WILD GUESSES AND MILD GUESSES IN ACTIVE CONCEPT LEARNING

Anirudh Chari
Massachusetts Institute of Technology
Cambridge, MA, USA
anichari@mit.edu

Neil Pattanaik
University of California, Berkeley
Berkeley, CA, USA
neil_pattanaik@berkeley.edu

ABSTRACT

Human concept learning is typically active: learners choose which instances to query or test in order to reduce uncertainty about an underlying rule or category. Active concept learning must balance informativeness of queries against the stability of the learner that generates and scores hypotheses. We study this trade-off in a neuro-symbolic Bayesian learner whose hypotheses are executable programs proposed by a large language model (LLM) and reweighted by Bayesian updating. We compare a Rational Active Learner that selects queries to maximize approximate expected information gain (EIG) and the human-like Positive Test Strategy (PTS) that queries instances predicted to be positive under the current best hypothesis. Across concept-learning tasks in the classic Number Game, EIG is effective when falsification is necessary (e.g., compound or exception-laden rules), but underperforms on simple concepts. We trace this failure to a support mismatch between the EIG policy and the LLM proposal distribution: highly diagnostic boundary queries drive the posterior toward regions where the generator produces invalid or overly specific programs, yielding a support-mismatch trap in the particle approximation. PTS is information-suboptimal but tends to maintain proposal validity by selecting "safe" queries, leading to faster convergence on simple rules. Our results suggest that "confirmation bias" may not be a cognitive error, but rather a rational adaptation for maintaining tractable inference in the sparse, open-ended hypothesis spaces characteristic of human thought.

1 INTRODUCTION

Human concept learning is typically *active*: learners select queries and interventions to reduce uncertainty about latent structure (Lake et al., 2017). We use active concept learning in reference to this query-driven setting when the goal is to identify a structured concept, such as a rule or program, rather than to optimize a predictive model over a fixed hypothesis class. In Bayesian accounts such as Bayesian Program Learning (BPL), concepts correspond to hypotheses $h \in \mathcal{H}$ (often programs), and learning is inference over $\mathcal{H}$ conditioned on observed labels (Tenenbaum, 1999; 2000; Lake et al., 2015). A growing line of neuro-symbolic work replaces hand-engineered program proposals with large language models (LLMs) that generate executable code from natural-language prompts (Ellis et al., 2023). This substitution substantially expands the effective hypothesis space, but it also introduces a new bottleneck: inference and active learning are now limited by the support and calibration of the generator used to propose hypotheses. From a cognitive-science perspective, this "proposal bottleneck" operationalizes a concrete resource constraint on hypothesis generation, making it possible to test when human-like query heuristics are adaptive because they preserve tractable inference rather than maximize information per query.

This paper studies active Bayesian concept learning in an LLM-proposed program space. A natural baseline is to choose the next query by maximizing expected information gain (EIG), a normative criterion in optimal experiment design and active learning (Oaksford & Chater, 1994; Nelson, 2005; Settles, 2009). However, human learners frequently adopt a Positive Test Strategy (PTS), preferentially querying instances expected to be positive under their current hypothesis (Klayman & Ha, 1987). PTS is often characterized as "confirmation bias" (Wason, 1960), yet rational analyses show

it can be adaptive under sparsity, cost, or asymmetric noise (Oaksford & Chater, 1994; Navarro & Perfors, 2011).

We find that in an LLM-driven neuro-symbolic learner, the EIG-optimal policy can be computationally adversarial even when it is information-theoretically sensible. On simple concepts, EIG repeatedly selects boundary cases that maximize predictive entropy under the current particle set. After observing the label, many particles are eliminated, and the learner must replenish with new hypotheses from the LLM. In precisely these high-diagnostic regimes, the proposal distribution often produces invalid, inconsistent, or overly specific programs, leading to particle degeneracy and slow recovery. We refer to this as a support-mismatch trap. In contrast, PTS tends to keep the learner within regions where the generator proposes stable, consistent hypotheses, trading off formal optimality for robust progress.

Contributions

1. We formalize an active learning loop for neuro-symbolic Bayesian concept learning with an LLM-based proposal distribution and particle posterior approximation.
2. We identify and empirically characterize a failure mode where EIG-driven querying induces proposal-support collapse, degrading performance on simple concepts.
3. We provide evidence that PTS can act as a stability-preserving heuristic in open-ended program spaces, clarifying one computational condition under which confirmation-style sampling is beneficial.

