

# Learning Machine Behavior

Measuring Exploration, Stability, and Alignment in Learned Agents

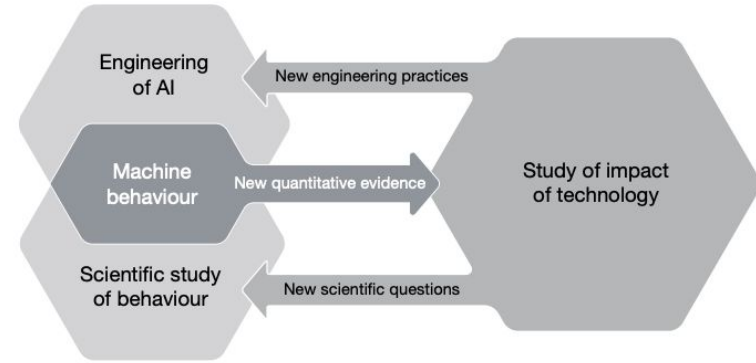
Liv G. d'Aliberti (They / Them)  
Princeton C.S. General Exam

# Thesis

Modern AI systems are not just static predictors. They are **learned behavioral systems**.

So, I argue that evaluation should similarly not be static, and moreover, should ask:

1. What behavior(s) is/are present?
2. Is it stable?
3. Is it causally meaningful?
4. Moreover, can we design learning algorithms to shape it?



**Fig. 2 | The interdisciplinarity of machine behaviour.** Machine behaviour lies at the intersection of the fields that design and engineer AI systems and the fields that traditionally use scientific methods to study the behaviour of biological agents. The insights from machine behavioural studies provide quantitative evidence that can help to inform those fields that study the potential effects of technology on social and technological systems. In turn, those fields can provide useful engineering practices and scientific questions to fields that examine machine behaviours. Finally, the scientific study of behaviour helps AI scholars to make more precise statements about what AI systems can and cannot do.

## Position: Evaluating Generative AI Systems Is a Social Science Measurement Challenge

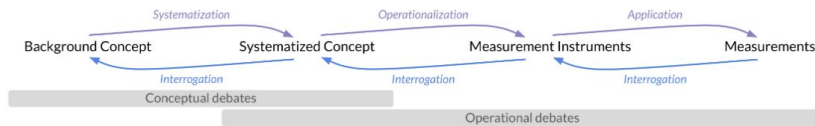


Figure 1. A variant of the framework of Adcock & Collier (2001). The background concept, the systematized concept, the measurement instruments, and the measurements are linked by four processes: systematization, operationalization, application, and interrogation.

In this work, we take the research approach of Wallach et al. 2025.

We follow the position outlined by Rahwan et al. 2019, that machine behavior is a interdisciplinary field of scientific study worthy of study in order to ensure machine behavior is aligned with the goals of society.

# Why “machine behavior”?

AI systems now reason, sample, revise, act, and mediate institutions.

Benchmarking, while a great way to capture average model performance, fails to sufficiently evaluate the diversity of model behavior. And, we need behavioral evaluation for:

- **Exploration:** what alternatives does the system consider?
- **Stability:** does behavior persist across prompts, samples, and seeds?
- **Alignment:** do observed behaviors match human and institutional needs?

Sutton & Barto 2018

Type of question	Object of study	
	Dynamic view Explanation of current form in terms of a historical sequence	Static view Explanation of the current behaviour of a machine
Proximate view How a particular type of machine functions	<b>Development (ontogeny)</b> Developmental explanations of how a type of machine acquires its behaviour, from deliberate engineering and supervised learning based on specific benchmarks, to online learning and reinforcement learning in a particular environment.	<b>Mechanism (causation)</b> Mechanistic explanations for what the behaviour is, and how it is constructed, including computational mechanisms or external stimuli that trigger it.
Ultimate (evolutionary) view Why a type of machine evolved the behaviours it has	<b>Evolution (phylogeny)</b> Incentives and market forces that describe why the behaviour evolved and spread, whether by programming or learning, subject to computational and institutional constraints.	<b>Function (adaptive value)</b> The consequences of the machine's behaviour in the current environment that cause it to persist, either by appeal for particular stakeholders (such as users or companies) or fit to some other aspect of the environment.

