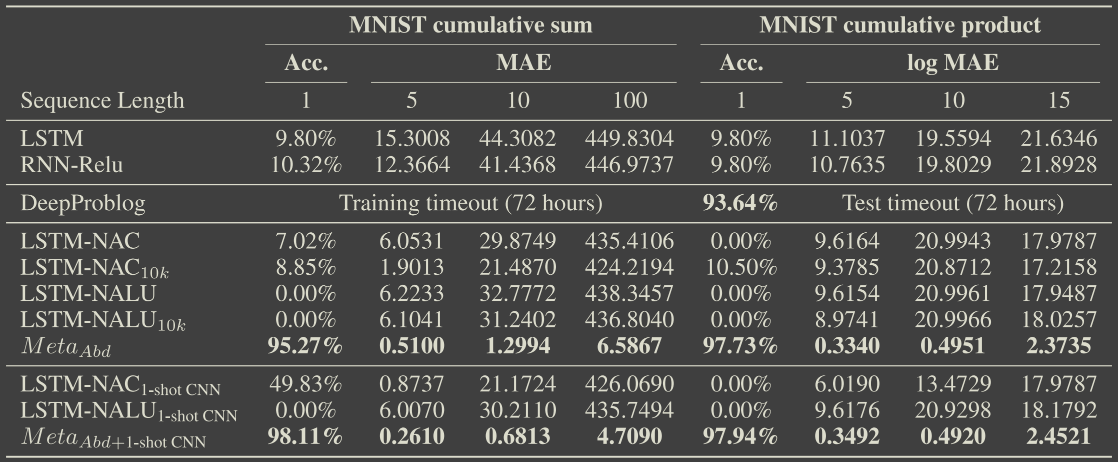
- $f = add \circ add$, i.e., `z1+z2+z3`
- $f = mult \circ add$, i.e., `(z1+z2)*z3`
- $f = add \circ mult$, i.e., `(z1*z2)+z3`,
- $f = add^n$, i.e., `z1+z2+z3+...`
- …

(2) And then, abduce `zi` with *f* (1st-order):

~~0,0,0~~	~~0,0,1~~	…	…	…	~~0,9,9~~
~~1,0,0~~	~~1,0,1~~	…	1,2,3	…	~~1,9,9~~
…	…	2,3,1	…	…	…
~~9,0,0~~	…	…	…	…	~~9,9,9~~

(3) Pick the best explanation

Experiment: Accumulative Sum/Product



	MNIST cumulative sum				MNIST cumulative product			
	Acc.	**MAE**			**Acc.**	**log MAE**		
Sequence Length	1	5	10	100	1	5	10	15
LSTM	9.80%	15.3008	44.3082	449.8304	9.80%	11.1037	19.5594	21.6346
RNN-Relu	10.32%	12.3664	41.4368	446.9737	9.80%	10.7635	19.8029	21.8928
DeepProblog	Training timeout (72 hours)				**93.64%**	Test timeout (72 hours)		
LSTM-NAC	7.02%	6.0531	29.8749	435.4106	0.00%	9.6164	20.9943	17.9787
LSTM-NAC$_{10k}$	8.85%	1.9013	21.4870	424.2194	10.50%	9.3785	20.8712	17.2158
LSTM-NALU	0.00%	6.2233	32.7772	438.3457	0.00%	9.6154	20.9961	17.9487
LSTM-NALU$_{10k}$	0.00%	6.1041	31.2402	436.8040	0.00%	8.9741	20.9966	18.0257
$Meta_{Abd}$	**95.27%**	**0.5100**	**1.2994**	**6.5867**	**97.73%**	**0.3340**	**0.4951**	**2.3735**
LSTM-NAC$_{\text{1-shot CNN}}$	49.83%	0.8737	21.1724	426.0690	0.00%	6.0190	13.4729	17.9787
LSTM-NALU$_{\text{1-shot CNN}}$	0.00%	6.0070	30.2110	435.7494	0.00%	9.6176	20.9298	18.1792
$Meta_{Abd+\text{1-shot CNN}}$	**98.11%**	**0.2610**	**0.6813**	**4.7090**	**97.94%**	**0.3492**	**0.4920**	**2.4521**

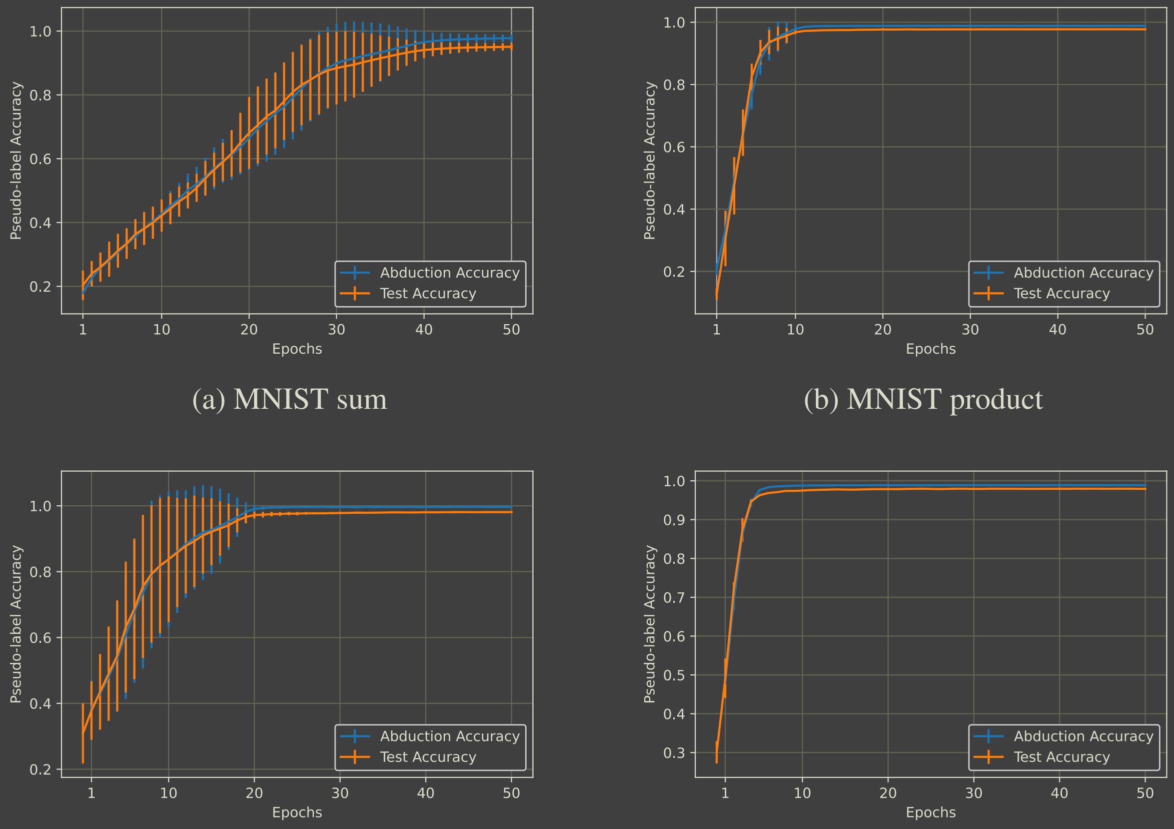
Table 2: Accuracy on the MNIST cumulative sum/product tasks.

