Effects of machine-learned logic theories on human comprehension in machine-human teaching

About

- Doctoral student supervised by Prof. Muggleton, Imperial
- Inductive Logic Programming, XAI and comprehensibility
- UK EPSRC Human-Like Computing Network
- EU TAILOR Network

What we do

- Inductive logic programming (ILP) (learning from entailment)

$$\forall e^+ \in E^+, B \cup H \models e^+$$

$$\forall e^- \in E^-, B \cup H \not\models e^-$$

- Utilise logic formalism: deduction, induction and abduction
- Meta-Interpretive Learning (MIL) (use second-order metarule “templates”)
  - Higher-order programs
  - Recursion
  - Predicate invention (compact theories, efficient learning, etc)

Questions

1) Learned logic programs always human comprehensible?

2) Model explanatory effects for predictive decision-making?

3) Explanatory effects in sequential settings?

L. Ai, S.H. Muggleton, C. Hocquette, M. Gromowski, and U. Schmid. Beneficial and harmful explanatory machine learning. Machine Learning, 2021. https://doi.org/10.1007/s10994-020-05941-0.

L. Ai, J. Langer, S.H. Muggleton, and U. Schmid. Explanatory machine learning for sequential human teaching. arXiv:2205.10250v1. 2022

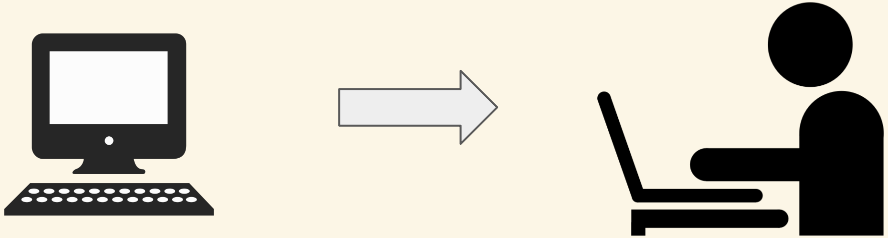
Content

- Contributions
- Framework
    - Explanatory effects
    - Cognitive window
    - Sequential teaching
    - MIL
- Empirical results
- Impacts
- Future work

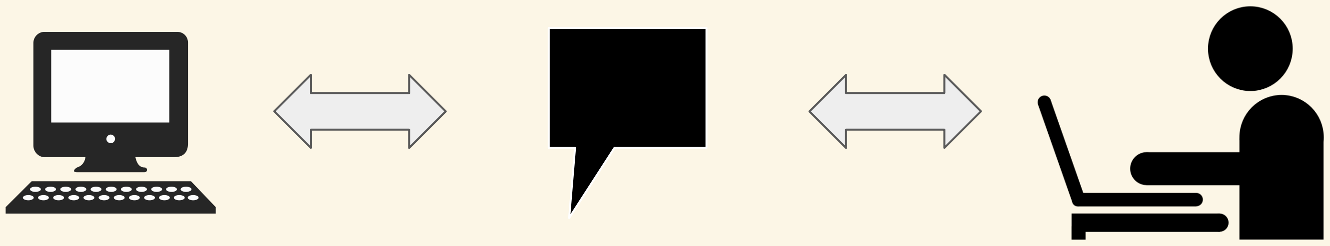
Contributions

- Operational definitions
  - explanatory effects
  - sequential teaching curricula
- Cognitive window framework
- Both ***beneficial*** and ***harmful*** explanatory effects
- ***Sequential teaching*** improvement
- Human strategy rediscovery and optimisation

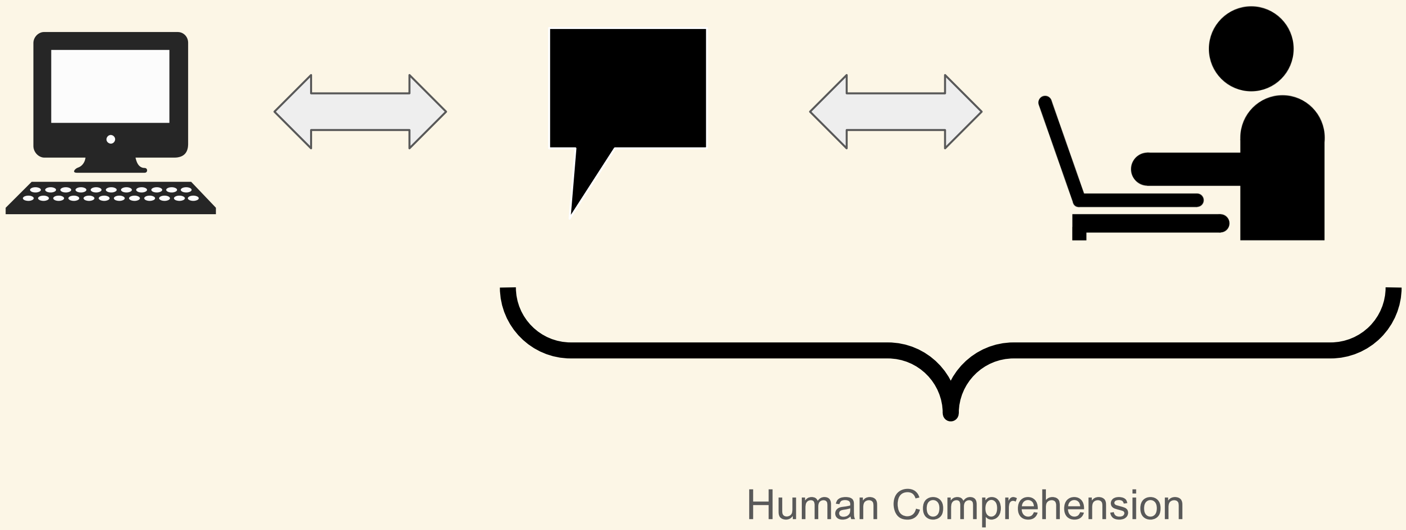
Novelty



Novelty



Novelty



Novelty

Comprehensibility = *Model complexity*

Novelty

Comprehensibility = *Model complexity* + *Human learning complexity*

Human comprehension

Definition 1 (Unaided human comprehension of examples, $C_h(D, H, E)$) Given that D is a logic program representing the definition of a target predicate, H is a human group and E is a set of examples of the target predicate. The unaided human comprehension of examples E is the mean accuracy with which a human $h \in H$ after a brief study of E and without further sight can classify new material sampled randomly from the domain of D.

Machine-aided comprehension

Definition 2 (Machine-explained human comprehension of examples, $C_{ex}(D, H, M(E))$): Given that D is a logic program representing the definition of a target predicate, H is a human group, $M(E)$ is a theory learned using machine learning algorithm M and E is a set of examples of the target predicate. The machine-explained human comprehension of examples E is the mean accuracy with which a human $h \in H$ after a brief study of an explanation based on $M(E)$ and without further sight can classify new material sampled randomly from the domain of D.

Explanatory effectiveness

$$E_{ex}(D, H, M(E)) = C_{ex}(D, H, M(E)) - C_h(D, H, E)$$

Effect = machine-aided comprehension - self-learning comprehension

Beneficial = positive effect

Harmful = negative effect

Two MIL systems

MIGO:

Sufficient and necessary BK

Positive examples only

Learns minimax algorithm

S.H. Muggleton and C. Hocquette. Machine discovery of comprehensible strategies for simple games using meta-interpretive learning. New Generation Computing, 37:203-217, 2019.

