Doing Experiments and Revising Rules with Natural Language and Probabilistic Reasoning

NeurIPS 2024

Top Piriyakulkij[1], Cassidy Langenfeld[1], Tuan-Anh Le[2], Kevin Ellis[1]

Cornell University[1], Google[2]

Goal

Give a model of how to infer natural language rules by doing experiments

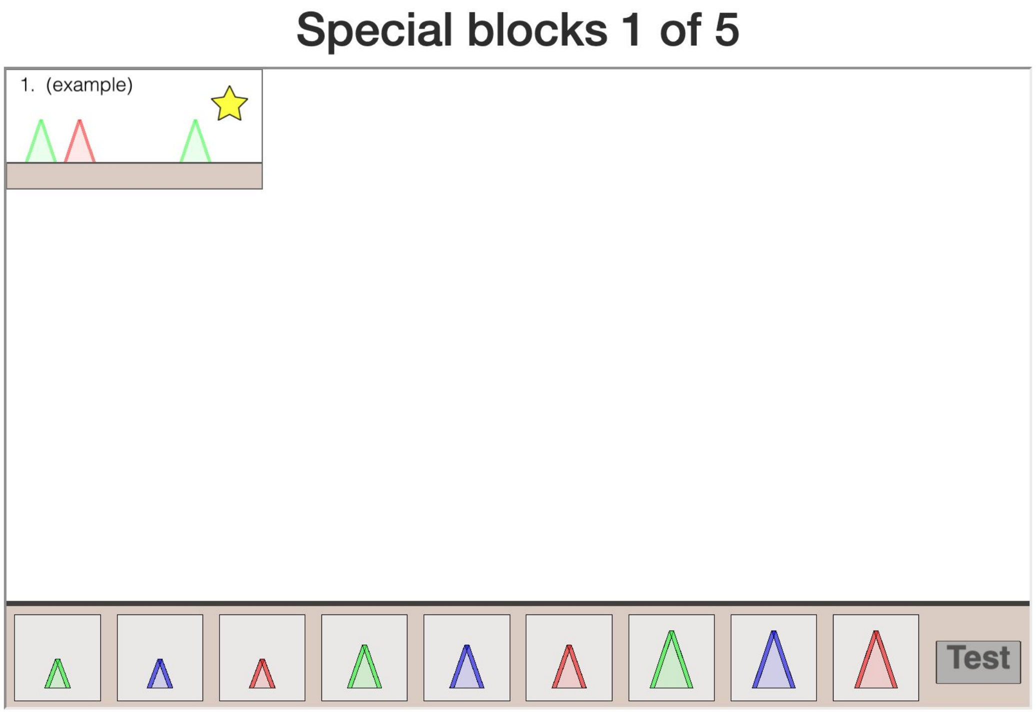
Goal

Give a model of how to infer natural language rules by doing experiments

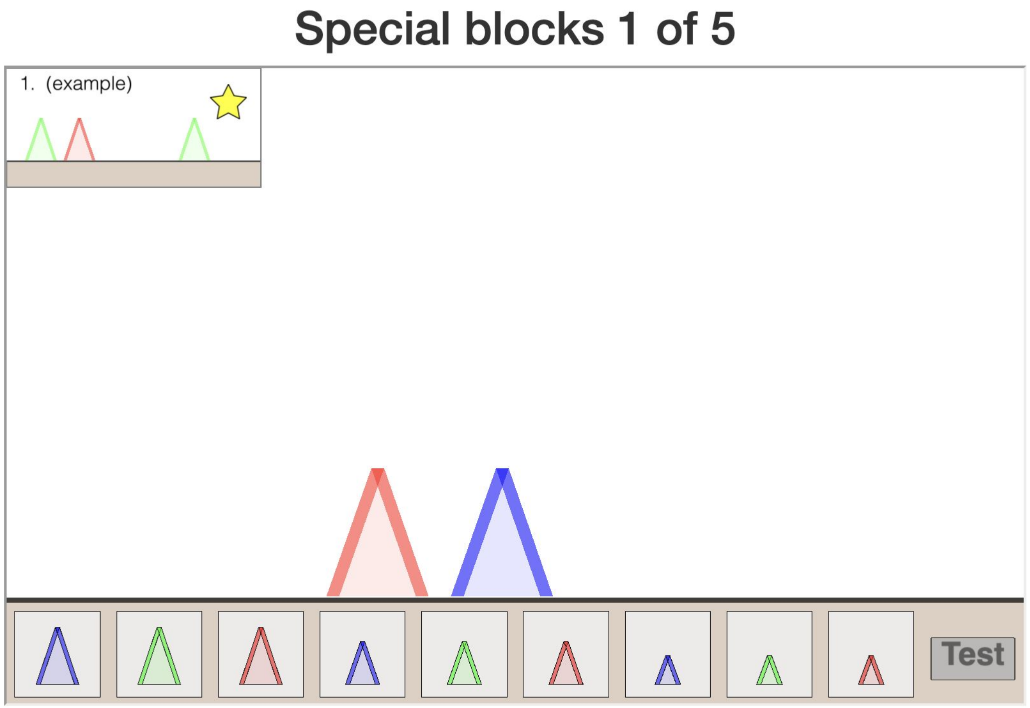
Motivation: how scientists learn