Learned Programs:

```
%% Accumulative Sum
f(A,B):-add(A,C), f(C,B).
f(A,B):-eq(A,B).

%% Accumulative Product
f(A,B):-mult(A,C), f(C,B).
f(A,B):-eq(A,B).
```

Experiment: Neural Net Classifier Accuracy



(a) MNIST sum

(b) MNIST product

(c) MNIST sum with 1-shot CNN pre-train

(d) MNIST product with 1-shot CNN pre-train

Figure 5: Pseudo-label accurracy during $Meta_{Abd}$ and $Meta_{Abd+\text{1-shot CNN}}$ learning.

Rethink about Symbolism

Problem solving in general (for human)

BIG Problem

Problem solving in general (for human)

`Working Memory` → [small problem 1, small problem 2, …]

Problem solving in general (for human)

`Working Memory` → [**small problem #1**, small problem #2, …]

Problem solving in general (for human)

`Working Memory` → [[smaller problem #1.1, smaller problem #1.2], small problem #2, ...]

Problem solving in general (for human)

`Working Memory` → [[~~smaller problem #1.1~~, smaller problem #1.2], small problem #2, ...]

Problem solving in general (for human)

`Working Memory` → [[~~smaller problem #1.1~~, ~~smaller problem #1.2~~], small problem #2, …]

Problem solving in general (for human)

`Working Memory` → [~~small problem #1~~, **small problem #2**, …]

Rethink about Symbolism

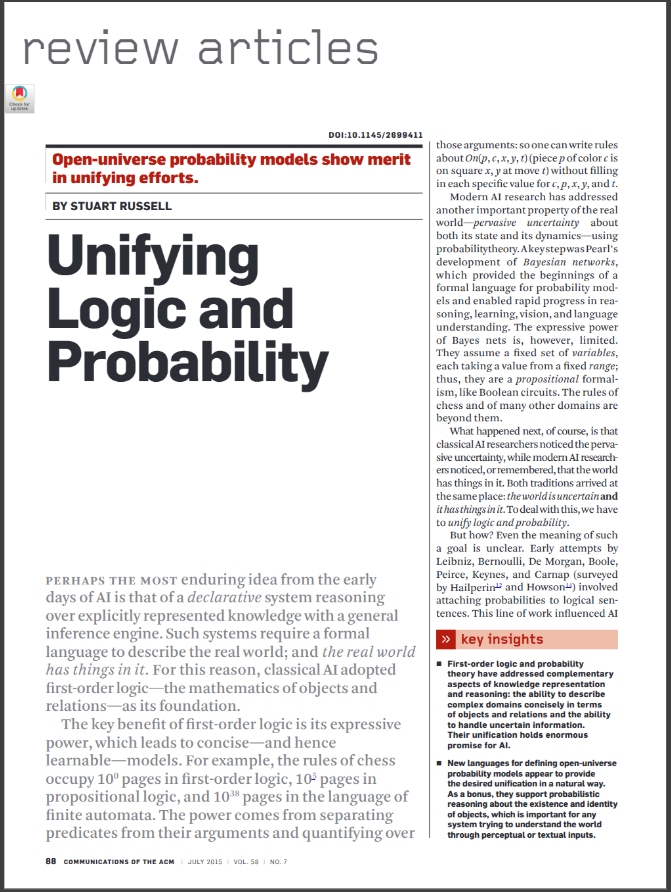
In general, task-solving should be:

1. **Procedural**: Divide and conquer
2. **Recursive**: Repeatedly divide and conquer (until its trivial)
   - If `subtask[i]` is not trivial, then further decompose it into `[subtask[i][1],...]`
     - For trivial tasks, solve it by **fast thinking**
3. **Compositional**: symbols can be reused & (sub)procedures can be generalised, e.g.,
   - Procedure of `subtask[i]`

Big Question: The existence of symbols

> “… these approaches miss an important consequence of uncertainty in a world of things: *uncertainty about what things are in the world*. Real objects seldom wear unique identifiers or preannounce their existence like the cast of a play. In areas such as vision, language understanding, … , the existence of objects *must be inferred from raw data* (pixels, strings, and so on) that contain no explicit object references.”

[Russell, 2015]



review articles

DOI:10.1145/2699411

Open-universe probability models show merit in unifying efforts.

BY STUART RUSSELL

Unifying Logic and Probability

PERHAPS THE MOST enduring idea from the early days of AI is that of a *declarative* system reasoning over explicitly represented knowledge with a general inference engine. Such systems require a formal language to describe the real world; and *the real world has things in it*. For this reason, classical AI adopted first-order logic—the mathematics of objects and relations—as its foundation.

The key benefit of first-order logic is its expressive power, which leads to concise—and hence learnable—models. For example, the rules of chess occupy 10^0 pages in first-order logic, 10^5 pages in propositional logic, and 10^{38} pages in the language of finite automata. The power comes from separating predicates from their arguments and quantifying over those arguments: so one can write rules about $On(p, c, x, y, t)$ (piece p of color c is on square x, y at move t) without filling in each specific value for c, p, x, y, and t.

Modern AI research has addressed another important property of the real world—*pervasive uncertainty* about both its state and its dynamics—using probability theory. A key step was Pearl's development of *Bayesian networks*, which provided the beginnings of a formal language for probability models and enabled rapid progress in reasoning, learning, vision, and language understanding. The expressive power of Bayes nets is, however, limited. They assume a fixed set of *variables*, each taking a value from a fixed *range*; thus, they are a *propositional* formalism, like Boolean circuits. The rules of chess and of many other domains are beyond them.

What happened next, of course, is that classical AI researchers noticed the pervasive uncertainty, while modern AI researchers noticed, or remembered, that the world has things in it. Both traditions arrived at the same place: *the world is uncertain* **and** *it has things in it*. To deal with this, we have to *unify logic and probability*.