Two MIL systems

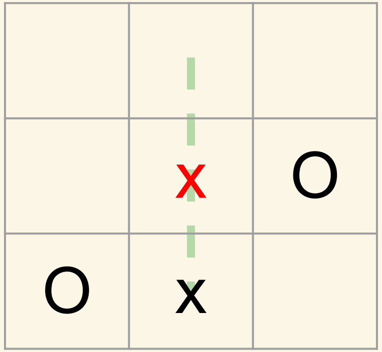
MIPlain (adapted MIGO):

BK involves an additional primitive

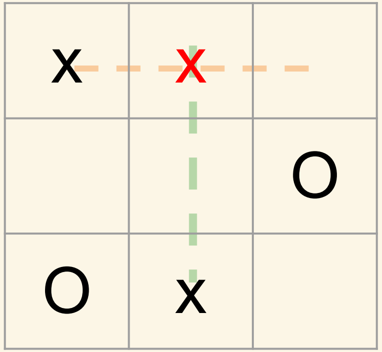
Positive and negative examples

Learns programs with less inferential cost

Extended BK



number_of_pairs(A, x, **1**)



number_of_pairs(B, x, **2**)

MIGO learned hypothesis

Depth	Rules
1	win_1(A,B):- win_1_1_1(A,B),won(B). win_1_1_1(A,B):-move(A,B),won(B).
2	win_2(A,B):-win_2_1_1(A,B),not(win_2_1_1(B,C)). win_2_1_1(A,B):-move(A,B), not(win_1(B,C)).
3	win_3(A,B):-win_3_1_1(A,B),not(win_3_1_1(B,C)). win_3_1_1(A,B):-win_2_1_1(A,B), not(win_2(B,C)).

MIPlain learned hypothesis

Depth	Rules
1	win_1(A,B):-move(A,B),won(B).
2	win_2(A,B):-move(A,B),win_2_1(B). win_2_1(A):-number_of_pairs(A,x,2), number_of_pairs(A,o,0).
3	win_3(A,B):-move(A,B),win_3_1(B). win_3_1(A):-number_of_pairs(A,x,1),win_3_2(A). win_3_2(A):-move(A,B),win_3_3(B). win_3_3(A):-number_of_pairs(A,x,0),win_3_4(A). win_3_4(A):-win_2(A,B),win_2_1(B).

Cognitive cost of predicates

pred: number_of_pairs(state1, x, N)

C(pred) = 1 + 6 + 1+1

Cognitive cost

q: win_2(s1(...),B)
…
number_of_pairs(s10(...), x, N)
number_of_pairs(s10(...), x, 1)
…
win_2(s1(...),s5(...))

Execution stack St

Cognitive cost of a program (datalog)

$$Cog(P, q) = \sum_{t \in St} C(t)$$

Where **P** is a program and **q** is a query.

Two ILP learned strategies

Clauses	Smaller program size (unfolded + no redundancy)	Lower cognitive cost
win_1	Both are same	Both are same
win_2	MIPlain	MIPlain
win_3	MIPlain	MIPlain

Does MIPlain guarantee a beneficial effect?

Primitive solution

Definition 7 **(Minimum primitive solution program, $\bar{M}_\phi(E)$):** Given a set of primitives ϕ and examples E, a datalog program learned from examples E using a symbolic machine learning algorithm $\bar{M}$ and a set of primitives $\phi' \subseteq \phi$ is a minimum primitive solution program $\bar{M}_\phi(E)$ if and only if for all sets of primitives $\phi'' \subseteq \phi$ where $|\phi''| < |\phi'|$ and for all symbolic machine learning algorithm M' using ϕ'', there exists no machine learned program $M'(E)$ that is consistent with examples E.

A minimum primitive solution uses a ***sufficient*** and ***necessary*** subset of a given BK.

Programs learn by MIGO = minimum primitive solutions

Human hypothesis space bound

Conjecture 1 **(Cognitive bound on the hypothesis space size, $B(P, H)$):** Consider a symbolic machine-learned datalog program P using p predicate symbols and m meta-rules each having at most j body literals. Given a group of humans H, $B(P, H)$ is a population-dependent bound on the size of hypothesis space such that at most n clauses in P can be comprehended by all humans in H and $B(P, H) = m^n p^{(1+j)n}$.

Human may only learn fraction of the rules presented.

Cognitive window

A balance between memory and computational complexity

D. Michie. “Experiments on the Mechanization of Game-Learning. 2-Rule-Based Learning and the Human Window.” Comput. J., pages 105-113, 1982.

Cognitive window

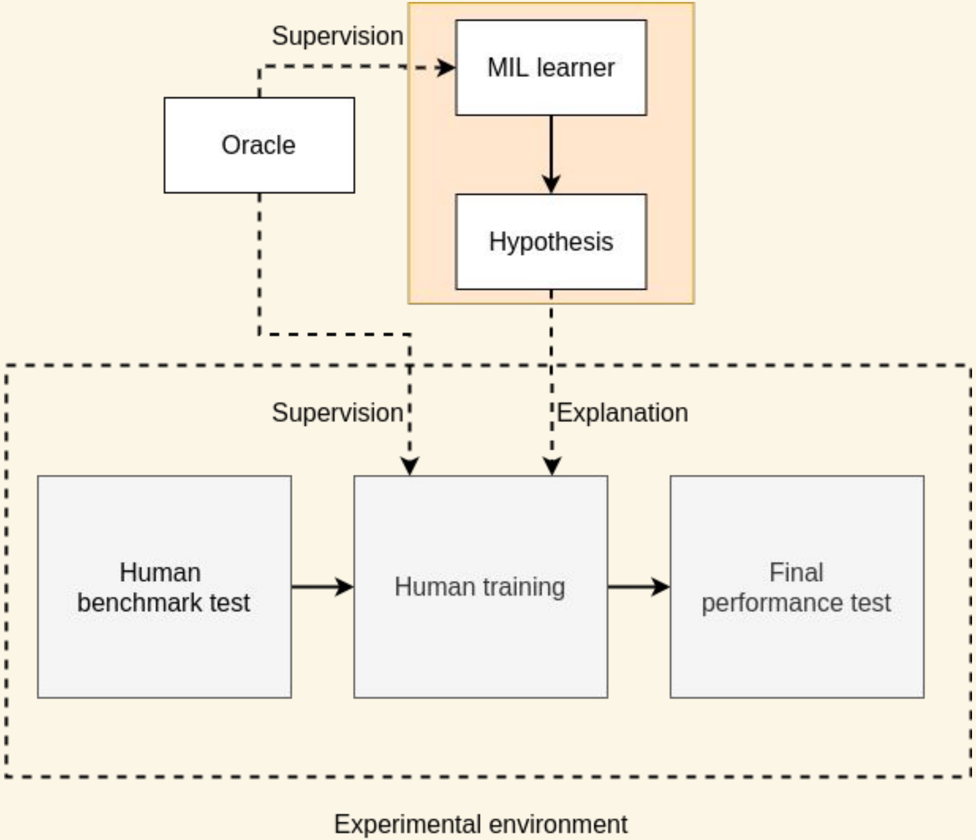
A comprehensible program 1) cannot be textually complex for human learning and 2) must provide “shortcuts” for human execution.