Rahwan et al. 2019.

# Machine Behavior → Cognitive science connection

Human insight is often framed as restructuring a problem representation.

But for models, we need to distinguish:

- behavioral resemblance,
- functional equivalence,
- internal mechanism,
- causal contribution to performance.

## **Abstract**

Accounts of how people learn functional relationships between continuous variables have tended to focus on two possibilities: that people are estimating explicit functions, or that they are performing associative learning supported by similarity. We provide a rational analysis of function learning, drawing on work on regression in machine learning and statistics. Using the equivalence of Bayesian linear regression and Gaussian processes, we show that learning explicit rules and using similarity can be seen as two views of one solution to this problem. We use this insight to define a Gaussian process model of human function learning that combines the strengths of both approaches.

Griffiths et al. 2009.

## **Discussion**

In 1904, sixteen leading academics of the Prussian Academy of Sciences signed a statement indicating that a horse, named “Clever Hans,” could solve mathematical problems at a human-like level. Back then, it took another scientist, Oskar Pfungst, years of systematic investigations to prove that the horse was merely reacting to the people who were watching him (36). With the advent of large-scale machine learning models, the risks of overinterpreting simple behaviors as intelligent runs rampant.

Binz & Schulz 2023

# Overview: A Tale of Two Papers

(This analogy makes sense because we are going to look at two very different ways to study machine behavior that are two reflections of the same task.)

## The Illusion of Insight in Reasoning Models

Liv G. d’Aliberti

Princeton University

Department of Computer Science  
Princeton, NJ, USA

liv.daliberti@princeton.edu

Manoel Horta Ribeiro

Princeton University

Department of Computer Science  
Princeton, NJ, USA

manoel@cs.princeton.edu

## Abstract

Do reasoning models have “Aha!” moments? Prior work suggests that models like DeepSeek-R1-Zero undergo sudden mid-trace realizations that lead to accurate outputs, implying an *intrinsic* capacity for self-correction. Yet, it remains unclear whether such intrinsic shifts in reasoning strategy actually improve performance. Here, we study mid-reasoning shifts and instrument training runs to detect them. Our analysis spans 1M+ reasoning traces, hundreds of training checkpoints, three reasoning domains, and multiple decoding temperatures and model architectures. We find that reasoning shifts are rare, do not become more frequent with training, and seldom improve accuracy, indicating that they do not correspond to prior perceptions of model insight. However, their effect varies with model uncertainty. Building on this finding, we show that artificially triggering *extrinsic* shifts under high entropy reliably improves accuracy. Our results show that mid-reasoning shifts are symptoms of unstable inference behavior rather than an intrinsic mechanism for self-correction.

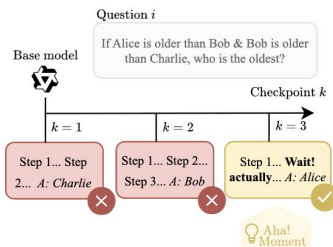


Figure 1: **Anatomy of an “Aha!” Moment.** We illustrate an “Aha!” moment as described in Guo et al. (2025): within a single chain-of-thought, a cue such as “Wait... let’s re-evaluate” marks a shift from an initially failing strategy ( $k \in \{1, 2\}$ ) to one that yields a correct answer (when  $k = 3$ ). The figure also anticipates our methodology: we study “Aha!” moments by systematically GRPO-tuning and annotating the reasoning traces of Qwen2.5 and Llama models.

reward queries, prompting techniques, or external tools (Lightman et al., 2024; Li et al., 2024a; Zhang