But how? Even the meaning of such a goal is unclear. Early attempts by Leibniz, Bernoulli, De Morgan, Boole, Peirce, Keynes, and Carnap (surveyed by Hailperin[12] and Howson[14]) involved attaching probabilities to logical sentences. This line of work influenced AI

» key insights

- **First-order logic and probability theory have addressed complementary aspects of knowledge representation and reasoning: the ability to describe complex domains concisely in terms of objects and relations and the ability to handle uncertain information. Their unification holds enormous promise for AI.**
- **New languages for defining open-universe probability models appear to provide the desired unification in a natural way. As a bonus, they support probabilistic reasoning about the existence and identity of objects, which is important for any system trying to understand the world through perceptual or textual inputs.**

88 COMMUNICATIONS OF THE ACM | JULY 2015 | VOL. 58 | NO. 7

The real-world challenge for us

Can we learn a program / plan / grammar / graph / …

from sensory raw data only,

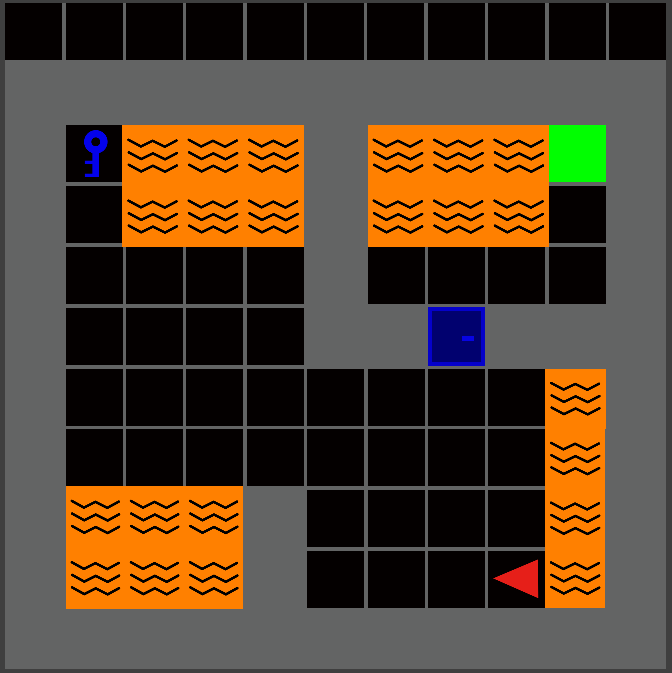
and **without pre-defined primitive symbols**,

and works in a world that **only has sensory raw inputs and only allows low-level motions as outputs**?

Learning Procedural Knowledge from Raw Traces

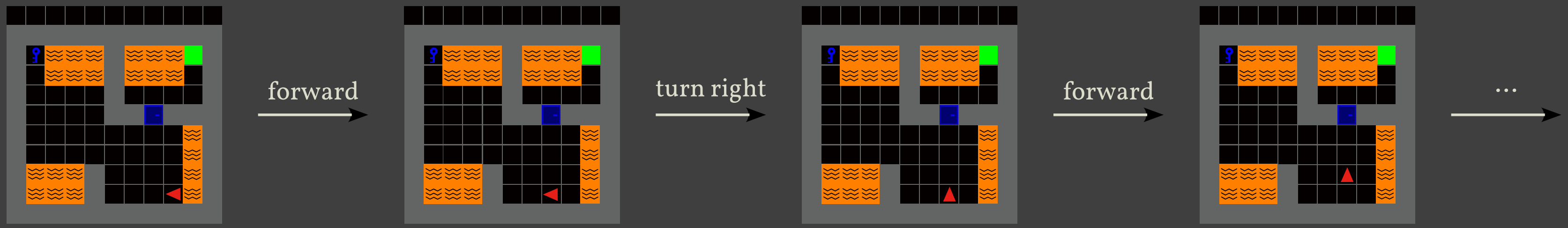
An Example of Abstraction

- **Environment**: Minigrid
- **Task**: Reach to the goal (🟩)
- **Low-level Inputs**: Raw images
- **Low-level Actions**: ⬆️, ↩️, ↪️, 🫴, 🫳, ⏻👉
- **Reward**: { -1 (fail), 1 (success) }



Z. Wang, et. al., From End-to-end to Step-by-step: Learning to Abstract via Abductive Reinforcement Learning, IJCAI 2025.

Original Problem (Ground MDP)



An end2end reinforcement learning (RL) task is an MDP in a sub-symbolic environment: $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$

- $\mathcal{S}$: original image
- $\mathcal{A}$: low-level actions
- P: state transition of low-level actions

Z. Wang, et. al., From End-to-end to Step-by-step: Learning to Abstract via Abductive Reinforcement Learning, IJCAI 2025.

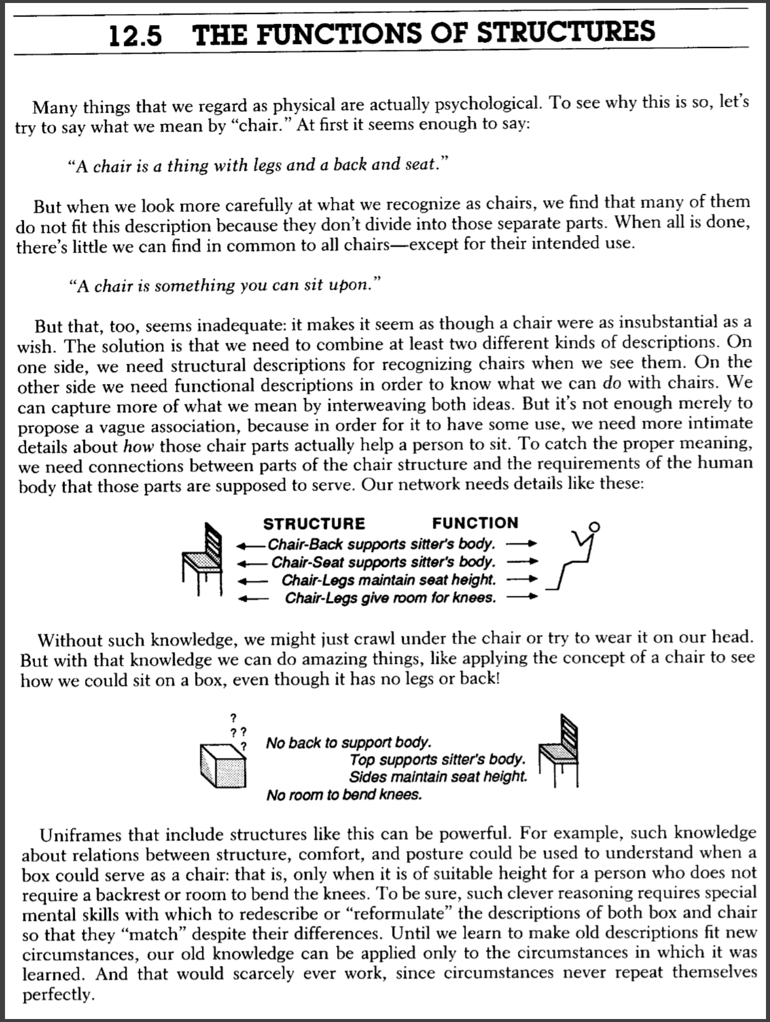
Ground-truth abstraction

The abstraction of the task is a sub-task decomposition:

- If there are propositional symbols:
  - K - has key; U - door unlocked; G - reached to the goal
- $\neg K \wedge \neg U \wedge \neg G \implies K \wedge \neg U \wedge \neg G \implies K \wedge U \wedge \neg G \implies K \wedge U \wedge G$
- How to learn such task abstraction without symbols of objects, and even without the definition of grids?
- Very difficult! **So we only discover descrete, abstract states**, and learn the state-transitions

Z. Wang, et. al., From End-to-end to Step-by-step: Learning to Abstract via Abductive Reinforcement Learning, IJCAI 2025.