1. $E_{ex}(D, H, M(E)) < 0$ if $\underline{|S| > B(M(E), H)}$
2. $E_{ex}(D, H, M(E)) \leq 0$ if $\underline{Cog(M(E), x) \geq CogP(E, \bar{M}, \phi, x)}$ for queries x that $h \in H$ have to perform after study

Where **S** is the hypothesis space associated with M(E) and **CogP** computes

cognitive cost of primitive solutions which is equivalent to Cog for datalog programs

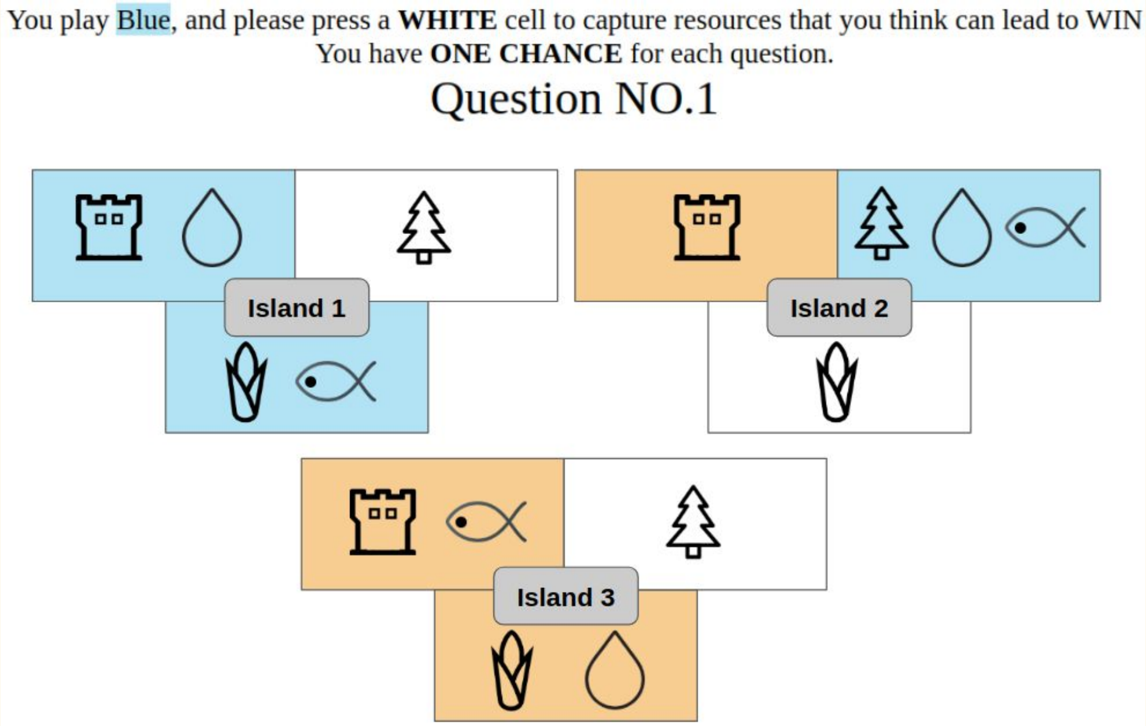
Comprehensibility test



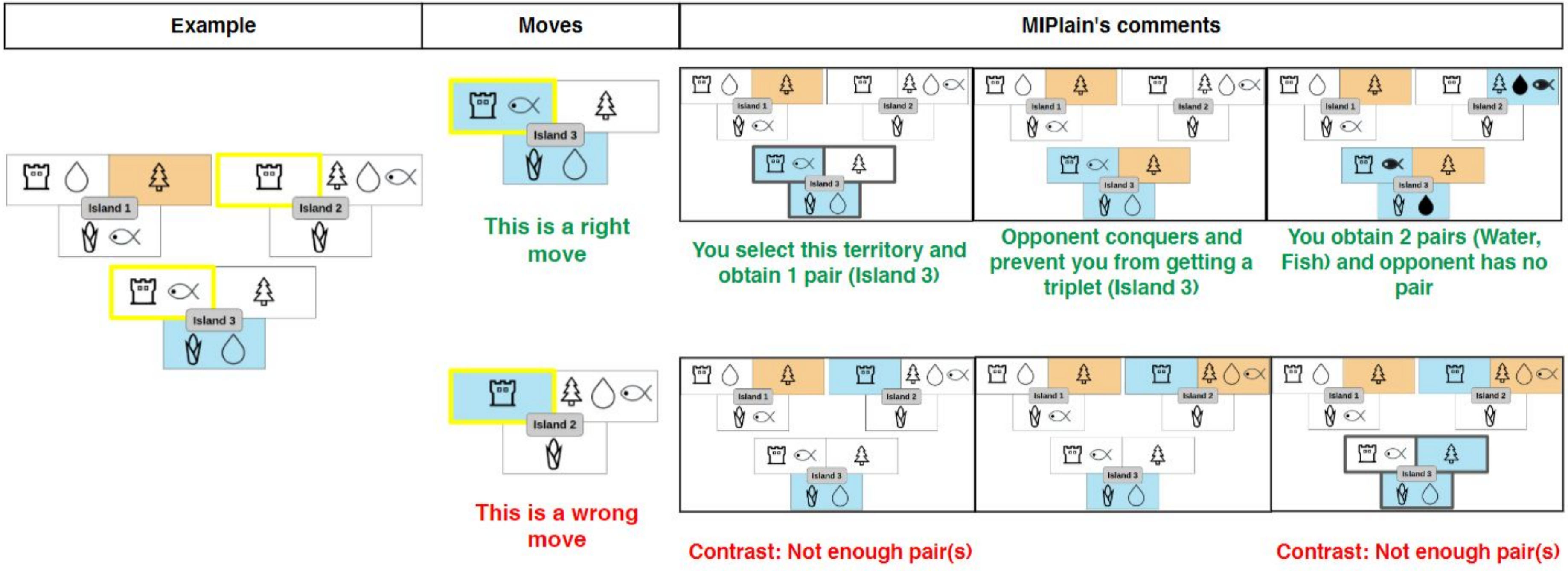
Experimental challenges

- Clarity of interface and task description
- Avoid prematurely exposing materials
- Avoid ceiling effect
- Preserve same problem complexity
- Alter spatial and representational arrangement

Materials


You play Blue, and please press a WHITE cell to capture resources that you think can lead to WIN
You have ONE CHANCE for each question.
Question NO.1
Island 1
Island 2
Island 3

Explanations



MIGO learned hypothesis

Depth	Rules
1	`win_1(A,B):- win_1_1_1(A,B),won(B).` `win_1_1_1(A,B):-move(A,B),won(B).`
2	`win_2(A,B):-win_2_1_1(A,B),not(win_2_1_1(B,C)).` `win_2_1_1(A,B):-move(A,B), not(win_1(B,C)).`
3	`win_3(A,B):-win_3_1_1(A,B),not(win_3_1_1(B,C)).` `win_3_1_1(A,B):-win_2_1_1(A,B), not(win_2(B,C)).`

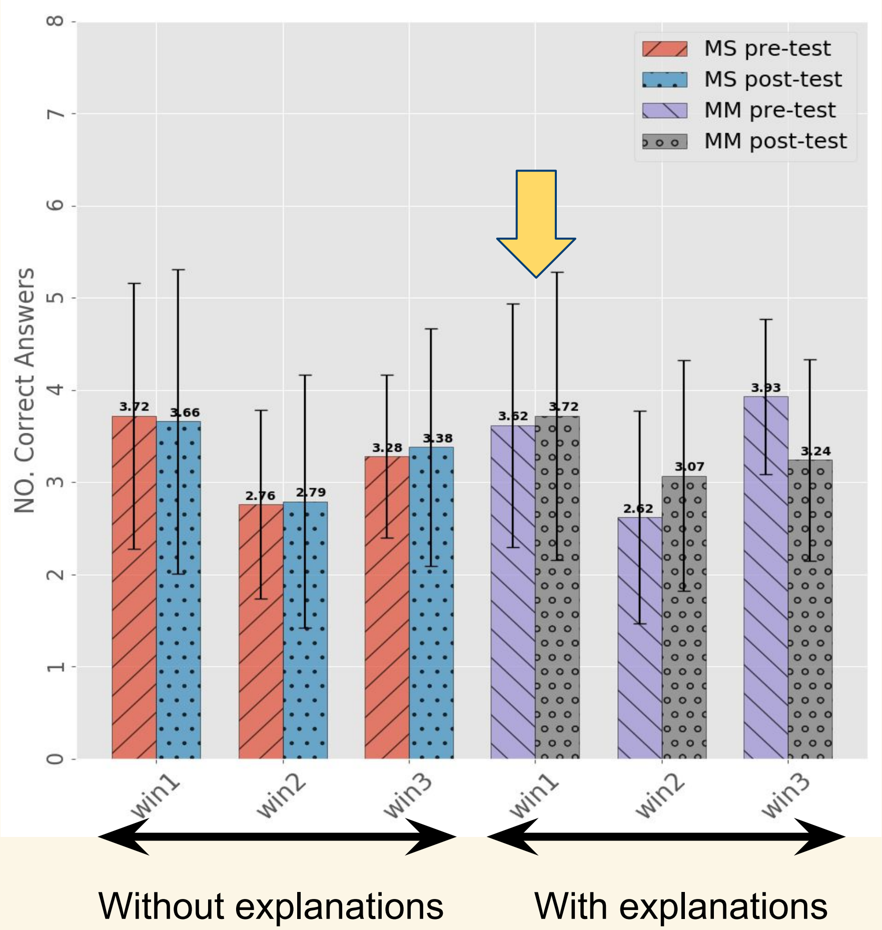
MIPlain learned hypothesis

Depth	Rules
1	`win_1(A,B):-move(A,B),won(B).`
2	`win_2(A,B):-move(A,B),win_2_1(B).` `win_2_1(A):-number_of_pairs(A,x,2), number_of_pairs(A,o,0).`

