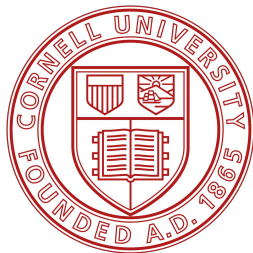


Doing Experiments and Revising Rules with Natural Language and Probabilistic Reasoning

NeurIPS 2024

Top Piriyakulkij¹, Cassidy Langenfeld¹, Tuan-Anh Le², Kevin Ellis¹

Cornell University¹, Google²



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

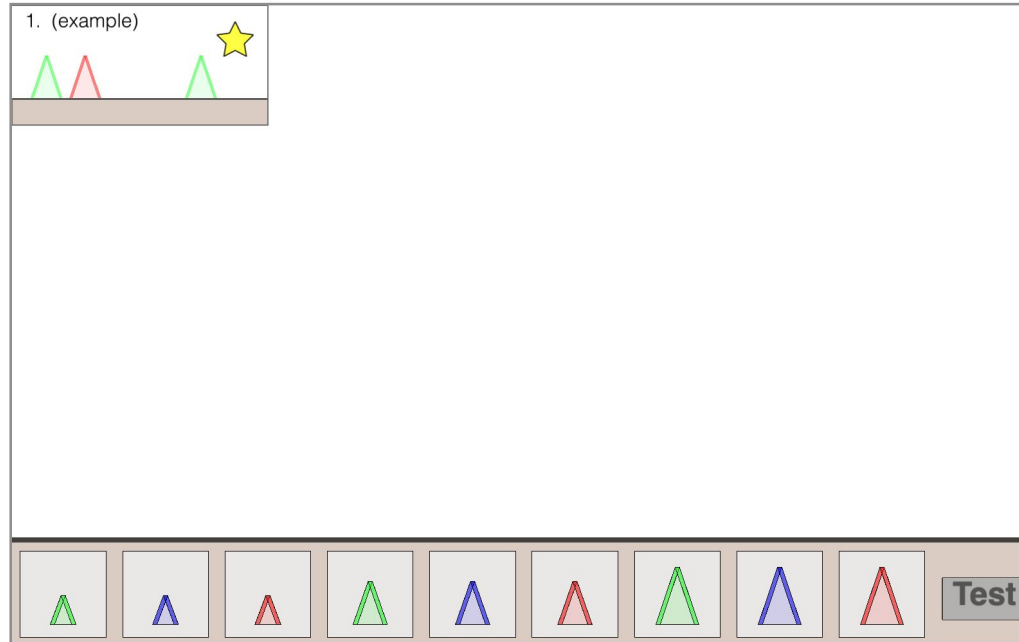
Motivation: how scientists learn

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

Domain: Zendo, an induction game

Special blocks 1 of 5

1. (example) 



The interface displays a stack of three triangles (green, red, green) on a shelf. Below the shelf is a large empty area for building. At the bottom is a toolbar with nine triangle icons (green, blue, red, green, blue, red, green, blue, red) and a 'Test' button.