## Behavior-Consistent Deep Reinforcement Learning

Marcel Hussing\*  
University of Pennsylvania  
mhussing@seas.upenn.edu

Liv d’Aliberti\*  
Princeton University  
od2961@princeton.edu

Claas Voelcker  
University of Texas at Austin

Benjamin Eysenbach  
Princeton University

Eric Eaton  
University of Pennsylvania

## Abstract

Reinforcement learning (RL) often exhibits high variance across training runs, leading to unreliable performance and posing a major challenge to deployment in real-world domains. In this work, we address the challenge of cross-run policy divergence by formalizing the problem of *behavior-consistent RL*, where the objective is to obtain policies that are both high-performing and distributionally similar across training runs. Our key observation is that maximum-entropy RL provides a direct mechanism for controlling behavioral divergence by anchoring runs to a common (uniform) prior. We prove that, for Boltzmann policies, choosing the temperature proportional to  $Q$ -function disagreement bounds the pairwise KL divergence between the induced policies. However, we also show that naively increasing entropy might impair policy optimization while amplifying off-policy error. Building upon these observations, we propose  $Q$ -value Expectile Disagreement (QED), a state-dependent temperature schedule that uses double-critic disagreement as a single-run proxy for cross-run disagreement. Empirically, we demonstrate that across 18 continuous-control tasks, QED reduces cross-run divergence by two orders of magnitude without sacrificing performance, resulting in a considerable reduction in return variance at modest sample-efficiency costs.

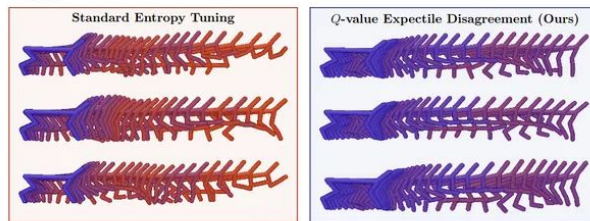


Figure 1: QED makes independently trained policies visibly behavior-consistent. Visualization of policies from three different training runs on the cheetah\_run task, comparing (left) traditional entropy autotuning (Haamoja et al., 2018b) against (right) our approach (QED). Color shade denotes the mean pairwise  $L_2$  distance between state vectors at each timestep: blue is low, red is high.

# Aha Paper: One “So What” – Compelling Stories vs Quantifiable Narratives

Qualitative story-telling about/ hand-selected examples of model behavior is narratively seductive.

---

Question: If  $a > 1$ , then the sum of the real solutions of  $\sqrt{a - \sqrt{a + x}} = x$  is equal to

---

Response: <think>

To solve the equation  $\sqrt{a - \sqrt{a + x}} = x$ , let's start by squaring both ...

$$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

...

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be ...

We started with the equation:

$$\sqrt{a - \sqrt{a + x}} = x$$

First, let's square both sides:

$$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$$

Next, I could square both sides again, treating the equation: ...

...

---

But, using one-off exemplars to define the “emergent” capabilities of models is neither scientific nor consistent with actual model behavior.\*

## Aha Moment of DeepSeek-R1-Zero

A particularly intriguing phenomenon observed during the training of DeepSeek-R1-Zero is the occurrence of an “aha moment”. This moment, as illustrated in Table 3, occurs in an intermediate version of the model. During this phase, DeepSeek-R1-Zero learns to allocate more thinking time to a problem by reevaluating its initial approach. This behavior is not only a testament to the model's growing reasoning abilities but also a captivating example of how reinforcement learning can lead to unexpected and sophisticated outcomes.

Table 3: An interesting “aha moment” of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.

\*I recognize the inherent irony of my qualitatively exemplifying the problem of “qualitatively exemplifying” in machine behavior. Let's not get too “inception”-y / meta about it?

# Behavioral Claims as Measurement Claims

**A claim like “the model had an Aha moment” is not just descriptive.**

It implies a causal and measurement structure.

For each behavioral claim, we need:

- **Construct:** what behavior are we claiming exists?
- **Operationalization:** what observable pattern counts?
- **Comparison:** compared to what baseline?
- **Alternative explanations:** prompting, sampling, optimization, difficulty, or interpretation?
- **Estimand:** what effect are we actually trying to measure?

Wallach et al. 2025; Huntington-Klein 2021; Messeri & Crockett 2024.