3	`win_3(A,B):-move(A,B),win_3_1(B).`
	`win_3_1(A):-number_of_pairs(A,x,1),win_3_2(A).`
	`win_3_2(A):-move(A,B),win_3_3(B).`
	`win_3_3(A):-number_of_pairs(A,x,0),win_3_4(A).`
	`win_3_4(A):-win_2(A,B),win_2_1(B).`

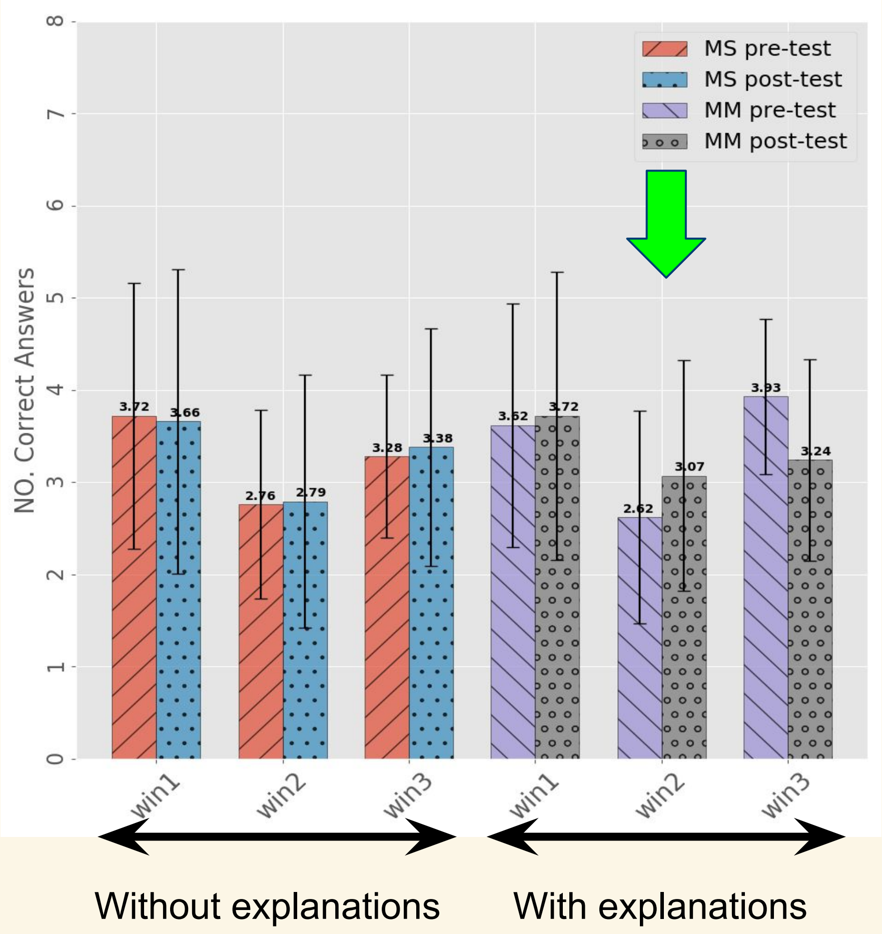
Yellow (no significant effect)

No execution shortcuts



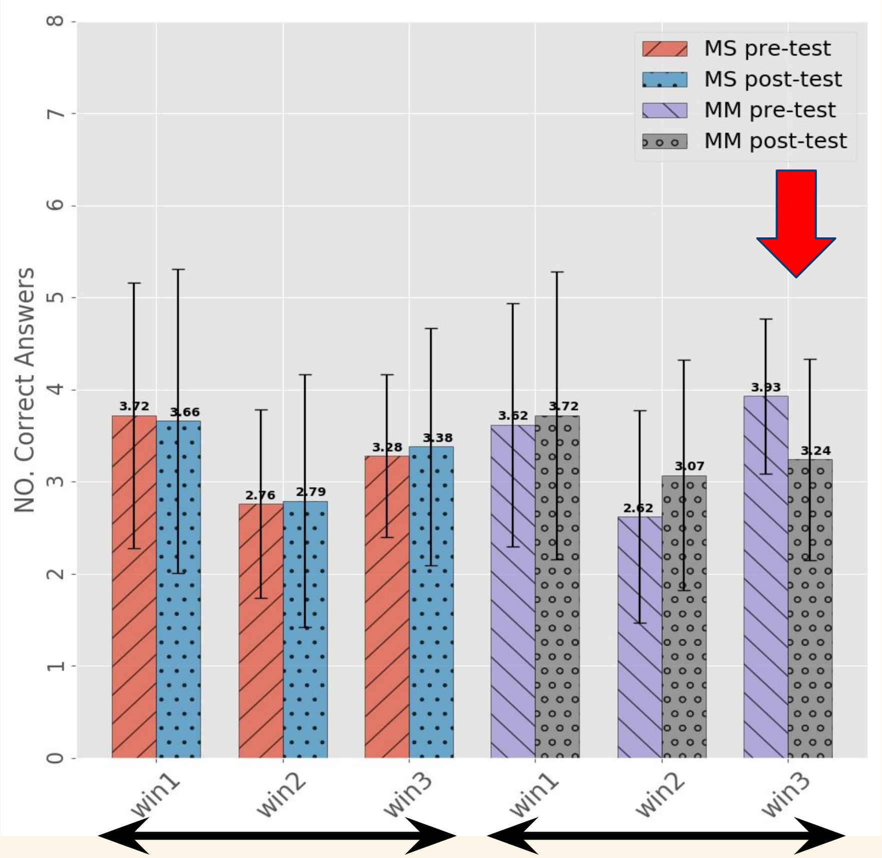
Green (beneficial effect)