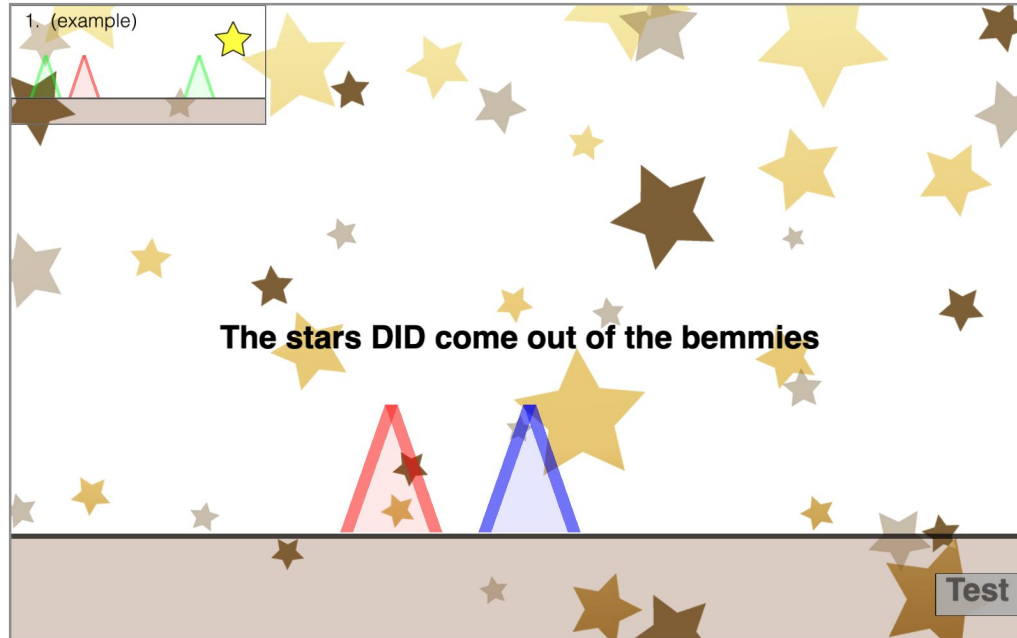
Domain: Zendo, an induction game

Special blocks 1 of 5

The image shows a screenshot of a Zendo game interface. At the top left, there is a small box labeled "1. (example)" containing three triangles (two green, one red) and a yellow star, representing a special block. The main workspace is a large white area with a light brown border. In the center of the workspace, two triangles are placed: a red one on the left and a blue one on the right. At the bottom, there is a toolbar with nine slots, each containing a triangle of a different color (blue, green, red, blue, green, red, blue, green, red). To the right of the toolbar is a grey button labeled "Test".

Domain: Zendo, an induction game

Special blocks 1 of 5



Domain: Zendo, an induction game

Special blocks 1 of 5



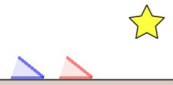
The screenshot displays the Zendo game interface. At the top, there are two panels for special blocks:

- Panel 1: (example) contains a green triangle, a red triangle, a yellow star, and a green triangle.
- Panel 2: contains a grey star, a red triangle, a blue triangle, and a yellow star.

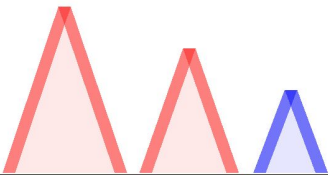
The main play area is filled with various stars (yellow, brown, grey) and two bemmies (a blue triangle and a red triangle) on a brown ground line. The text "The stars DID come out of the bemmies" is centered in the play area. A "Test" button is located in the bottom right corner.

Domain: Zendo, an induction game

Special blocks 1 of 5

| | | |
|---|---|--|
| 1. (example)  | 2.  | 3.  |
|---|---|--|

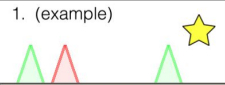


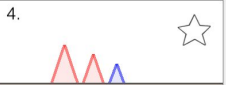












The stars DID NOT come out of the bemmies



Test

Domain: Zendo, an induction game

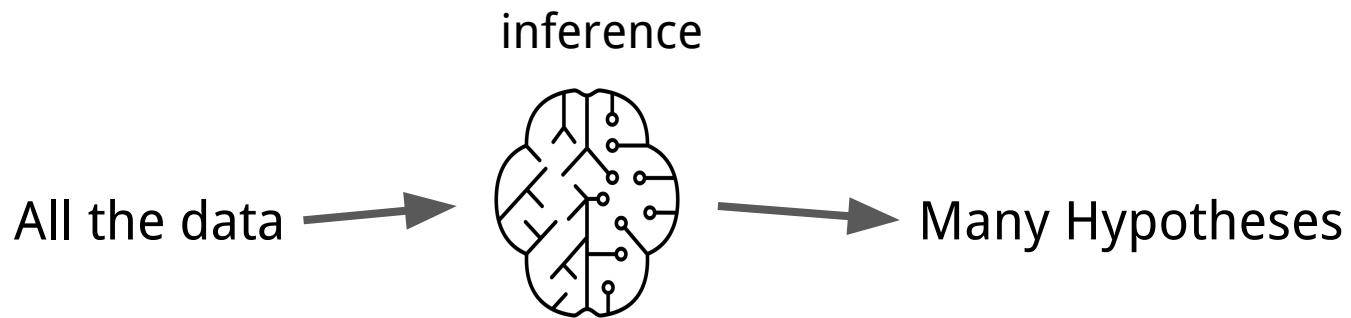
Special blocks 1 of 5

| | | | |
|--|--|---|--|
| 1. (example)  | 2.  | 3.  | 4.  |
| 5.  | 6.  | 7.  | 8.  |
| Which of the pictures show bemmies that stars will come out of? | | | |
|  |  |  |  |
|  |  |  |  |
| | | | |

Inference

How should we perform inference?