Minsky's Example (*The Society of Mind*, 1986)



12.5 THE FUNCTIONS OF STRUCTURES

Many things that we regard as physical are actually psychological. To see why this is so, let's try to say what we mean by "chair." At first it seems enough to say:

> "*A chair is a thing with legs and a back and seat.*"

But when we look more carefully at what we recognize as chairs, we find that many of them do not fit this description because they don't divide into those separate parts. When all is done, there's little we can find in common to all chairs—except for their intended use.

> "*A chair is something you can sit upon.*"

But that, too, seems inadequate: it makes it seem as though a chair were as insubstantial as a wish. The solution is that we need to combine at least two different kinds of descriptions. On one side, we need structural descriptions for recognizing chairs when we see them. On the other side we need functional descriptions in order to know what we can *do* with chairs. We can capture more of what we mean by interweaving both ideas. But it's not enough merely to propose a vague association, because in order for it to have some use, we need more intimate details about *how* those chair parts actually help a person to sit. To catch the proper meaning, we need connections between parts of the chair structure and the requirements of the human body that those parts are supposed to serve. Our network needs details like these:

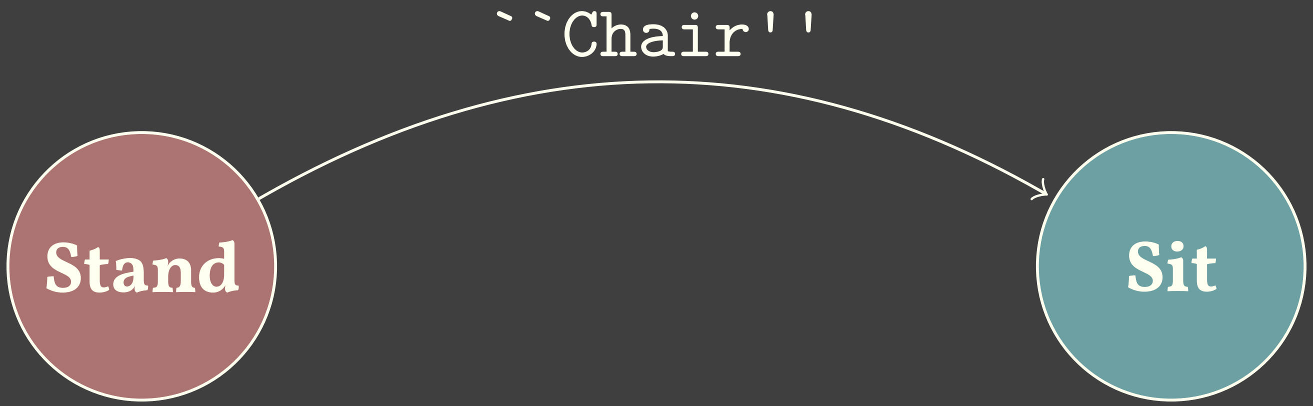
STRUCTURE
FUNCTION
Chair-Back supports sitter's body.
Chair-Seat supports sitter's body.
Chair-Legs maintain seat height.
Chair-Legs give room for knees.

Without such knowledge, we might just crawl under the chair or try to wear it on our head. But with that knowledge we can do amazing things, like applying the concept of a chair to see how we could sit on a box, even though it has no legs or back!

Uniframes that include structures like this can be powerful. For example, such knowledge about relations between structure, comfort, and posture could be used to understand when a box could serve as a chair: that is, only when it is of suitable height for a person who does not require a backrest or room to bend the knees. To be sure, such clever reasoning requires special mental skills with which to redescribe or "reformulate" the descriptions of both box and chair so that they "match" despite their differences. Until we learn to make old descriptions fit new circumstances, our old knowledge can be applied only to the circumstances in which it was learned. And that would scarcely ever work, since circumstances never repeat themselves perfectly.

Functions can define concepts

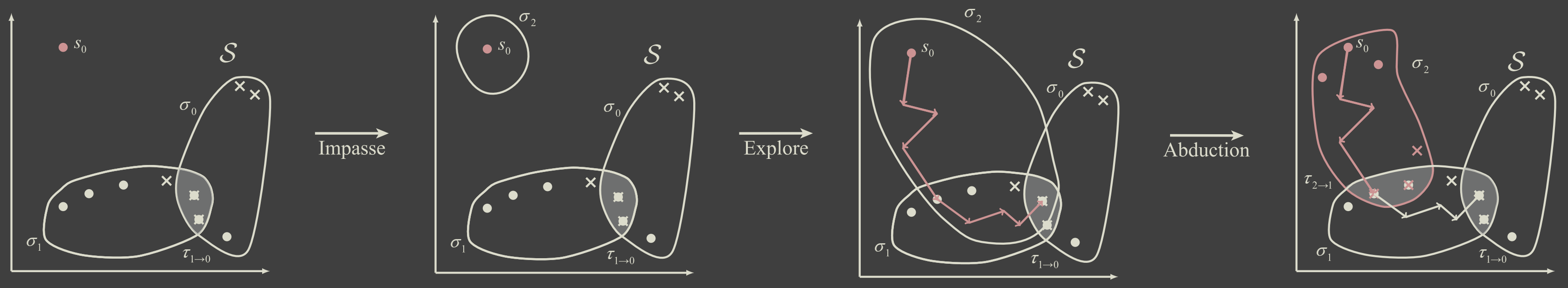
The concept of “chair” can be described as state transition functions, such as:



Learning to abstract

Our work learns to abstract via impasse-driven discovery [Unruh and Rosenbloom, 1989], which is implemented based on the idea of Abductive Learning (ABL).

- Meeting impasse → Exploring and gathering successful trajectories.
- Trajectories → Abductive learning to get σ_{new} and transition $\tau_{\text{new}\to\text{old}}$.
- The Abstract State Machine is updated → Training atomic policies in sub-MDPs.
  - (Sub-MDP is a subalgebra of the original MDP defined based on abstract states)

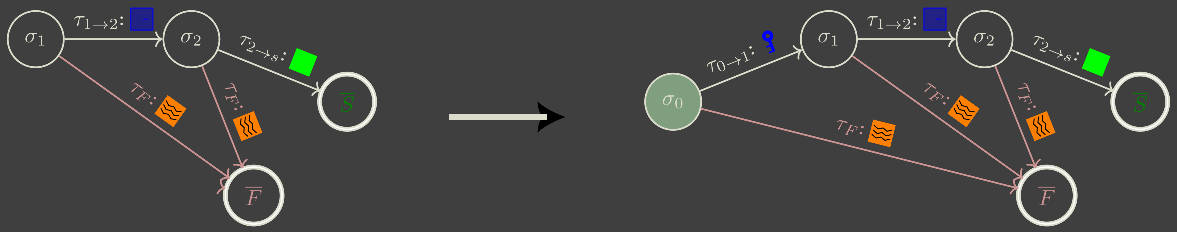


Z. Wang, et. al., From End-to-end to Step-by-step: Learning to Abstract via Abductive Reinforcement Learning, IJCAI 2025.

Learning to abstract

Our work learns to abstract via impasse-driven discovery [Unruh and Rosenbloom, 1989], which is implemented based on the idea of Abductive Learning (ABL).