2 Problem Setup

We study binary concept learning with active queries. An unknown target concept $h^\star \in \mathcal{H}$ maps instances $x \in \mathcal{X}$ to labels $y \in \{0, 1\}$. At time t, the learner has observed a dataset

$$\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{t}, \tag{1}$$

and selects a new query $x_{t+1} \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}$, receiving $y_{t+1} = h^\star(x_{t+1})$.

In the Number Game experiments, $\mathcal{X} = \{0, \ldots, 100\}$ and $\mathcal{H}$ is an open-ended space of executable predicates (represented as Python functions) that may include arithmetic, modular, and digit-based properties. Because $\mathcal{H}$ is effectively unbounded, exact posterior inference is intractable; we approximate $p(h \mid \mathcal{D}_t)$ with a particle posterior induced by an LLM proposal distribution.

3 Neuro-Symbolic Bayesian Learner

We maintain a set of N hypothesis particles $\{h_i\}_{i=1}^N$ with weights $\{w_i\}_{i=1}^N$ approximating the posterior:

$$p(h \mid \mathcal{D}_t) \approx \sum_{i=1}^{N} w_i \, \delta(h = h_i), \qquad \sum_{i=1}^{N} w_i = 1. \tag{2}$$

Following Ellis et al. (2023), we use an LLM as a proposal distribution $q_{\mathrm{LLM}}(h \mid \mathcal{D}_t)$ by prompting it with the current observations and requesting candidate executable programs. We filter proposals for syntactic validity and consistency with $\mathcal{D}_t$; invalid or inconsistent programs are discarded.

After observing (x_{t+1}, y_{t+1}), we update each weight by checking consistency with the new label. When particle diversity collapses (effective sample size below a threshold), we rejuvenate by drawing additional proposals from $q_{\mathrm{LLM}}(h \mid \mathcal{D}_{t+1})$ and reweighting. In practice, rejuvenation quality depends strongly on whether the new dataset $\mathcal{D}_{t+1}$ lies within the proposal distribution's reliable support.

For each particle h_i, we define an unnormalized weight

$$\tilde{w}_i \propto p(\mathcal{D}_t \mid h_i)\, p(h_i), \tag{3}$$

and normalize across particles.

Likelihood (size principle) We use the size principle for positive examples:

$$p(\mathcal{D}_t \mid h) \propto \left(\frac{1}{|h|}\right)^{|\mathcal{D}_t^+|}, \tag{4}$$

where $|h|$ denotes the extension size (the number of $x \in \mathcal{X}$ such that $h(x) = 1$) and $\mathcal{D}_t^+$ is the set of observed positive examples. Intuitively, this penalizes overly broad concepts.

Prior (simplicity bias) We include a code-simplicity prior based on model scoring of the program text (e.g., average token log-probability under a code model), which biases toward shorter and more likely programs:

$$p(h) \propto \exp\left(\frac{1}{|h|_{\text{tok}}} \sum_{j=1}^{|h|_{\text{tok}}} \log p_{\text{Code}}(t_j \mid t_{<j})\right). \tag{5}$$

This prior is intended as a computational proxy for description-length biases used in program induction.

4 Active Query Policies

At each step, a policy selects the next query $x \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}$. We use a passive policy baseline, which samples x uniformly at random from unqueried instances.

4.1 Rational Active Learner: approximate EIG

Given the particle posterior, the predictive probability of label y at instance x is

$$p(y \mid x, \mathcal{D}_t) \approx \sum_{i=1}^{N} w_i \, \mathbb{I}[h_i(x) = y]. \tag{6}$$

We approximate the expected information gain of querying x as the expected reduction in posterior entropy:

$$\text{EIG}(x; \mathcal{D}_t) = H(p(h \mid \mathcal{D}_t)) - \sum_{y \in \{0,1\}} p(y \mid x, \mathcal{D}_t) \, H(p(h \mid \mathcal{D}_t \cup \{(x, y)\})), \tag{7}$$

where $H(\cdot)$ is Shannon entropy computed over the discrete particle weights. The EIG policy selects

$$x^\star_{\text{EIG}} = \arg \max_{x \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}} \text{EIG}(x; \mathcal{D}_t). \tag{8}$$

This criterion is normative with respect to the current posterior approximation; it does not account for failures of the hypothesis generator after updating.

4.2 Positive Test Strategy (PTS)

PTS queries instances predicted to be positive under the current MAP hypothesis:

$$h_{\text{MAP}} = \arg \max_{h_i} w_i, \qquad x^\star_{\text{PTS}} \sim \text{Unif}\Big(\{x \in \mathcal{X} \setminus \{x_1, \ldots, x_t\} : h_{\text{MAP}}(x) = 1\}\Big). \tag{9}$$

Unlike EIG, PTS is not designed to maximize disambiguation. Its computational advantage is that it tends to avoid queries that sharply constrain the hypothesis space in ways that exceed the generator's ability to propose alternatives.