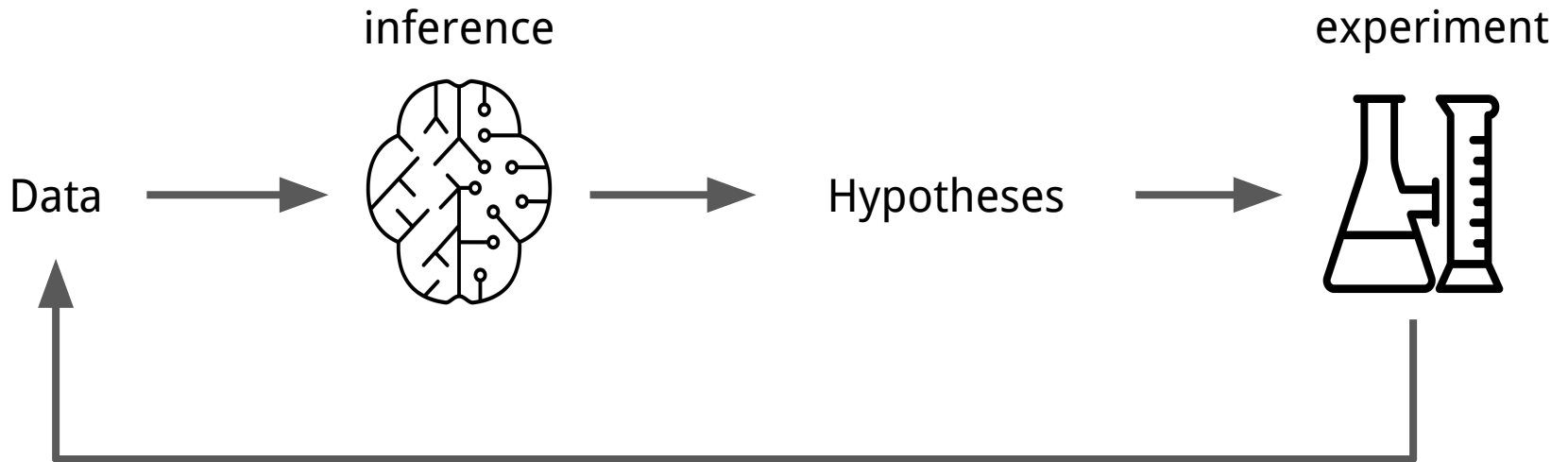
One solution: Batch Inference



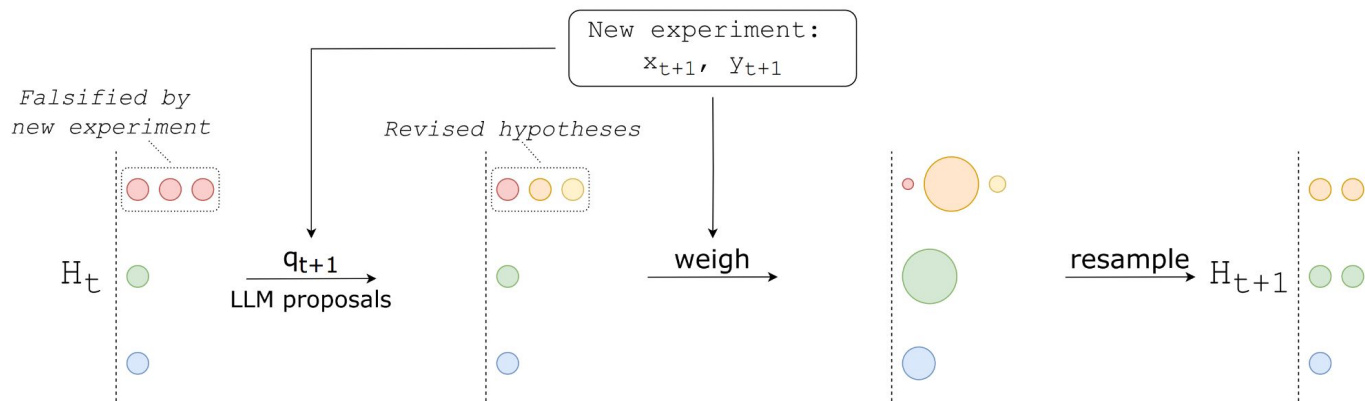
Inference

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 $p(h|x_{1:t}, y_{1:t}) \approx \sum_i w_t^{(i)} \mathbb{1}[h = h_t^{(i)}]$:

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 $Z_\pi = \int \gamma(h)dh$ 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.)