Satisfaction of cognitive window



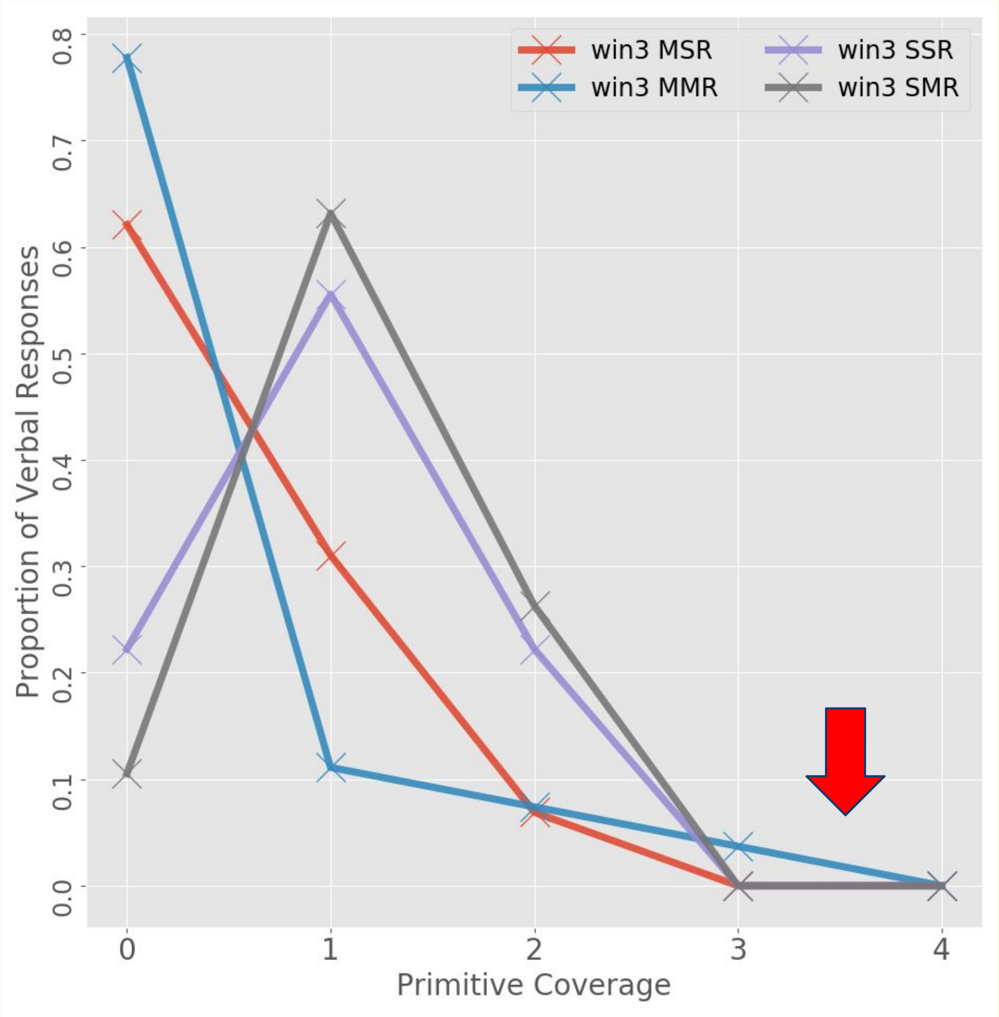
Red (harmful effect)

Only a fraction of the explanation is learned



Without explanations

With explanations



Low frequency of high coverage (key predicates) responses

Empirical results summary

1) Satisfaction of the cognitive window = ***beneficial*** effect

2) Satisfaction of Cognitive window requires

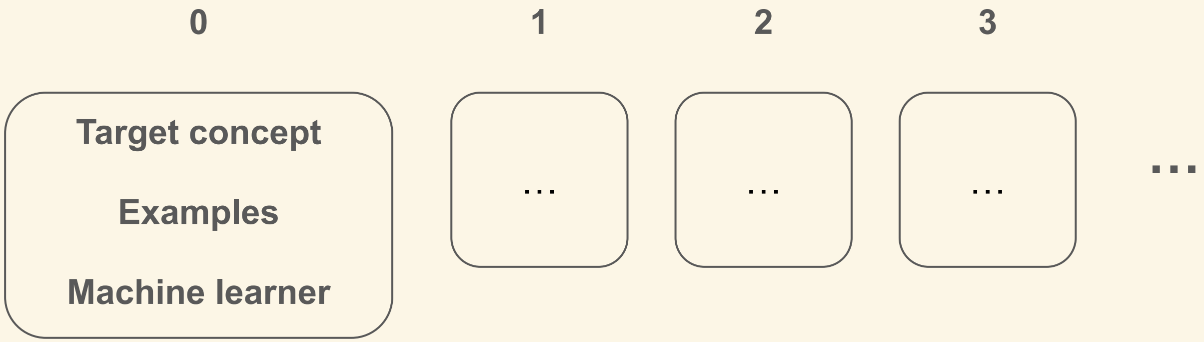
   a) ***Low*** descriptive complexity

   b) Appropriate ***background/primitives*** to allow efficient execution

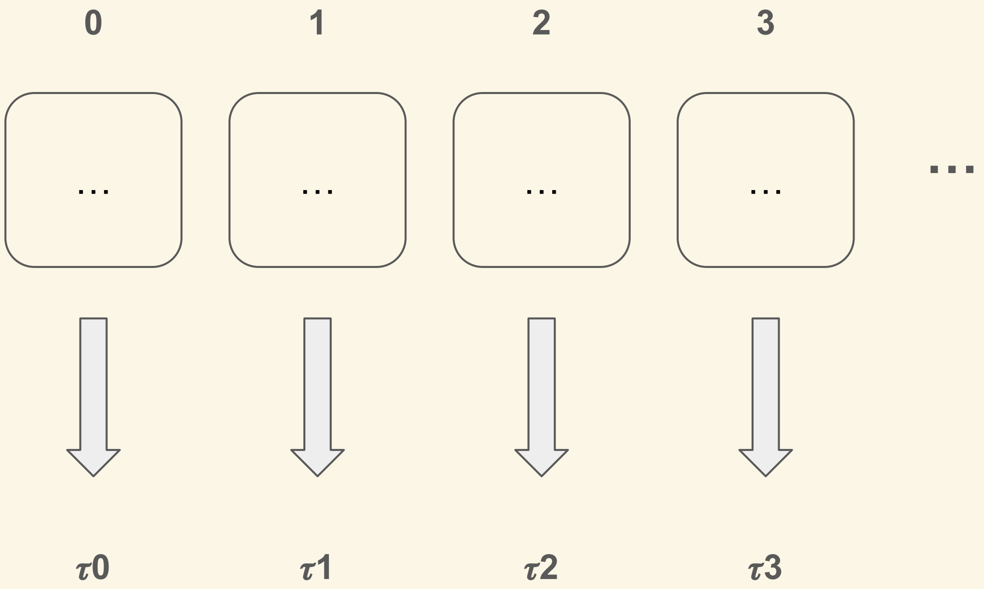
3) Confirm bound on human learning hypothesis space

Can we break a concept into sub-concepts and teach incrementally?

Sequential teaching curriculum



Comprehension of a sequential teaching curriculum



$$C_{seq}(T, H)$$

Comparison of curriculum comprehension

$$E_{seq}(C_1, C_2, D) = \tau_1 - \tau_2$$

Where $\boldsymbol{\tau}$**1** and $\boldsymbol{\tau}$**2** are scores of concept D from curriculum comprehension C1 and C2**.**

Sample complexity

Proposition 2 (Sample complexity [Cropper, 2017]). Given p predicate symbols, m metarules in $\mathcal{M}^i_j$, and a clause bound n, MIL has sample complexity s with error ϵ and confidence δ:

$$s \geq \frac{1}{\epsilon}(n \ln(m) + (j+1)n \ln(p) + \ln \frac{1}{\delta})$$

Cropper, A. Efficiently learning efficient programs. PhD thesis, Imperial College London, UK, 2017.