- Come up with theories, plan and perform experiments, then revise theories

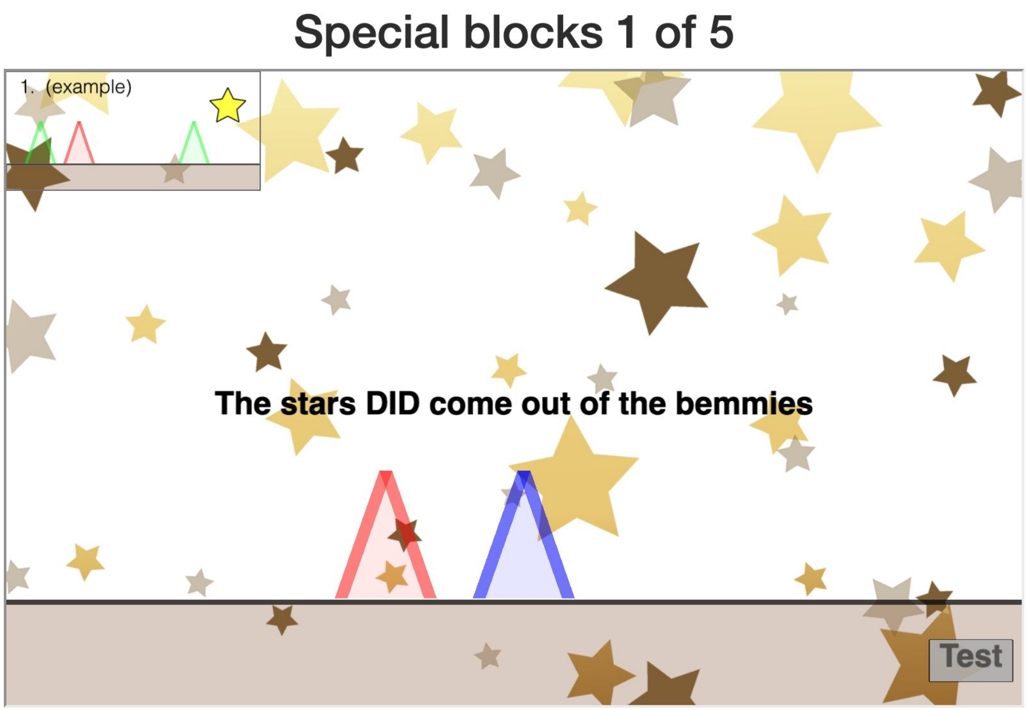
Domain: Zendo, an induction game



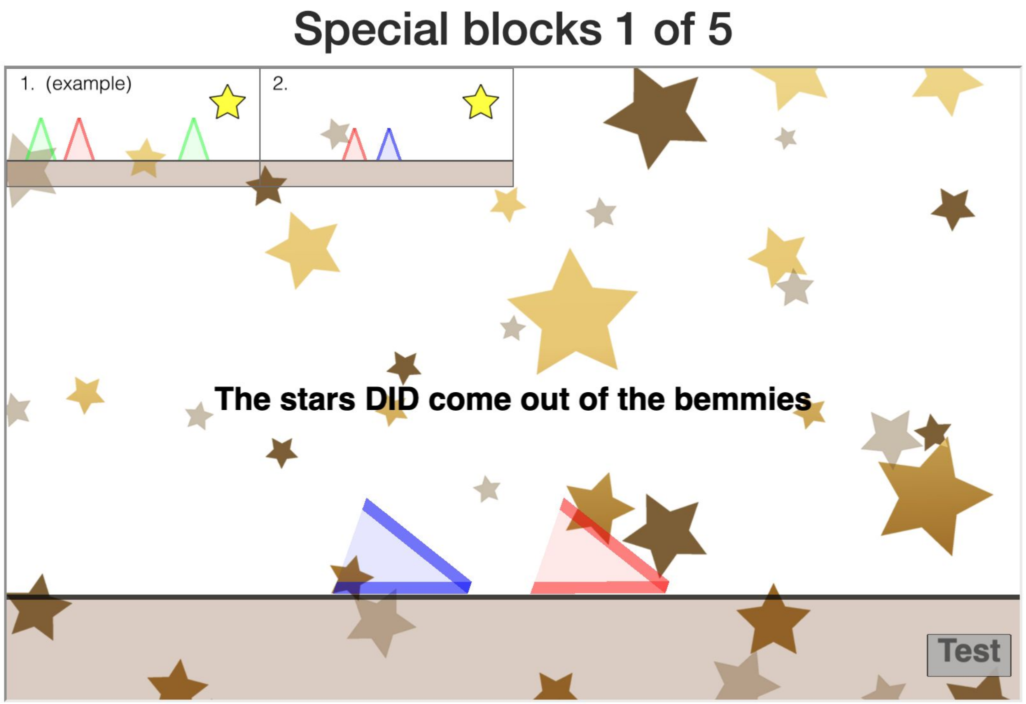
Domain: Zendo, an induction game



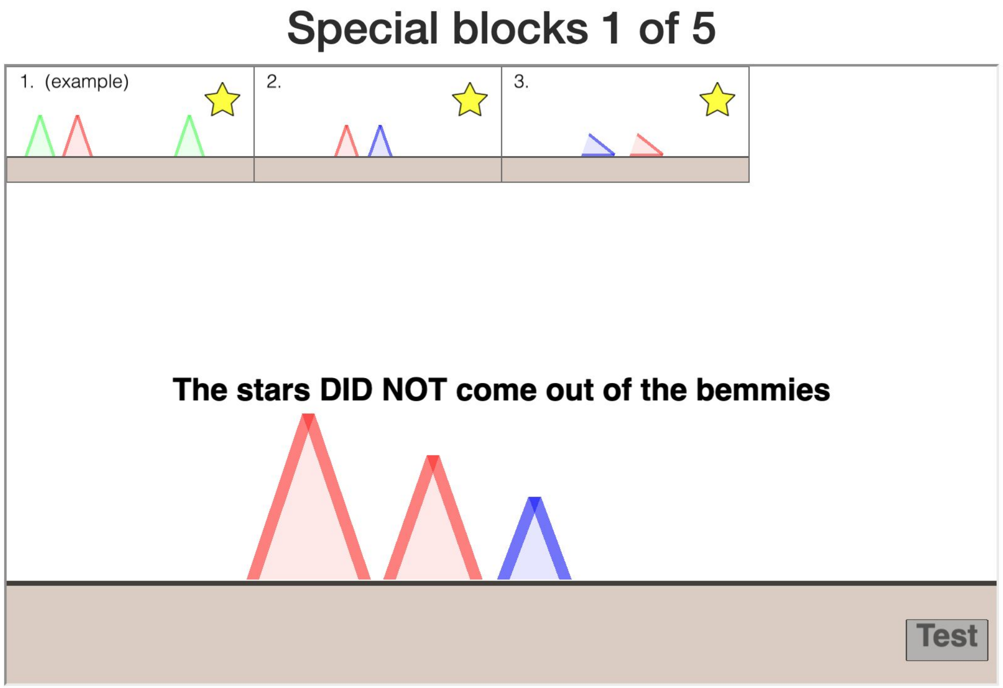