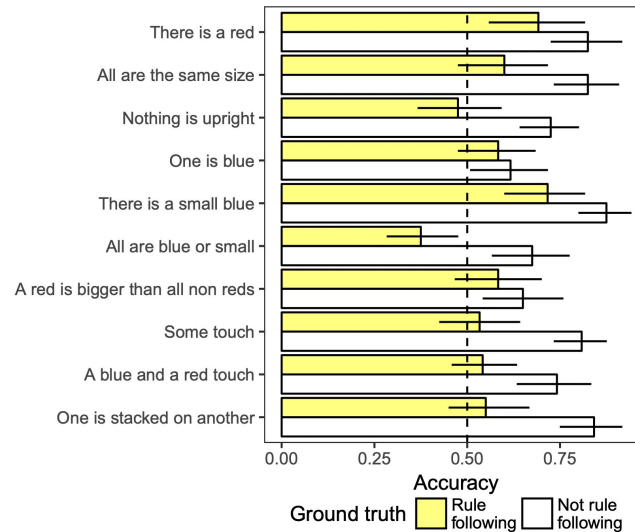
Results

| 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 | 0.89 ± 0.03 | 0.77 ± 0.04 | 0.96 ± 0.01 | 0.10 ± 0.07 |
| Batch w/ Refinement, Hard [9, 10] | 6.18 ± 0.14 | 0.86 ± 0.04 | 0.73 ± 0.04 | 0.91 ± 0.04 | 0.15 ± 0.08 |
| Online, Hard (Ours) | 6.55 ± 0.13 | 0.92 ± 0.03 | 0.87 ± 0.04 | 0.98 ± 0.01 | 0.35 ± 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 | 0.89 ± 0.03 | 0.77 ± 0.04 | 0.96 ± 0.01 | 0.10 ± 0.07 |
| Batch w/ Refinement, Hard [9, 10] | 6.18 ± 0.14 | 0.86 ± 0.04 | 0.73 ± 0.04 | 0.91 ± 0.04 | 0.15 ± 0.08 |
| Online, Hard (Ours) | 6.55 ± 0.13 | 0.92 ± 0.03 | 0.87 ± 0.04 | 0.98 ± 0.01 | 0.35 ± 0.11 |

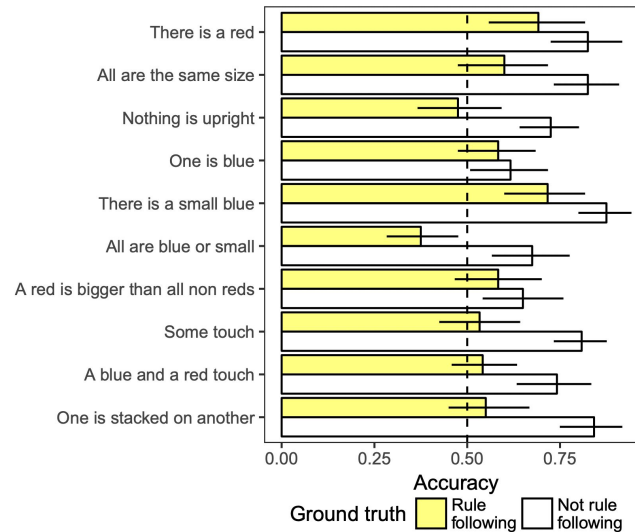
Understanding human behavior on rule induction tasks



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

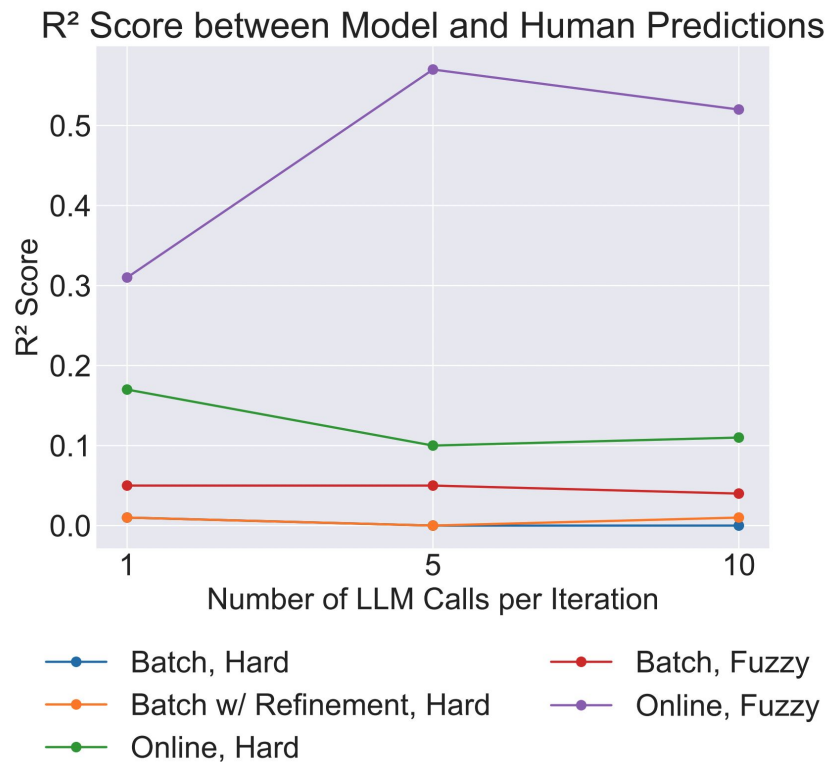


From Bramley et al 2018

“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!