5 Results

We evaluate on the Number Game with $\mathcal{X} = \{0, \ldots, 100\}$. Target concepts are grouped into Easy (single salient property), Medium (boolean combinations or constraints), and Hard (non-local or idiosyncratic structure). At each iteration we prompt an LLM to propose candidate Python predicates

Algorithm 1 Active neuro-symbolic Bayesian concept learning (particle approximation).

Require: Instance space $\mathcal{X}$, budget T, particles N, policy $\pi \in \{\mathrm{EIG}, \mathrm{PTS}, \mathrm{rand}\}$
Initialize $\mathcal{D}_0 \leftarrow \emptyset$
Propose initial particles $\{h_i\}_{i=1}^N \sim q_{\mathrm{LLM}}(h \mid \mathcal{D}_0)$; set weights $w_i \propto p(\mathcal{D}_0 \mid h_i)p(h_i)$
for $t = 0, 1, \ldots, T-1$ **do**
    Select query $x_{t+1} \leftarrow \pi(\mathcal{X}, \mathcal{D}_t, \{h_i, w_i\}_{i=1}^N)$
    Observe label $y_{t+1} \leftarrow h^\star(x_{t+1})$
    $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_{t+1}, y_{t+1})\}$
    Update weights: $w_i \leftarrow w_i \cdot \mathbb{I}[h_i(x_{t+1}) = y_{t+1}]$; normalize
    if particle degeneracy (low ESS) **then**
        Propose additional hypotheses from $q_{\mathrm{LLM}}(h \mid \mathcal{D}_{t+1})$, filter for consistency
        Reweight all particles by $p(\mathcal{D}_{t+1} \mid h)p(h)$; normalize
    end if
end for

Table 1: Number of queries to convergence (≤ 50) in the Number Game. **DNF** indicates no convergence within the budget.

Rule	EIG	PTS	Passive
Easy rules			
Square numbers	3	2	2
Multiples of 4	7	4	2
Odd numbers	8	6	5
Powers of 2	5	2	2
Medium rules			
Even numbers < 30	17	31	17
Multiples of 3 or 7	8	**DNF**	13
Odd multiples of 3	14	35	21
Numbers ending in 6	4	7	7
Hard rules			
Digits sum to < 8	43	**DNF**	**DNF**
Remainder $= 5 \pmod 9$	31	**DNF**	41
One less than a prime	**DNF**	**DNF**	**DNF**
Twice a square minus 2	**DNF**	**DNF**	**DNF**

consistent with the current dataset $\mathcal{D}_t$. We use Gemini 2.5 Flash as the proposal engine. Proposals are filtered for syntactic validity and consistency with all observed labels.

Each trial runs for at most $T = 50$ queries. We declare convergence when the particle posterior assigns confidence at least 0.95 to the MAP hypothesis (within the maintained particle set). Runs that do not converge within the budget are marked **DNF** (did not finish). For each target concept and strategy, we report the number of queries to convergence (lower is better). Because the posterior is approximate, this metric captures both informativeness of queries and robustness of the proposal-and-filtering pipeline.

Table 1 summarizes query counts across strategies.

Overthinking to Underperform On Easy rules, EIG is not reliably better than passive sampling and is often worse than PTS. Qualitatively, EIG tends to select boundary points that maximize predictive entropy under the current particle set. After observing the label, the posterior can collapse onto a narrow region where few valid hypotheses remain under the LLM proposal distribution. The subsequent rejuvenation step then produces many invalid or overly specific programs, delaying recovery. This is the **support-mismatch trap**: the policy is "optimal" for the current approximation but induces datasets for which the generator has poor coverage.

Falsification is Useful For Medium rules, the pattern reverses: PTS frequently fails to seek counterevidence needed to reject plausible subsets (e.g., committing to a stricter divisor rule when the target is looser). EIG more consistently identifies disambiguating queries, reducing time to convergence on several compound concepts.

Generator Bottleneck All methods struggle on Hard rules, with EIG sometimes succeeding where others do not. However, the overall failure rate suggests that query selection cannot compensate for a proposal distribution that assigns negligible mass to the target concept. Active learning can amplify existing modeling capacity, but it cannot reliably recover hypotheses that the generator almost never proposes.