Domain: Zendo, an induction game



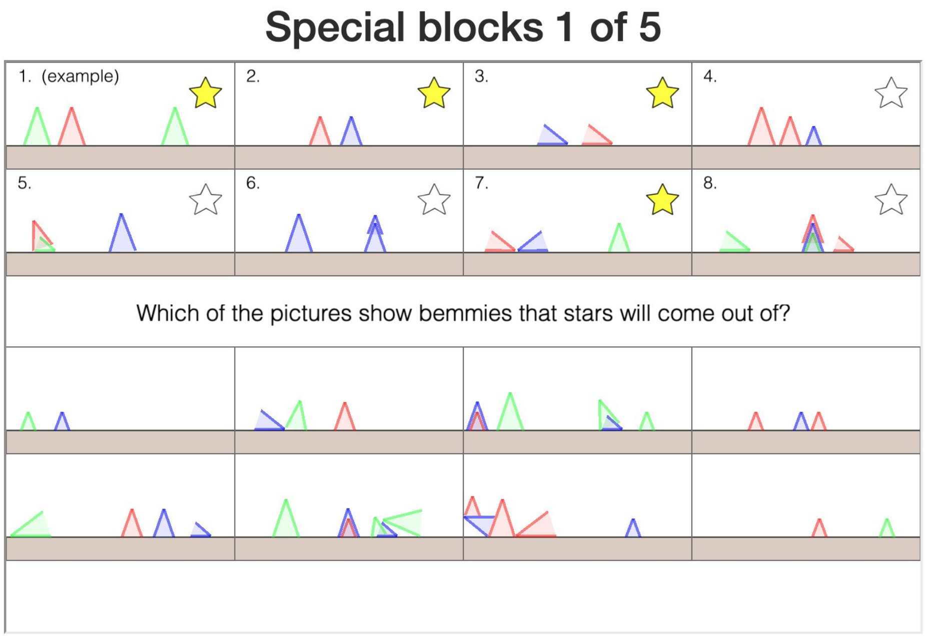
Domain: Zendo, an induction game



Domain: Zendo, an induction game



Domain: Zendo, an induction game

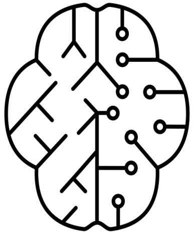


Inference

How should we perform inference?