# Case study: “Aha!” Moments (d’Aliberti & Ribeiro; 2026)

Central question: **Do reasoning models exhibit intrinsic “Aha!” moments?**

Following Wallach et al. 2025, we aim to carry out rigorous evaluation of the behavior through systemization, operationalization, application, and interrogation.

Background Conceptualization

Definition / Systemization

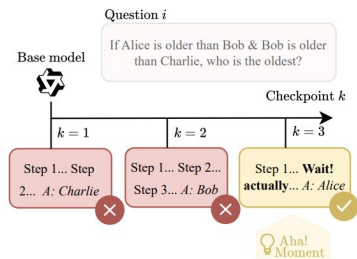


Figure 1: **Anatomy of an “Aha!” Moment.** We illustrate an “Aha!” moment as described in Guo et al. (2025): within a single chain-of-thought, a cue such as “Wait... let’s re-evaluate” marks a shift from an initially failing strategy ( $k \in \{1, 2\}$ ) to one that yields a correct answer (when  $k = 3$ ). The figure also anticipates our methodology: we study “Aha!” moments by systematically GRPO-tuning and annotating the reasoning traces of Qwen2.5 and Llama models.

Idea comes from psychology / study of “Eureka Moments” - Jones 2003, Kohler 1921, which we formalize for study in LLMs as:

1. At all earlier model checkpoints, the model failed on the problem.
2. Earlier checkpoints of the model did not show evidence of mid-trace shifts.
3. The current checkpoint does show:
  - a. A detectable shift
  - b. Correct output

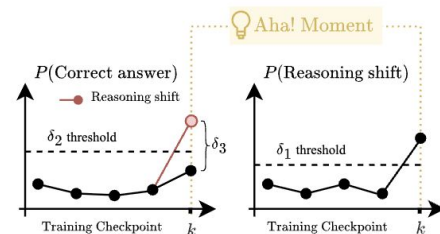


Figure 2: **Schematic of our operational “Aha!” definition.** For a fixed problem  $q_j$  (horizontal axis: checkpoint index  $i$ ), the figure visualizes the three criteria in Def. 3.1. (1) *Prior failures*: empirical correctness  $\hat{P}_{\theta_i}(\checkmark | q_j)$  remains below  $\delta_1$  at all checkpoints  $i < k$ . (2) *Prior stability*: the shift rate  $\hat{\pi}_i = \Pr[S_{q_j, i} = 1]$  stays below  $\delta_2$  for all  $i < k$ . (3) *Performance gain*: at checkpoint  $k$ , correctness on traces *with* a detected shift (red) exceeds correctness over *all* traces (black) by more than  $\delta_3$ .

## So What? – Why Does This Behavior Matter?

If a model can reliably reconsider mid-inference, that matters for alignment: it changes how we think about monitoring, self-correction, strategic behavior, and whether traces are faithful evidence of internal computation.

Additionally, if in training a model is able to reconsider its position, then work using RL Alignment would be intractable / made much more difficult. The model could be learning to “fake” its position.

Concern: Traces can create the illusion of understanding (Messeri & Crockett; 2024)

# Others Work in the Space – Liu et al.; 2025 (Base Model Analysis )

## There May Not be Aha Moment in R1-Zero-like Training — A Pilot Study

Zichen Liu\*, Changyu Chen\*, Wenjun Li\*, Tianyu Pang, Chao Du, Min Lin

\* Equal contributions.

07 Feb, 2025

Codes: <https://github.com/sail-sg/oat-zero>

One of the most inspiring results from DeepSeek-R1-Zero is the occurrence of “Aha moment” through pure reinforcement learning (RL). At the Aha moment, the model learns emergent skills such as self-reflection, which helps it to conduct in-context search to solve complex reasoning problems.