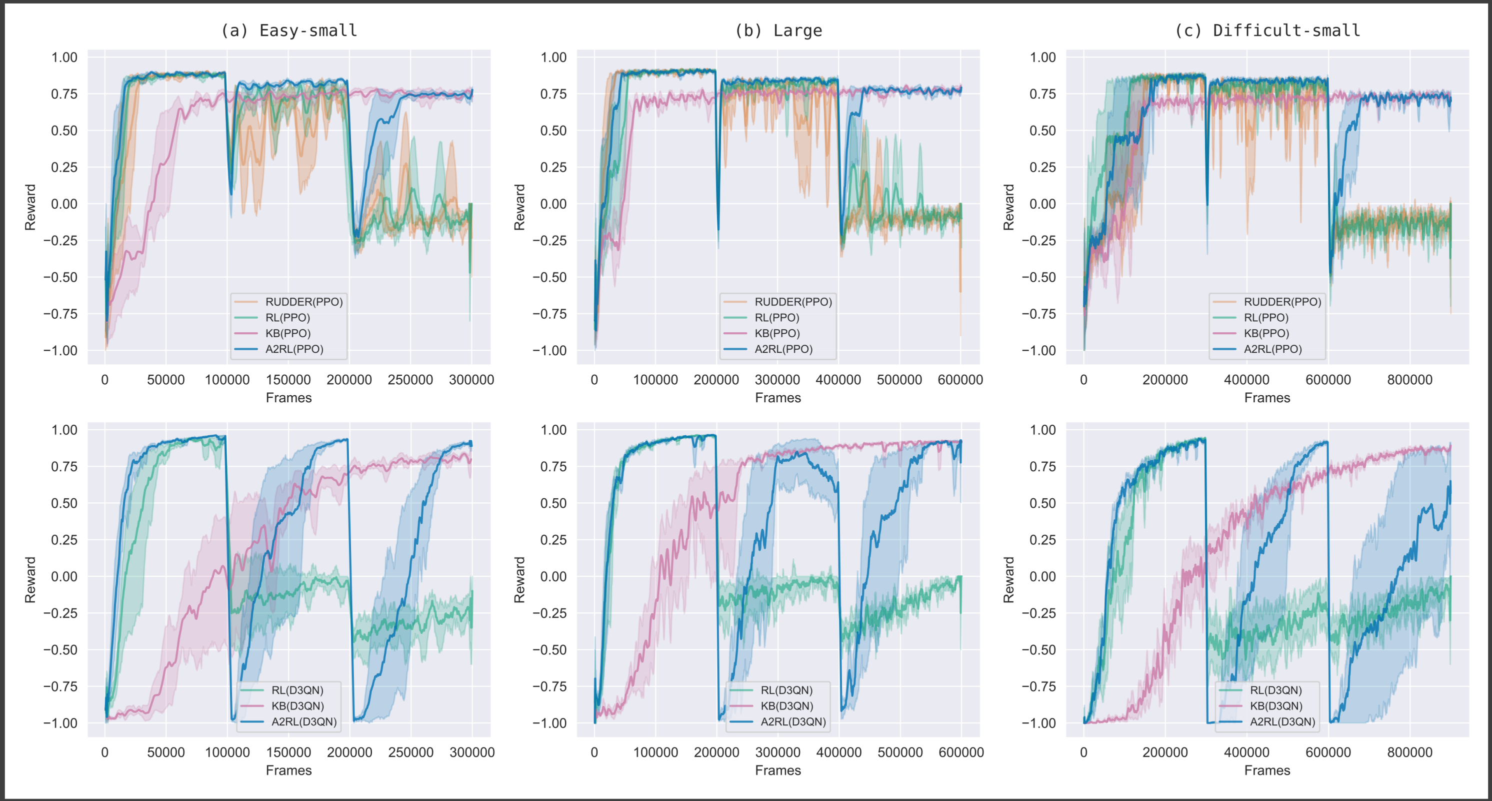
- Meeting impasse → Exploring and gathering successful trajectories.
- Trajectories → Abductive learning to get σ_{new} and transition $\tau_{\text{new}\to\text{old}}$.
- The Abstract State Machine is updated → Training atomic policies in sub-MDPs.
  - (Sub-MDP is a subalgebra of the original MDP defined based on abstract states)



$$\begin{aligned} Success(a,b) \leftarrow\quad & \boldsymbol{\pi}_{0\to1}(a,c) \land c \in \tau_{0\to1} \\ & \land\, \boldsymbol{\pi}_{1\to2}(c,d) \land d \in \tau_{1\to2} \\ & \land\, \boldsymbol{\pi}_{2\to s}(d,b) \land b \in \overline{S} \\ F(a,b) \leftarrow\quad & \boldsymbol{\pi}_F(a,b) \land b \in \overline{F} \end{aligned}$$

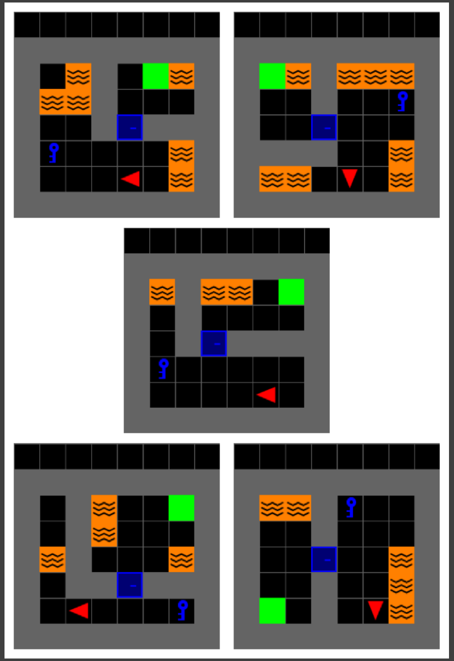
Z. Wang, et. al., From End-to-end to Step-by-step: Learning to Abstract via Abductive Reinforcement Learning, IJCAI 2025.