One solution: Batch Inference

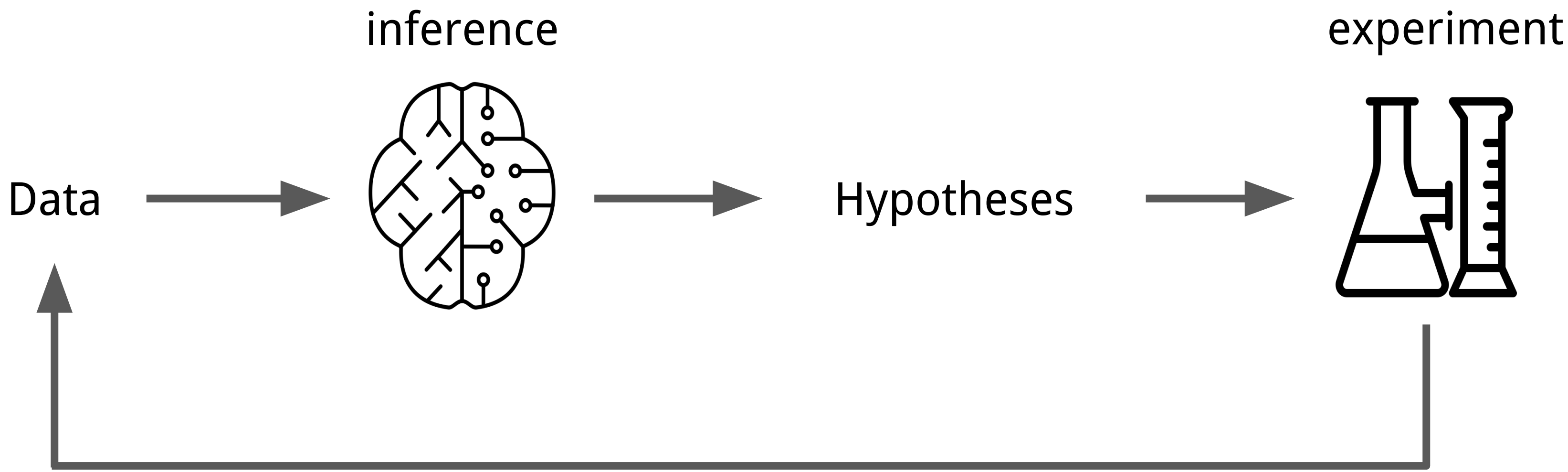
inference

All the data →  → Many Hypotheses

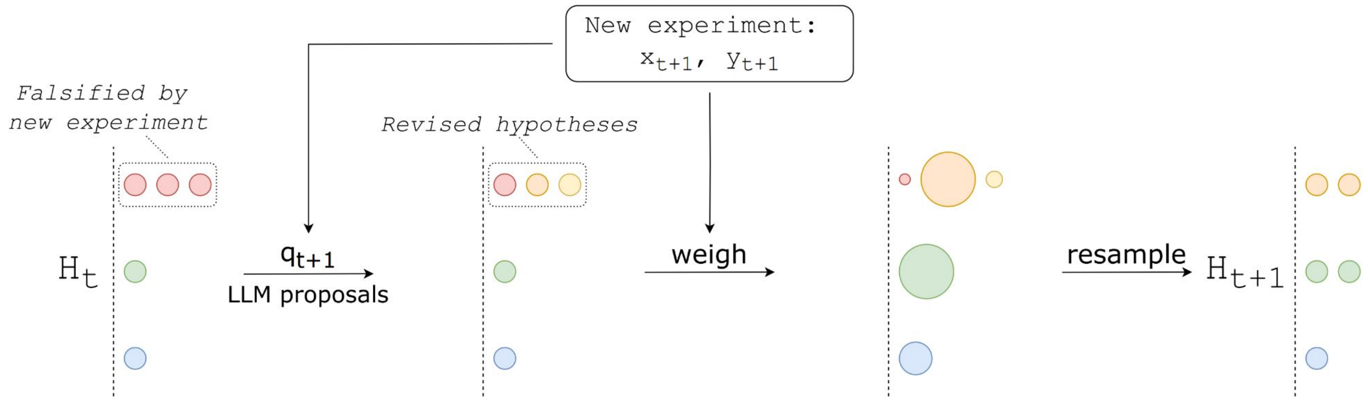
Inference

How should we perform inference?

The better and more human-like solution: Online Inference



Online inference with Sequential Monte Carlo (SMC)



LLM-SMC-S

Applying LLM-based kernel q to every particle is very expensive, however

- Introduce a novel variant of SMC: LLM-SMC-S

Procedure: LLM-SMC-S (A.3). Given H_t, W_t where $p(h|x_{1:t}, y_{1:t}) \approx \sum_i w_t^{(i)} \mathbb{1}\left[h = h_t^{(i)}\right]$:

1. Define unnormalized target densities $\gamma(h) = p(h, y_{1:t}, x_{1:t})$ and $\gamma'(h) = p(h, y_{1:t+1}, x_{1:t+1})$.
2. Sample $h' \sim q_{t+1}(\cdot|H_t, x_{1:t+1}, y_{1:t+1})$ (i.e., using LLM to revise hypotheses)
3. Compute the weight w' for h' following

$$w' = \frac{A(h', H_t, W_t)}{q_{t+1}(h'|H_t, x_{1:t+1}, y_{1:t+1})} \text{ where } A(h', H_t, W_t) = \frac{1}{n}\sum_{i=1}^{n} w_t^{(i)} \frac{\gamma'(h')r(h_t^{(i)}|h')}{\gamma(h_t^{(i)})} \quad (5)$$

   with the reverse kernel $r(h|h')$ defined as uniform up to strings of a maximum length.
4. Repeat steps 2-3 (sampling/weighing) a total of n times, and normalize the weights. Optionally, resample to generate an unweighted posterior (we always resample).
5. Output: H_{t+1} and W_{t+1}, formed from n samples of h', w' with w' normalized from step 4, which approximate $p(h|x_{1:t+1}, y_{1:t+1})$.