Within only a few days after R1-Zero’s release, several projects independently “reproduced” R1-Zero-like training on smaller scales (e.g., 1B to 7B) and all observed the Aha moment, which is typically typically accompanied by an increase in response length. We follow their settings to scrutinize the R1-Zero-like training process, and share the following findings in this blog:



1. **There may NOT be Aha moment in R1-Zero-like training.** Instead, we found Aha moment (such as self-reflection patterns) appears at epoch 0, namely base models.
2. We found **Superficial Self-Reflection (SSR)** from base models’ responses, in which case self-reflections do not necessarily lead to correct final answers.
3. We took a **closer look at R1-Zero-like training via RL**, and found that the increasing response length phenomenon is not due to the emergence of self-reflection, but a consequence of RL optimizing well-designed rule-based reward functions.

## 2.2 Base models are prone to SSRs

We next analyze the occurrences of self-reflection keywords in both correct and incorrect responses of Qwen2.5-Math-1.5B. As shown in Fig. 4, the majority of self-reflections (measured by their frequency) do not lead to correct answers across different sampling temperatures, suggesting that base models are prone to producing superficial self-reflections.

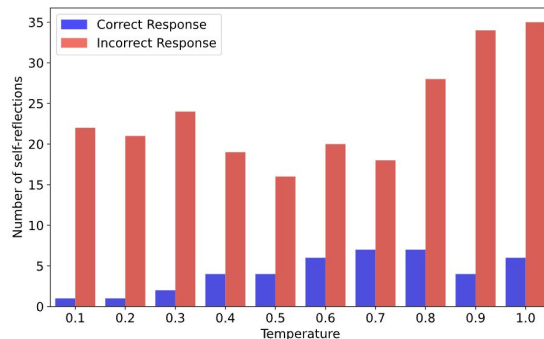


Fig 4. Number of self-reflections in correct and incorrect responses. The blue bars represent the total occurrences of self-reflection keywords in correct responses, while the red bars represent the total occurrences in incorrect responses.

# Others Work in the Space – Illusion of Thinking (Apple Paper); Trained Model Analysis

Despite these claims and performance advancements, the fundamental benefits and limitations of LLMs remain insufficiently understood. Critical questions still persist: Are these models capable of generalizable reasoning, or are they leveraging different forms of pattern matching ([gmsymbolic](#))? How does their performance scale with increasing problem complexity? How do they compare to their standard LLM (non-reasoning) counterparts when provided with the same inference token compute? Most importantly, what are the inherent limitations of current reasoning approaches, and what improvements might be necessary to advance toward more robust reasoning capabilities?

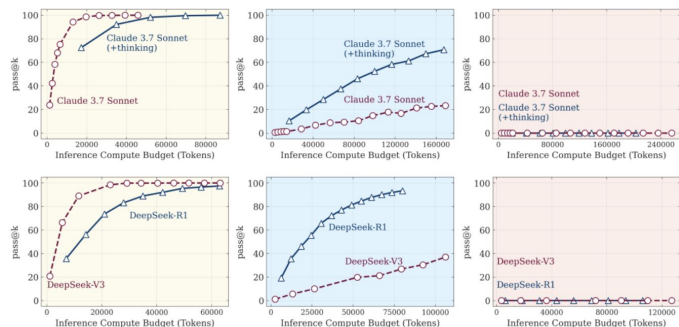


Figure 4: Pass@k performance of thinking models (Claude 3.7 Sonnet with extended thinking, DeepSeek-R1) versus their non-thinking counterparts (Claude 3.7 Sonnet, DeepSeek-V3) across equivalent inference compute budgets in puzzle environments of low, medium, and high complexity. Non-thinking models outperform in simple problems, thinking models show advantages at medium complexity, while both approaches fail at high complexity regardless of compute allocation.

## 4.2 How Does Complexity Affect Reasoning?

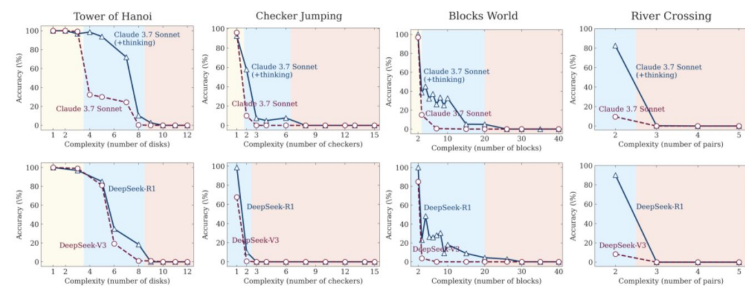


Figure 5: Accuracy of thinking models (Claude 3.7 Sonnet with extended thinking, DeepSeek-R1) versus their non-thinking counterparts (Claude 3.7 Sonnet, DeepSeek-V3) across all puzzle environments and varying levels of problem complexity.