Sequential teaching curriculum improvement

For a concept D in two curricula (C1 and C2), $E_{seq}(C_1, C_2, D) > 0$

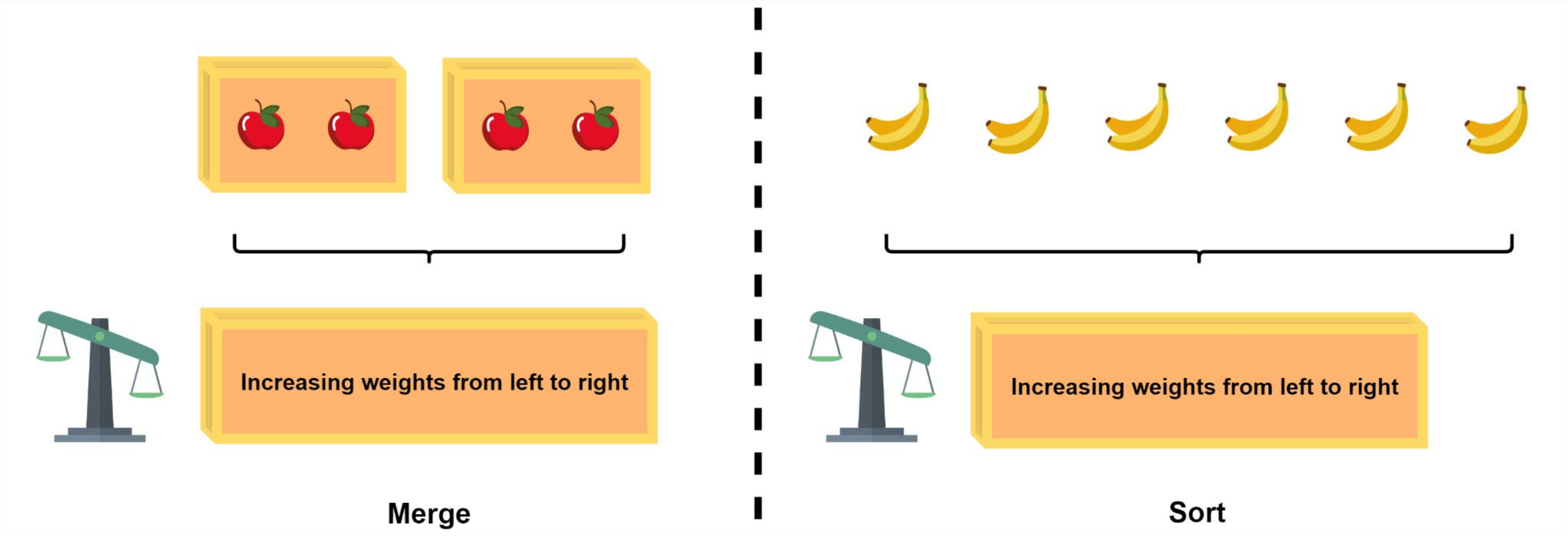
when:

$$n \ln(p) < (n + k) \ln(p + c)$$

(LHS) sample complexity of D in C1

(RHS) sample complexity of D in C2

Sequential teaching of sorting



Learning merge sort variant

MetagolO:

BK involves composite objects and primitives

Learns a program to operate a mini robot

Minimises both textual and resource complexity

A. Cropper and S. H. Muggleton. Learning efficient logical robot strategies involving composable objects. In Proceedings of the 24th International Conference on Artificial Intelligence, page 3423–3429, 2015.

Learning merge sort variant (MetagolO)

Iteration 1

[4, 6, 5, 2, 3, 1]

Iteration 2

[4 < 6, 2 < 5, 1 < 3]

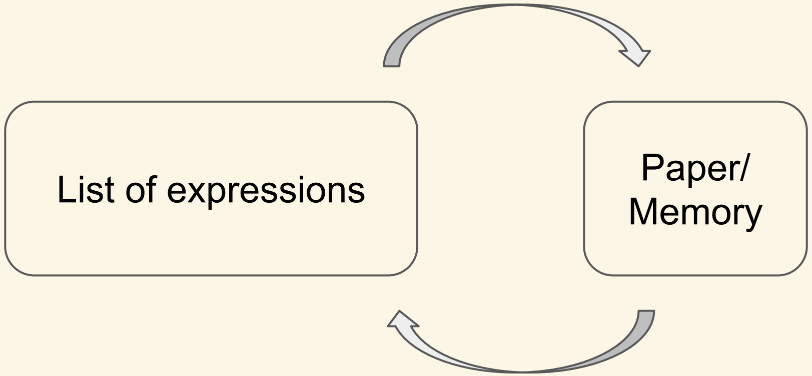
Iteration 3

[2 < 4 < 5 < 6, 1 < 3]

Iteration 4

[1 < 2 < 3 < 4 < 5 < 6]

Use “left hand” and “right hand” to write expressions (merging)



Restore/recycle expressions

Learning efficient sorting algorithms (MetagolO)