The correctness of the above procedure is most easily understood using the following definition:

Definition: Proper Weighting [27]. Let $\gamma(h)$ be an unnormalized target density, which we can evaluate. Let the corresponding normalized target density be $\pi(h) = \frac{\gamma(h)}{Z_\pi}$ where $Z_\pi = \int \gamma(h)\mathrm{d}h$ is the normalization constant. A weighted particle h, w is properly weighted with respect to γ if for any function f,

$$E[wf(h)] = Z_\pi E_{\pi(h)}[f(h)]$$

Proposition 1. If H, W input to Procedure LLM-SMC-S is properly weighted with respect to γ, then the output h', w' is properly weighted with respect to γ'. (Proof in Appendix A.1.)

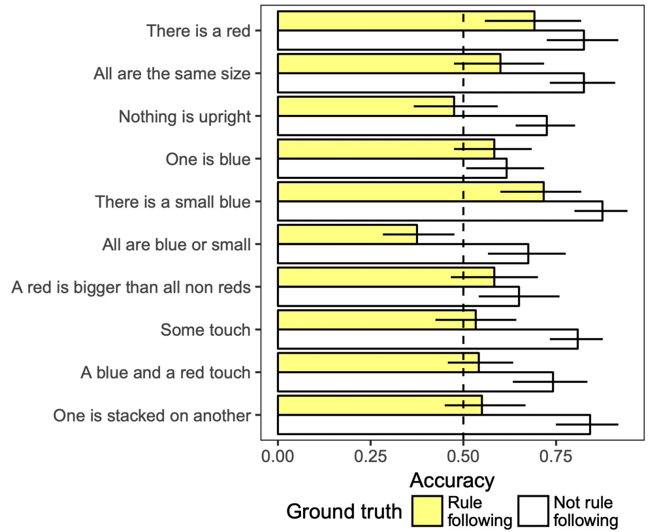
Results

Method	Zendo	ActiveACRE			
	Avg Pred Posterior	Avg Pred Posterior	ROC AUC	F1	Task Solving
Human from [13]	5.26	-	-	-	-
Direct LLM [31]	4.60 ± 0.19	0.83 ± 0.05	0.60 ± 0.02	0.86 ± 0.04	0.00 ± 0.00
Batch, Hard	6.01 ± 0.19	$\mathbf{0.89 \pm 0.03}$	0.77 ± 0.04	0.96 ± 0.01	0.10 ± 0.07
Batch w/ Refinement, Hard [9, 10]	6.18 ± 0.14	$\mathbf{0.86 \pm 0.04}$	0.73 ± 0.04	0.91 ± 0.04	0.15 ± 0.08
Online, Hard (Ours)	$\mathbf{6.55 \pm 0.13}$	$\mathbf{0.92 \pm 0.03}$	$\mathbf{0.87 \pm 0.04}$	$\mathbf{0.98 \pm 0.01}$	$\mathbf{0.35 \pm 0.11}$

Results

Method	Zendo	ActiveACRE			
	Avg Pred Posterior	Avg Pred Posterior	ROC AUC	F1	Task Solving
Human from [13]	5.26	-	-	-	-
Direct LLM [31]	4.60 ± 0.19	0.83 ± 0.05	0.60 ± 0.02	0.86 ± 0.04	0.00 ± 0.00
Batch, Hard	6.01 ± 0.19	$\mathbf{0.89 \pm 0.03}$	0.77 ± 0.04	0.96 ± 0.01	0.10 ± 0.07
Batch w/ Refinement, Hard [9, 10]	6.18 ± 0.14	$\mathbf{0.86 \pm 0.04}$	0.73 ± 0.04	0.91 ± 0.04	0.15 ± 0.08
Online, Hard (Ours)	$\mathbf{6.55 \pm 0.13}$	$\mathbf{0.92 \pm 0.03}$	$\mathbf{0.87 \pm 0.04}$	$\mathbf{0.98 \pm 0.01}$	$\mathbf{0.35 \pm 0.11}$