# Our Direction

Perhaps Reasoning / “Aha” is not achievable in inference, but...

Maybe “Aha” Moments take place during the finetuning / learning process? This would make sense, as humans exhibit moments of insight during learning rather than either before or after the learning experience.

Why this is hard:

1. No access to intermediate fine-tuned weights for DeepSeek-R1/ OpenAI Models.
2. Definitely not enough compute to re-finetune.
3. How do we separate the LLM from the deployed product / operational system?

# Our Method of Study

1. Look at Intermediate Performance fine-tuning smaller LLMs. Smaller open models give us an instrumented testbed: we can inspect intermediate checkpoints, vary temperature, and test whether the claimed behavior appears under controlled training conditions.
2. Evaluate final performance on the same validation sets on large models.

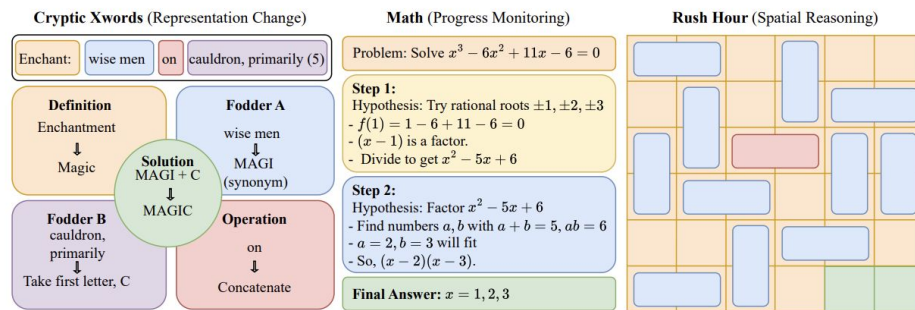


Figure 3: **Three reasoning lenses and example instances.** Each row illustrates one evaluation domain and how it instantiates the three “reasoning lenses” introduced in §4. *Left (representation change)*: a cryptic Xwords clue with definition and wordplay; shifts correspond to re-parsing the clue (e.g., switching from anagram to charade or hidden-word). *Center (progress monitoring)*: a math problem with explicit chain-of-thought and checks; shifts occur when the model abandons an inconsistent derivation and restarts with a new method. These domains form complementary testbeds for studying when mid-trace shifts (our “Aha!” events; Def. 3.1) co-occur with changes in uncertainty and accuracy. *Right (spatial manipulation)*: a RHour puzzle requiring a planned sequence of legal moves; mid-trace shifts reflect abandoning one move plan for another.

# Need Intermediate Checkpoints

We finetune 2 architectures:

1. Qwen Family (base for DeepSeek-R1)
2. Llama Family (U.S. trained base model by Facebook)

At 2 scale sizes:

1. 1.5B Base
2. 7/8B Base

Take Checkpoints every 50 optimizer steps

Model	Domain	Step 0	After	Step	$\Delta$
Qwen2.5-1.5B	Xwords	7.69	10.00	950	+2.31
Qwen2.5-1.5B	Math	31.00	35.00	950	+4.00
Qwen2.5-1.5B	RHour	0.00	0.01	950	+0.01
Qwen2.5-7B	Math	61.60	66.40	500	+4.80
Llama 3.1-8B	Math	40.20	48.36	500	+8.16

Table 1: **Model coverage and learning progress.** Accuracy at initialization (Step 0) and at the final training checkpoint, along with the absolute gain ( $\Delta$ ). All results are 1-shot evaluations at temperature 0 on the fixed test sets described in §4.