Experimental results

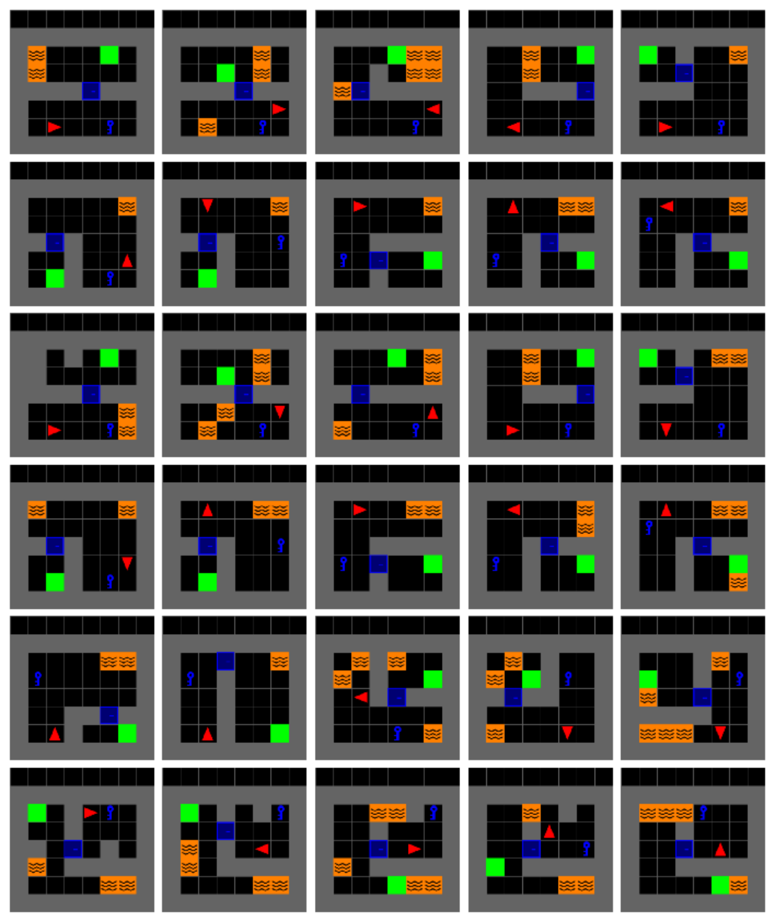


Abductive atate abstraction vs vanilla end2end reinforcement learning vs hierarchical reinforcement learning with groundtruth subtask hierarchy

Out-of-distribution Generalization

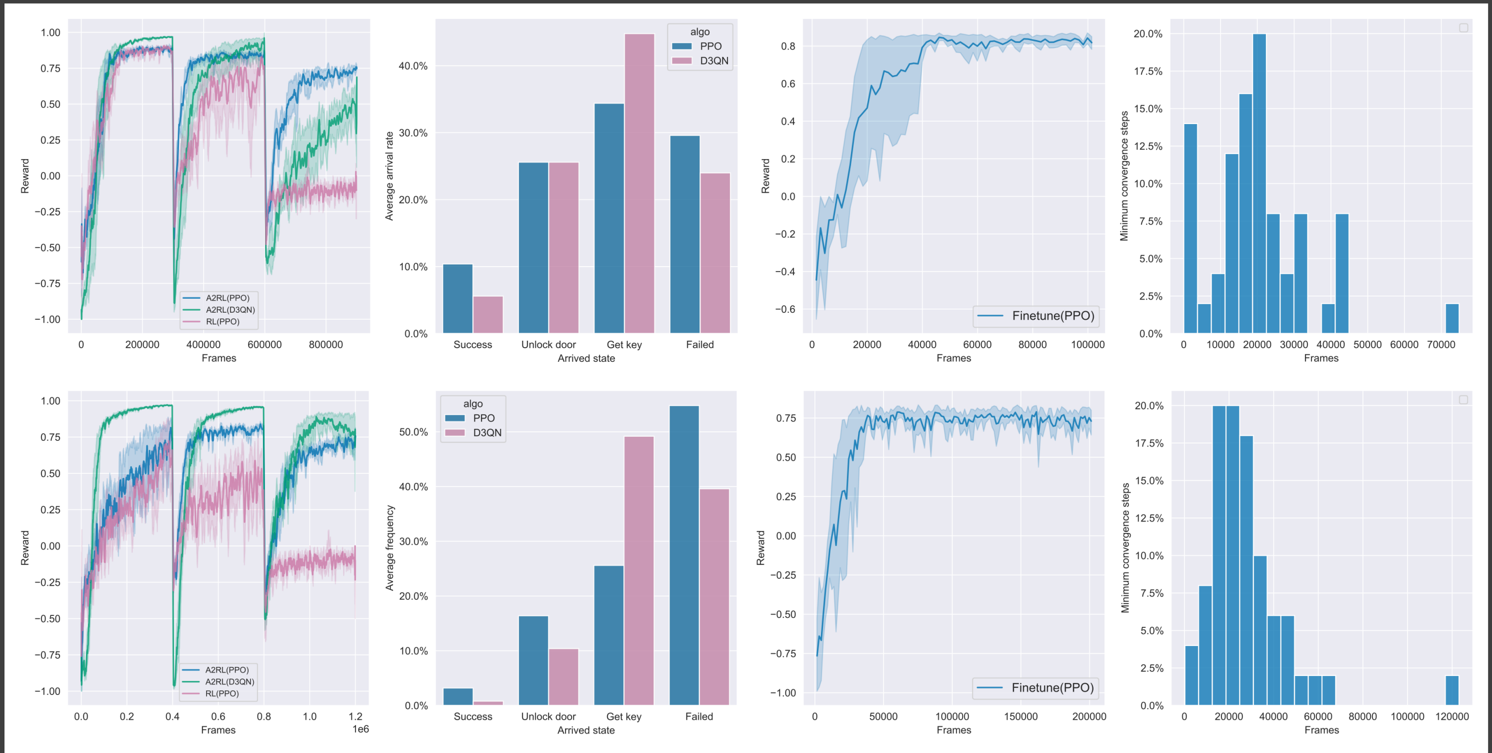


Trained on 5 maps



Tested on 50 unseen maps

Out-of-distribution Generalization

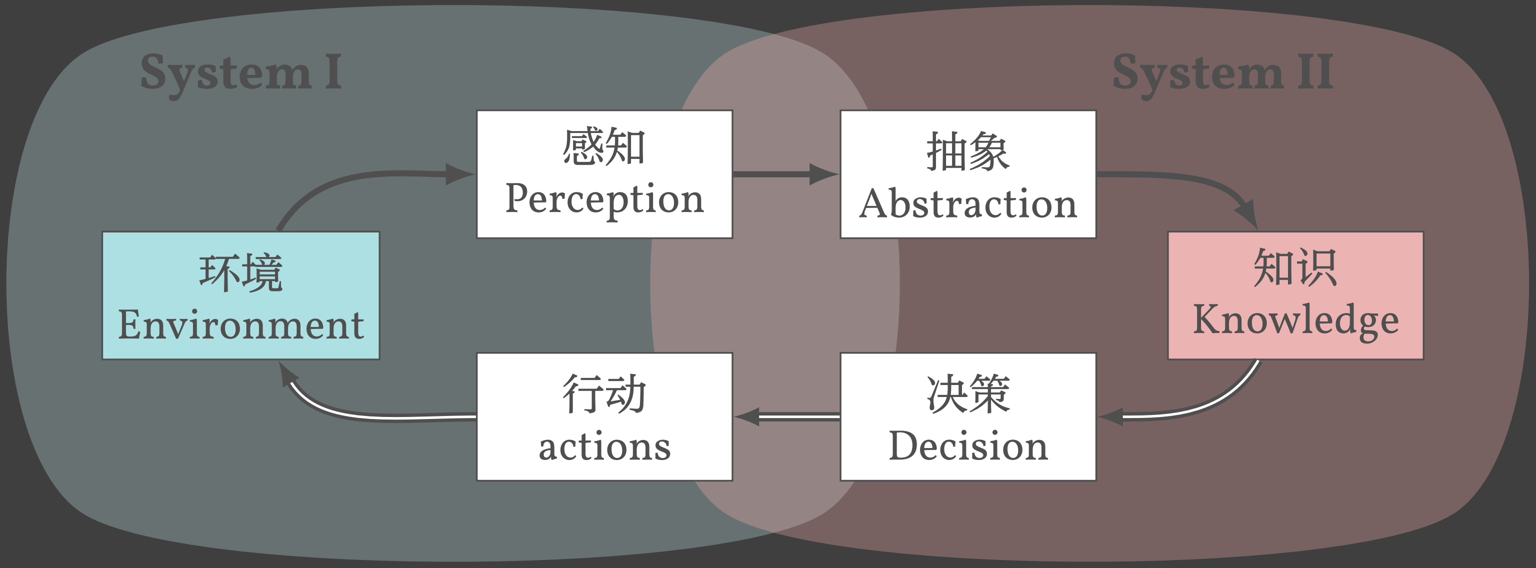


1st col.: training reward in 5 random maps. End2end DRL still cannot converge;
2nd col.: testing reward in 50 random maps. The abstracted model did extrapolates;
3rd-4th cols.: continual learning in 50 random maps, requires much less training data.