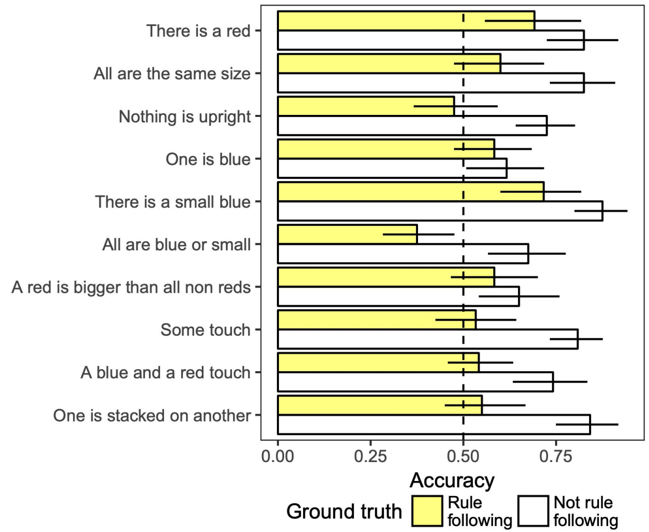
Understanding human behavior on rule induction tasks



From Bramley et al 2018

“I don’t know what’s the correct rule, but if the scene has or or does not have …, it’s definitely not going to give stars”

Understanding human behavior on rule induction tasks

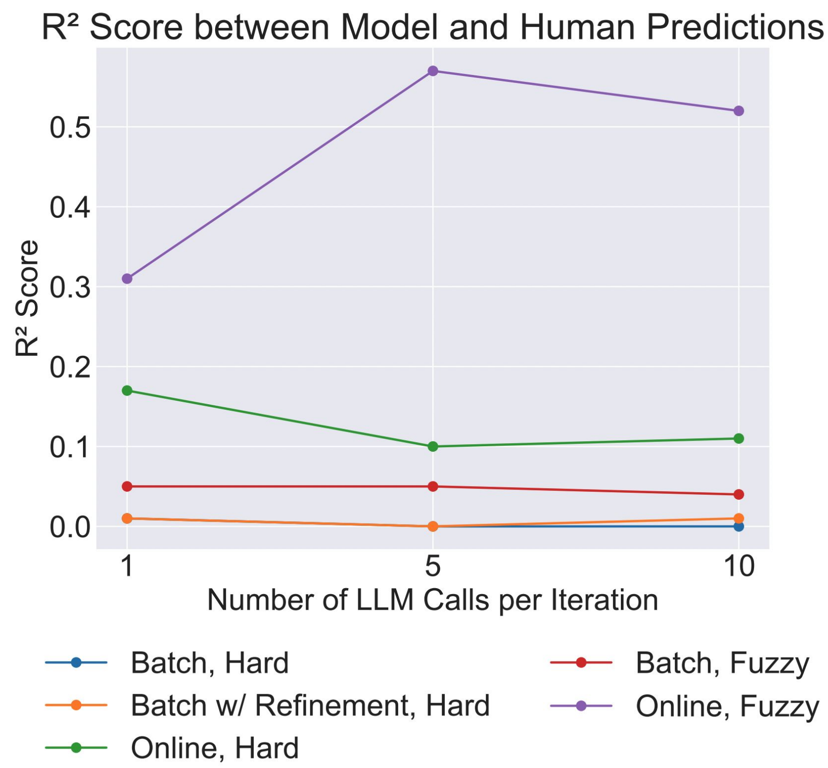


From Bramley et al 2018

“I don’t know what’s the correct rule, but if the scene has or or does not have …, it’s definitely not going to give stars”

- We can represent this with fuzzy, probabilistic rules

Being more human-like with fuzzy rules



Being more human-like with fuzzy rules

Recipe for human-like models on Zendo

- Natural language instead of formal language for hypothesis space
- Online inference instead of batch inference
- AND fuzzy rules instead of deterministic rules

Thank you!