6 Analysis and Discussion

6.1 A computational account of confirmation-style sampling

PTS improves performance on simple concepts not by maximizing information per query, but by maintaining a dataset that stays within the generator's reliable regime. In an LLM-driven learner, "rationality" must be evaluated with respect to the full system: inference and hypothesis proposal. Under proposal-support limitations, PTS can be viewed as a stability-preserving heuristic that avoids high-risk queries that cause particle collapse and low-quality regeneration.

6.2 When should an agent switch strategies?

The results are consistent with a meta-policy hypothesis: use low-cost, stability-preserving sampling early (PTS-like) and shift toward falsification-heavy sampling (EIG-like) when progress stalls, contradictions appear, or concepts appear compositional. Testing such a switch requires explicit criteria (e.g., posterior stagnation, effective sample size, or repeated generator failure) and is a concrete direction for workshop-relevant human-inspired AI reasoning.

6.3 Implications for neurosymbolic systems

The support-mismatch trap suggests that active learning objectives should incorporate generator awareness. Two practical directions are: generator-regularized acquisition functions that penalize queries expected to induce low proposal success, and robust rejuvenation mechanisms (e.g., structured DSL backstops or constrained decoding) that maintain coverage under sharper constraints. We do not explore these here; the present goal is to isolate and document the failure mode.

From a cognitive perspective, the same analysis offers a functional interpretation of PTS. If human hypothesis generation is similarly sparse and support-limited, then PTS can be viewed as a heuristic for keeping inference in a "high-coverage" region of hypothesis space. In this view, preferentially sampling predicted positives is not primarily about seeking confirmatory evidence per se, but about avoiding queries that would force abrupt shifts into regions where few coherent hypotheses can be generated or maintained. This provides a concrete mechanism by which PTS can enable rapid convergence on simple concepts: it preserves stable hypothesis proposal and prevents the kind of regeneration failures that arise after highly diagnostic boundary queries.

7 Conclusion

We studied active concept learning in a neuro-symbolic Bayesian learner whose hypotheses are programs proposed by an LLM. Although expected information gain is a principled acquisition objective, we find that it can be counterproductive in open-ended program spaces because it induces datasets that fall outside the generator's reliable support, causing particle degeneracy and slow recovery. A Positive Test Strategy, while not information-optimal, can act as a stability-preserving heuristic that accelerates learning on simple concepts. These findings motivate generator-aware active learning objectives and meta-policies that adaptively trade off falsification against stability—a direct bridge between cognitive regularities in human inquiry and practical design constraints in modern neurosymbolic AI.

REFERENCES

Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sablé-Meyer, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B. Tenenbaum. Human-like few-shot learning via bayesian reasoning over natural language. *arXiv preprint arXiv:2306.02797*, 2023.

Joshua Klayman and Young-Won Ha. Confirmation, disconfirmation, and information in hypothesis testing. *Psychological Review*, 94(2):211, 1987.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017.

Daniel J. Navarro and Amy F. Perfors. Hypothesis generation, sparse categories, and the positive test strategy. *Psychological Review*, 118(1):120, 2011.

Jonathan D. Nelson. Finding useful questions: on bayesian diagnosticity, probability, probability gain, and information gain. *Psychological Review*, 112(4):979, 2005.

Mike Oaksford and Nick Chater. A rational analysis of the selection task as optimal data selection. *Psychological Review*, 101(4):608, 1994.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Joshua B. Tenenbaum. *A Bayesian framework for concept learning*. PhD thesis, Massachusetts Institute of Technology, 1999.

Joshua B. Tenenbaum. Rules and similarities in concept learning. *Advances in Neural Information Processing Systems*, 12, 2000.

Peter C. Wason. On the failure to eliminate hypotheses in a conceptual task. *Quarterly journal of experimental psychology*, 12(3):129–140, 1960.

A Additional Results and Trajectories

	Rational Active Learner (EIG)		Positive Test Strategy (PTS)	
Iter	**Top hypothesis** (P_{conf})	**Query**	**Top hypothesis** (P_{conf})	**Query**
(a) Target concept: Multiples of 4				
1	Divisible by 4 (0.83)	**64** (Yes)	Power of 2 (0.72)	**1** (No)
2	Power of 2 (0.94)	**32** (Yes)	Divisible by 4 (0.99)	**36** (Yes)
3	Power of 2 (1.00)	**22** (No)	Divisible by 4 (0.99)	**24** (Yes)
4	Power of 2 (1.00)	**77** (No)	Divisible by 4 (1.00)	– *Converged* –
5	Power of 2 (1.00)	**36** (Yes)		
6	Divisible by 4 (0.97)	**81** (No)		
7	Divisible by 4 (1.00)	– *Converged* –		
(b) Target concept: Powers of 2				
1	Power of 2 (0.59)	**0** (No)	Power of 2 (0.59)	**64** (Yes)
2	Power of 2 (0.72)	**9** (No)	Power of 2 (1.00)	– *Converged* –
3	Power of 2 (0.99)	**1** (Yes)		
4	Power of 2 (0.96)	**3** (No)		

Table 2: Example learning trajectories under EIG and PTS.