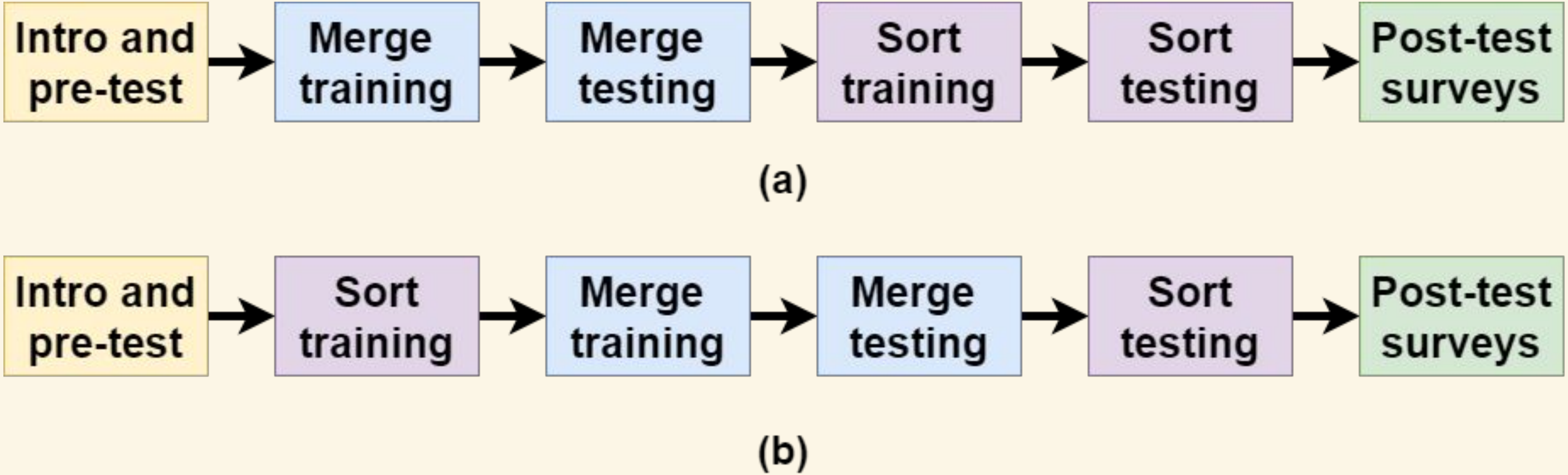
Definition	Rules

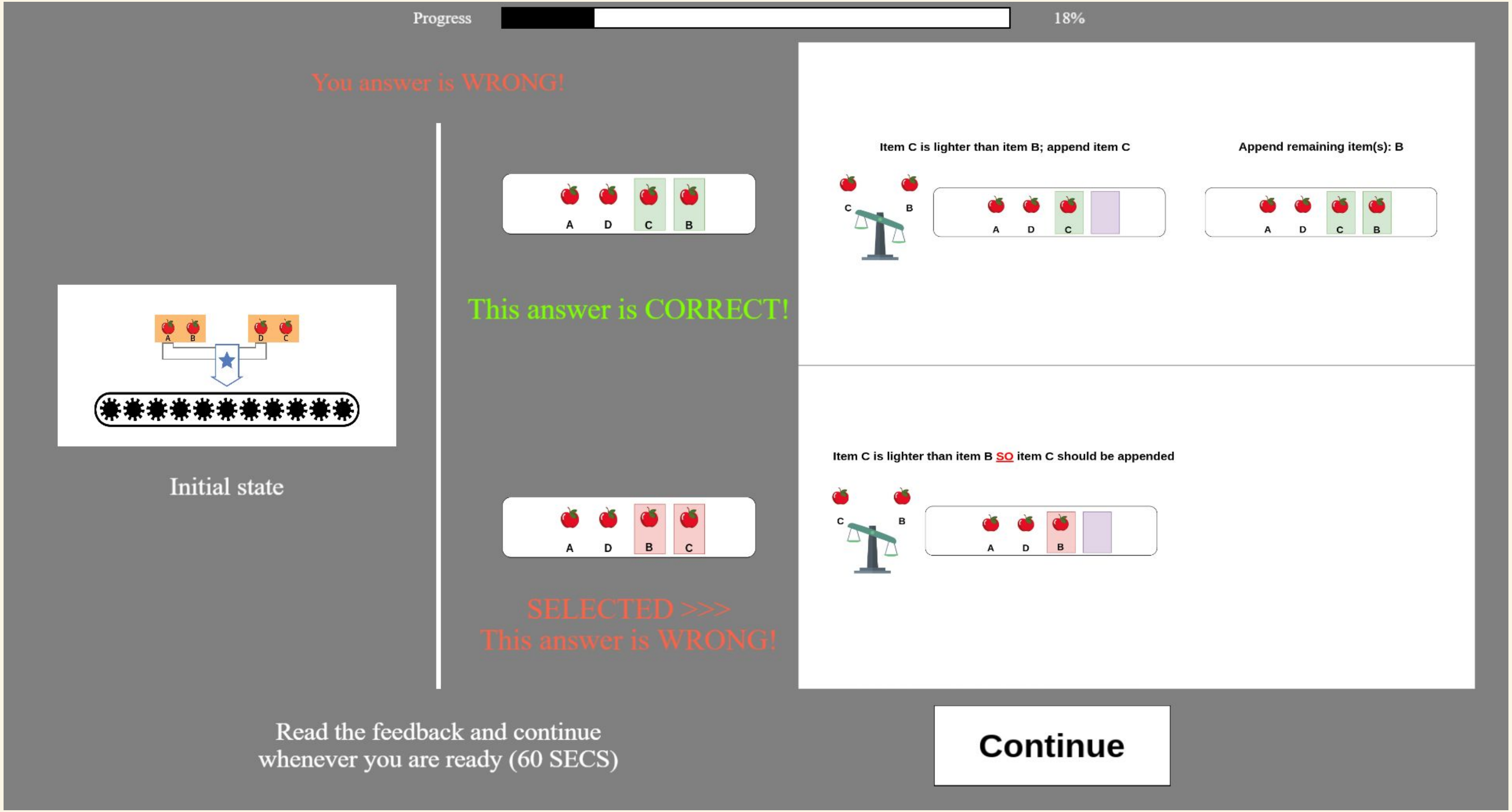
merger/2	`merger(A,B):-parse_exprs(A,C),merger_1(C,B).`
	`merger_1(A,B):- compare_nums(A,C),merger_1(C,B)`
	`merger_1(A,B):-compare_nums(A,C),drop_bag_remaining(C,B).`

sorter/2 (after learning merger/2)	`sorter(A,B):-merger(A,C),sorter(C,B).`
	`sorter(A,B):-recycle_memory(A,C), sorter(C,B).`
	`sorter(A,B):-single_expr(A,C), single_expr(C,B).`

sorter/2 (without learning merger/2)	`sorter(A,B):-parse_exprs(A,C),sorter(C,B).`
	`sorter(A,B):-compare_nums(A,C), sorter(C,B).`
	`sorter(A,B):-drop_bag_remaining(A,C), sorter(C,B).`
	`sorter(A,B):-recycle_memory(A,C), sorter(C,B).`
	`sorter(A,B):-single_expr(A,C), single_expr(C,B).`

Sequential teaching of sorting

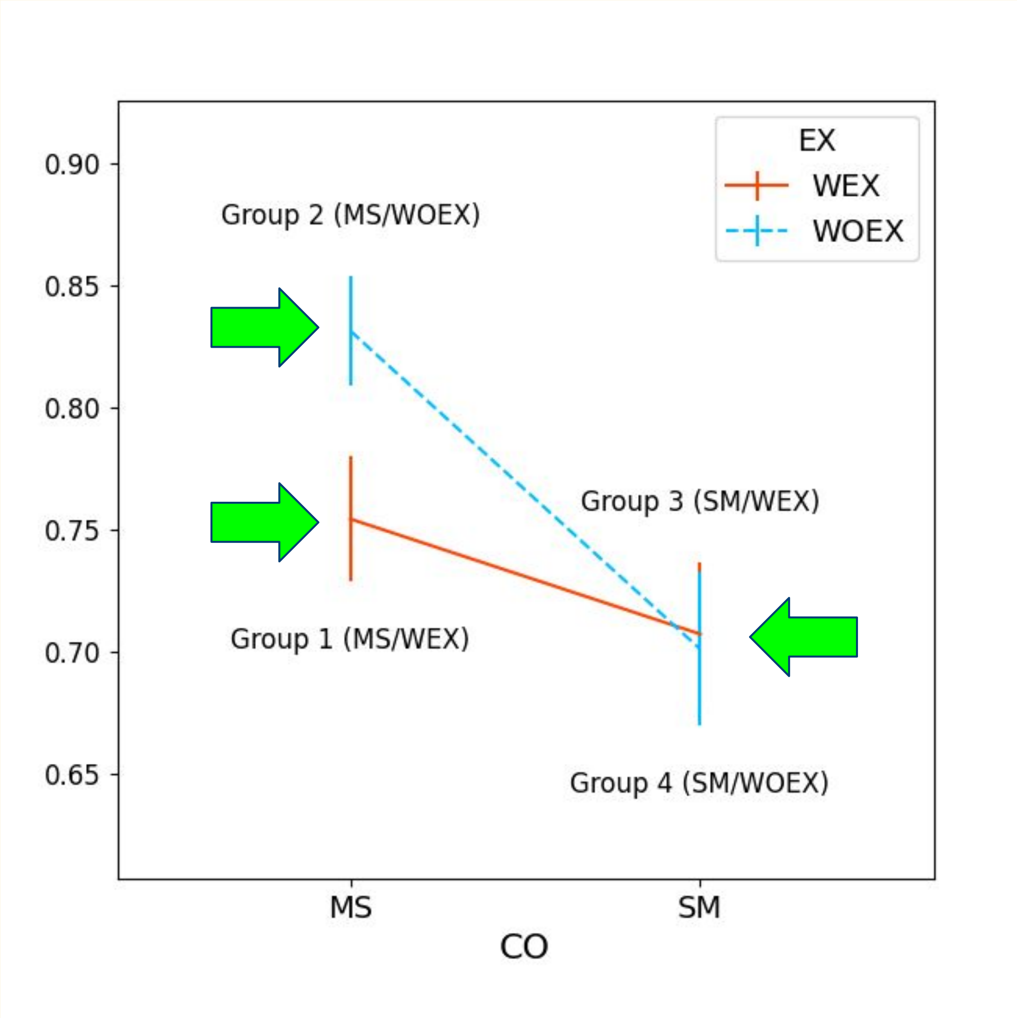




Explanations provided only for merging.

Sorting:

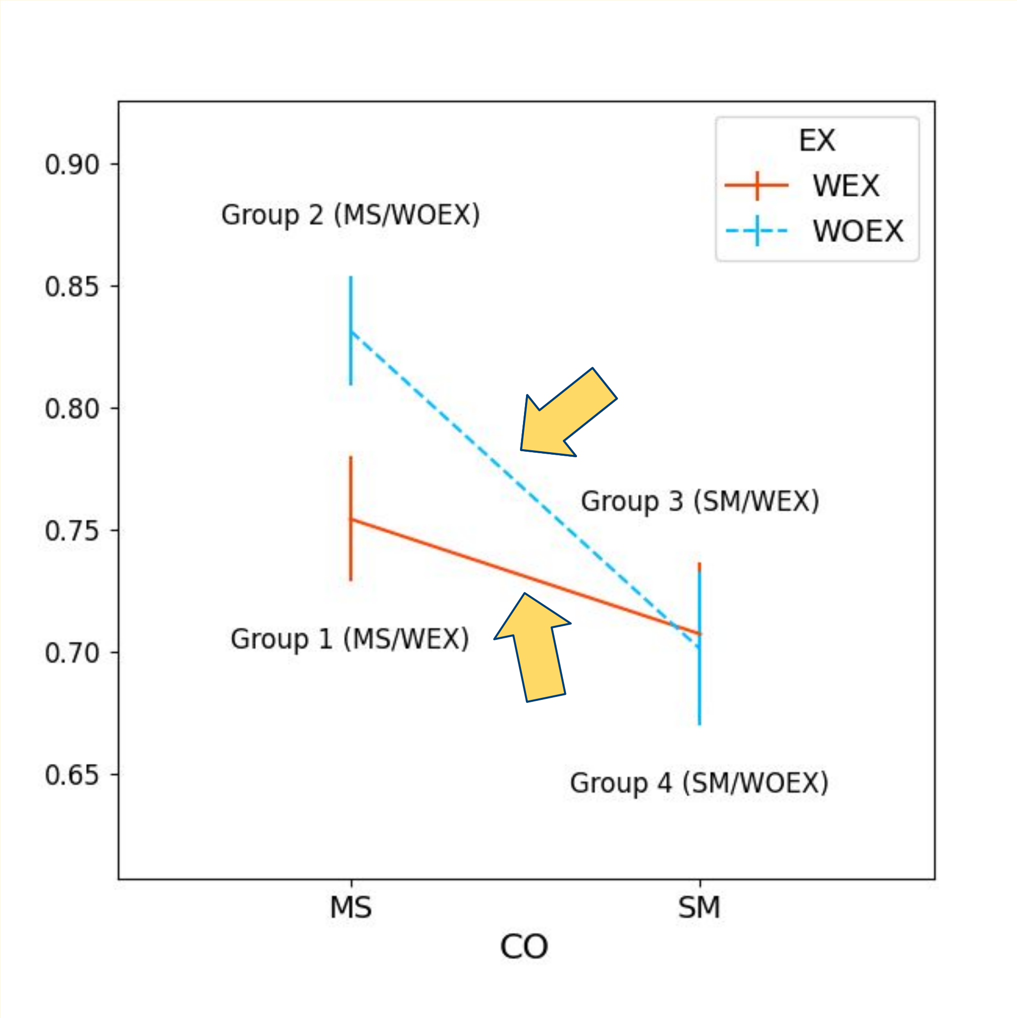
Incremental learning is beneficial.



Sorting:

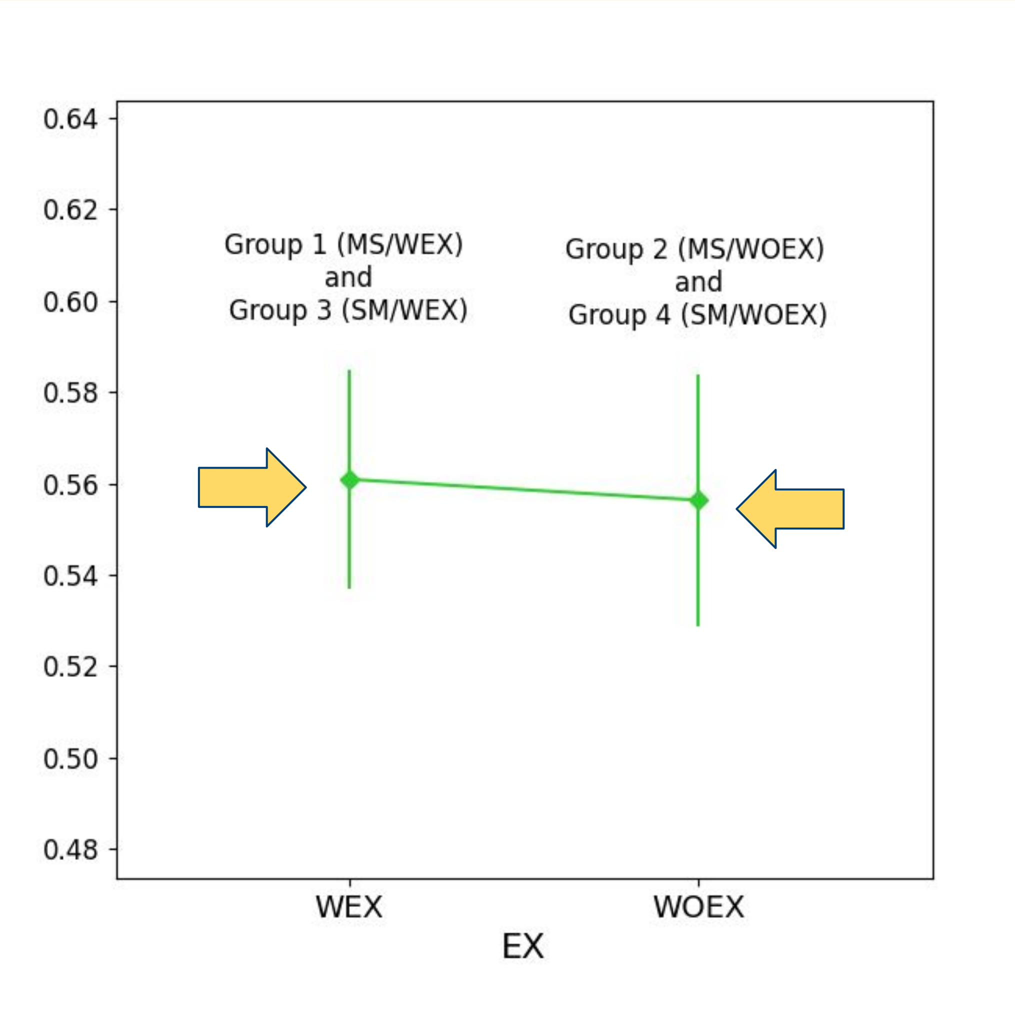
Incremental learning is beneficial.

However, explanations have no significant effect.



Merging:

Explanations have no significant effect.



PS \ Categories	*BS*	*DS*	*IS*	*MS*	*QS*	*Hybrid*	*Other*
Group 1 (MS/WEX)	–	–	–	–	–	–	–
Training	.012	.075	.150	.000	.175	.162	.425
Performance test	.056	.094	.162	.025	.238	.175	.250
Differences	.044	.019	.012	.025	**.063**	.013	-.175
Group 2 (MS/WOEX)	–	–	–	–	–	–	–
Training	.000	.062	.162	.025	.162	.225	.362
Performance test	.012	.038	.181	.100	.194	.181	.294
Differences	.012	-.024	.019	**.075**	.032	-.044	-.068
Group 3 (SM/WEX)	–	–	–	–	–	–	–
Training	.012	.050	.088	.038	.225	.175	.412
Performance test	.019	.138	.100	.025	.244	.119	.356
Differences	.007	**.088**	.012	-.013	.019	-.056	-.056
Group 4 (SM/WOEX)	–	–	–	–	–	–	–
Training	.000	.079	.184	.026	.158	.237	.316
Performance test	.013	.099	.243	.053	.158	.237	.197
Differences	.013	.020	**.059**	.027	.000	.000	-.119

Strategy rediscovery and optimisation

Incremental curriculum => ***more efficient*** sorting strategy (quick sort, merge sort).

Explanations => ***higher performance*** of adapted sorting strategy (quick sort, dictionary sort).

Empirical results summary

1) ***Incremental*** concept complexity = ***beneficial*** effect
2) Partial confirmation of cognitive window
    a) No executional shortcut for merging is provided
    b) No significant improvement of cognitive cost
3) Human novel ***rediscovery*** of algorithms as result of ***explanations*** and ***incremental*** teaching

Impact

- Evolution of human skill training scheme in industry 4.0
- Increasingly accessible online teaching platforms
- Comprehensibility = computability?

Future work

Impact of background knowledge on comprehension

- BK that reduces sample complexity vs. execution cost
- Appropriate primitives to optimise comprehension

Future work

For improving human performance

- Estimation of human errors/implicit knowledge
- Present tailored explanations to address them

Future work

Comprehensibility benchmark platform

- Involvement of psychologists
- Dynamically recruit quality participants to take tests
- Provide an interface for systems to evaluate comprehension scores

Q & A