Z. Wang, et. al., From End-to-end to Step-by-step: Learning to Abstract via Abductive Reinforcement Learning, IJCAI 2025.

Summary

Learning ↔ Abstraction ↔ Reasoning

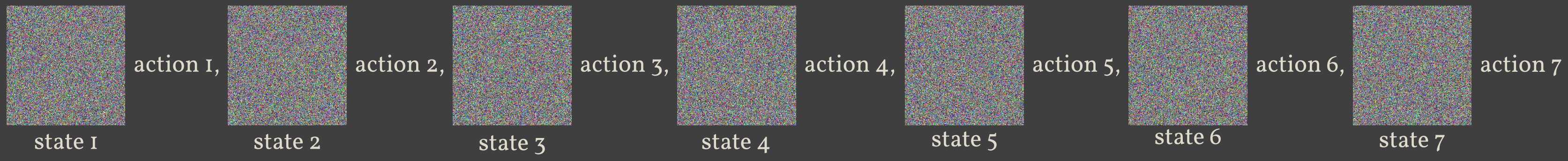


“The era of experience” —— Silver and Sutton, 2025

Abstracting raw traces?

Open problem: Program (grammar) induction with neuro-predicate (concept) invention.

For an under-trained agent, the traces look like this:

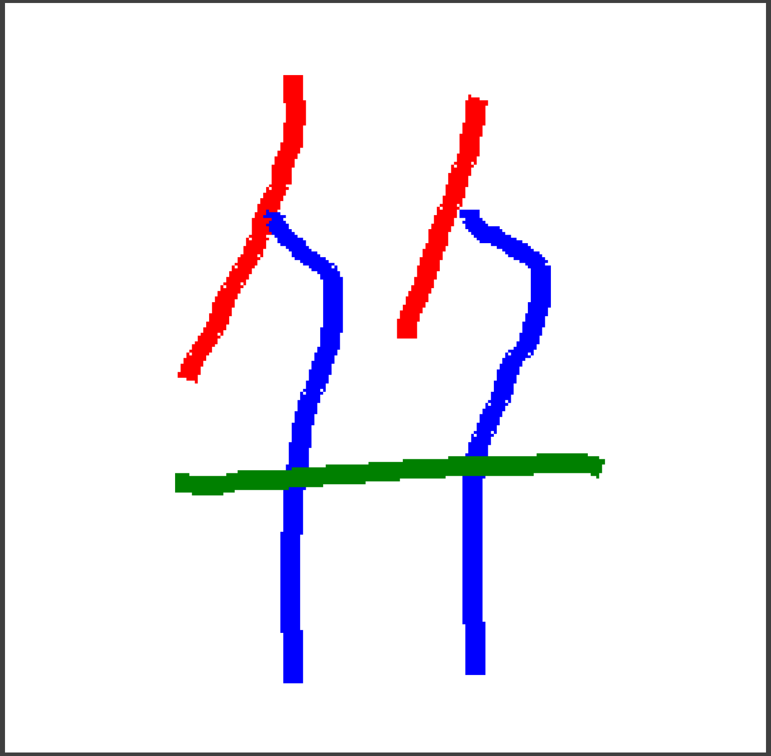


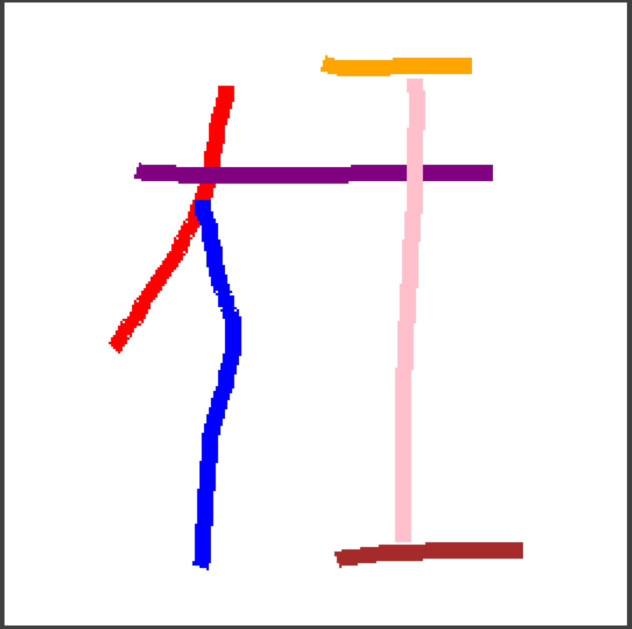
state 1, action 1, state 2, action 2, state 3, action 3, state 4, action 4, state 5, action 5, state 6, action 6, state 7, action 7

- We want model it like $a^*b^*c^*$, but which segments corresponds to a, b and c?
  - Needs to learn state perception models $\phi_a : \mathbb{R}^n \to \{0, 1\}$, $\phi_b : \mathbb{R}^n \to \{0, 1\}, \ldots$
  - while allowing the alphabet $\{a, b, c, \ldots\}$ to increase during learning
  - and induce rules like $c \leftarrow a \wedge b$ for high-level planning
  - meanwhile, train low-level action models $\pi_a : \mathbb{R}^n \to \mathbb{R}^m, \ldots$ to execute the plan
- A possible solution: combining **abduction** and **induction** and deep / statistical learning.