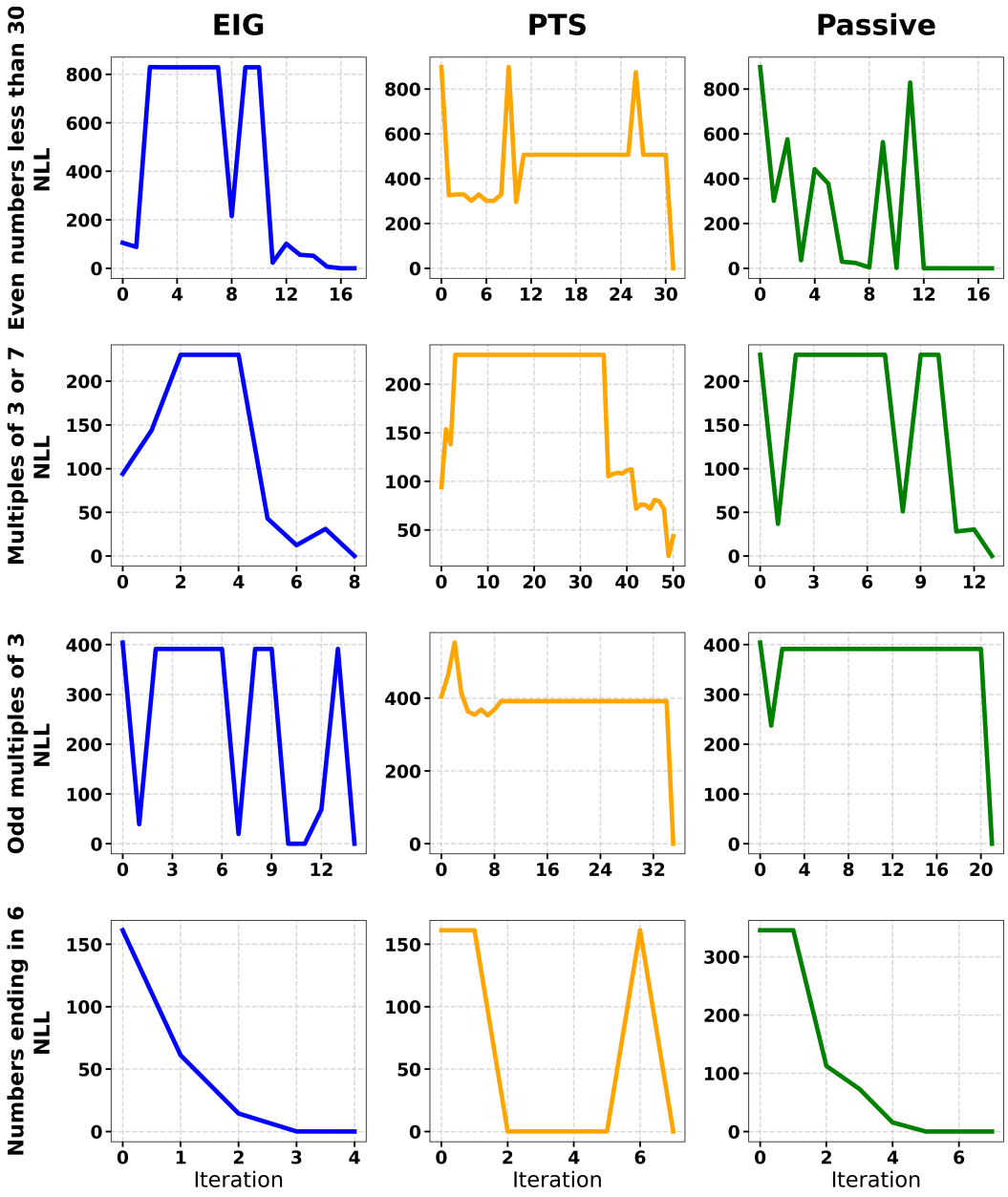

Figure 1: Negative log-likelihood (NLL) of the true concept over time for representative Medium-rule trials (lower is better).