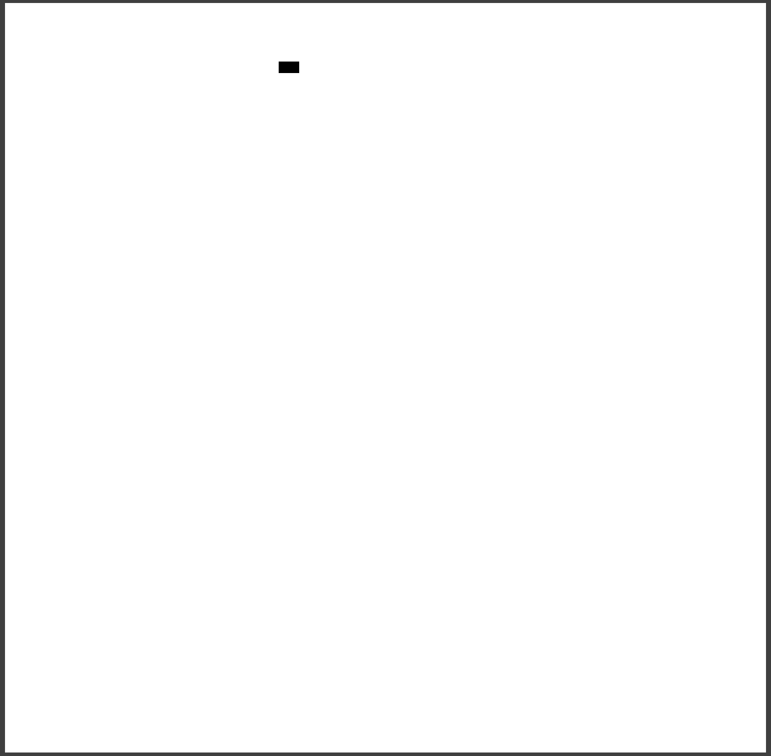
Dai and Muggleton, Abductive Knowledge Induction From Raw Data, IJCAI 2021.

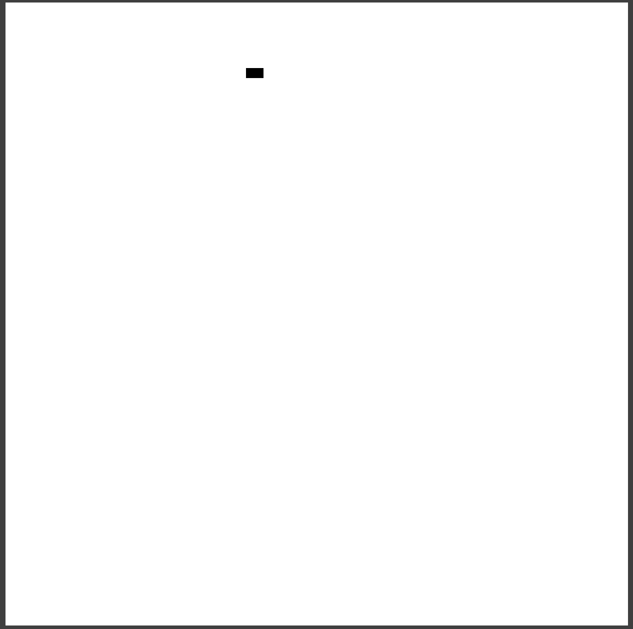
Ongoing: Oracle Bone Scripts



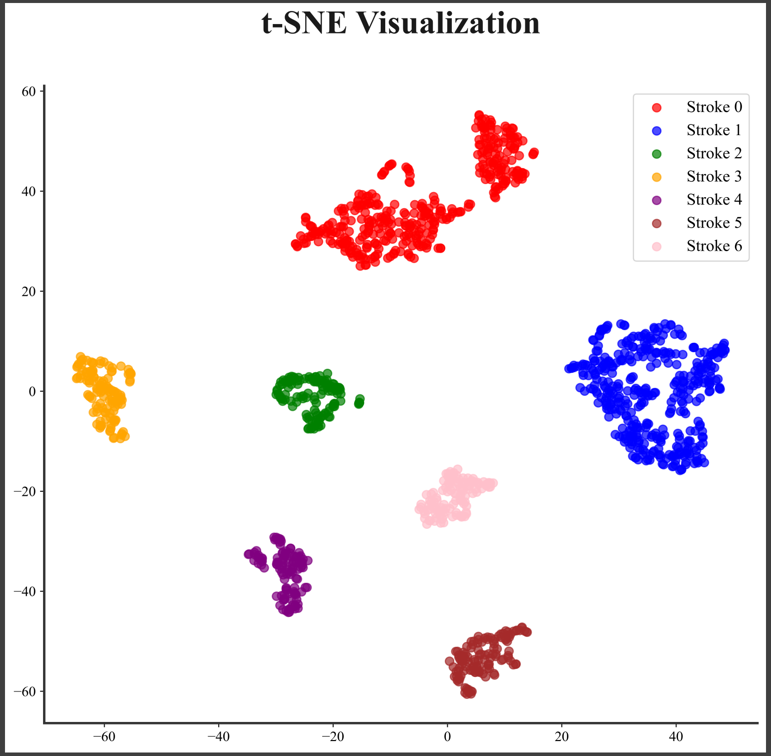


Oracle Bone Scripts - Procedural





Oracle Bone Scripts - Procedural



- 并（幷）：`[0,1,0,1,2]`
- 伐：`[0,1,4,3,6]`
- New symbols: `[0,1]`，`[2]`，`[4,3,6]`
    - （人、一、戈）

Bibliography

- W.-C. Hu, W.-Z. Dai, Y. Jiang, Z.-H. Zhou, **Efficient Rectification of Neuro-Symbolic Reasoning Inconsistencies by Abductive Reflection**, In: *Proceedings of the 39th Annual AAAI Conference on Artificial Intelligence* (AAAI'25), Philadelphia, PA, 2025.
- L. Tao, Y.-X. Huang, W.-Z. Dai and Y. Jiang. **Deciphering Raw Data in Neuro-Symbolic Learning with Provable Guarantees**. In: *Proceedings of the 38th Annual AAAI Conference on Artificial Intelligence* (AAAI'24), Vancouver, Canada, 2024.
- Y.-X. Huang, W.-Z. Dai, L.-W. Cai, S. H. Muggleton and Y. Jiang. **Fast Abductive Learning by Similarity-based Consistency Optimization**, In: *Proceedings of the 35th Conference on Neural Information Processing Systems* (NeurIPS'21), pp.26574-26584.
- W.-Z. Dai and S. H. Muggleton, **Abductive Knowledge Induction From Raw Data**, In: *Proceedings of the 30th International Joint Conference on Artificial Intelligence* (IJCAI'21), pp. 1845-1851.

Bibliography

- L.-W. Cai, W.-Z. Dai Y.-X. Huang, Y.-F. Li, S. H. Muggleton and Y. Jiang, **Abductive Learning with Ground Knowledge Base**, In: *Proceedings of the 30th International Joint Conference on Artificial Intelligence* (IJCAI'21), pp.1815-1821.
- W.-Z. Dai, Q. Xu, Y. Yu, and Z.-H. Zhou. **Bridging machine learning and logical reasoning by abductive learning**. In: *Advances in Neural Information Processing Systems 32* (NeurIPS'19) (Vancouver, Canada), 2019.
- Z.-H. Zhou. **Abductive learning: towards bridging machine learning and logical reasoning**. *Science China Information Sciences*, 62(7), 2019.

Job Opportunity @ Nanjing

- Nanjing University is seeking talented faculty and researchers to persue advancements in the field of classical symbolic AI, statistical relational AI and neuro-symbolic AI.
- We offer all kinds of positions, from postdoc to full professor.
- AI research @ Nanjing is leading by top-notch scholars, such as Stephen Muggleton FREng, Tan Tieniu FREng, and Zhi-Hua Zhou (current President of Trustee Board of IJCAI)