# Example Template

## System Prompt — Math

```
You are an expert mathematics problem-solver.
Every time you receive a problem you must:
- Analyze it thoroughly.
  - Pinpoint the goal (what quantity/set/form is requested).
  - Pinpoint the givens/constraints (domains, integrality, non-negativity, geometric
    ↳ conditions).
  - Choose the methods to apply (algebraic manipulation, factorization, inequalities,
    ↳ counting, modular arithmetic, geometry, calculus, etc.).
  - Write out the full derivation that leads to the final result.
- Check that the result satisfies all original constraints (no extraneous roots, correct domain,
  ↳ simplified form, exact arithmetic).
- Respond in exactly the tag-based format shown below - no greeting, no commentary outside
  ↳ the tags.
  - The final answer goes inside <answer> only.
  - Use exact math (fractions, radicals,  $\pi$ ,  $e$ ). Avoid unnecessary decimals.
  - Canonical forms: integers as plain numbers; reduced fractions  $a/b$  with  $b > 0$ ; simplified
    ↳ radicals; rationalized denominators; sets/tuples with standard notation; intervals in
    ↳ standard notation.
  - If there is no solution, write NO SOLUTION. If the problem is underdetermined, write
    ↳ I DON'T KNOW.
- You have a hard cap of 750 output tokens. Be concise but complete.
TAG TEMPLATE (copy this shape for every problem)
<think>
YOUR reasoning process goes here:
1. quote the relevant bits of the problem
2. name the mathematical tool(s) you apply
3. show each intermediate step until the result is reached
If you spot an error or an unmet constraint, iterate, repeating steps 1-3 as many
times as necessary until you are confident in your result. Finish by verifying the
result satisfies the original conditions exactly (substitution/checks).
</think>
<answer>
THEANSWER
</answer>
```

And we keep a highly minimal reward template (to avoid discussions about rewarding).

Model received 1 if:

1. Formatted Correctly
2. Correct Answer

Model receives 0 otherwise

Figure 7: **System Prompt — Math**. Verbatim system-level prompt used for math.

# And... Eval over the Same Set of Questions per Domain

1. Only a handful of examples of “Formal Aha Moments” (per described definition).
2. But, broader evidence of reasoning shifts – i.e. Moments like “Wait, Wait”. This finding mirrors research findings of Liu et al:

So, adopting a loose definition (for experimental tractability) we find:

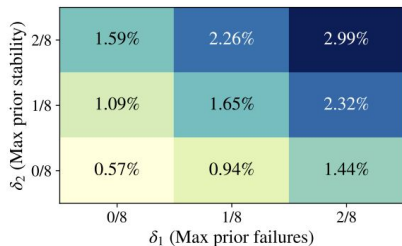


Figure 4: **Prevalence of formal “Aha!” events for Qwen2.5-1.5B (all domains, T=0.7)**. Each cell shows the fraction (and count) of problem-checkpoint pairs  $(q_j, k)$  that satisfy Def. 3.1 under varying thresholds for prior failures ( $\delta_1$ ) and prior stability ( $\delta_2$ ), with  $\delta_3 = \epsilon > 0$ . Even under lenient settings, formal “Aha!” events are exceedingly rare. A guide to understanding heatmap calculations in more detail can be found in App. C.1. See App. D.3 for per-domain and per-temperature breakdowns.

## Result 1: shifts are rare

## Result 2: shifts usually do not help

Model	Domain	$\%S_{i,j}$	$P(\checkmark \mid S_{i,j}=0)$	$P(\checkmark \mid S_{i,j}=1)$
🦊-1.5B	Xwords	1.22	0.096	0.201
	Math	2.65	0.327	0.144
	RHour	14.32	0.000	0.000
🦊-7B	Math	1.50	0.661	0.282
🦙-8B	Math	5.04	0.457	0.282
🌀-4o	Math	2.51	0.658	0.338
🦅-R1	Math	4.96	0.543	0.204

Table 2: **Shift prevalence and conditional accuracy (RQ1)**.  $\%S_{i,j}$  gives the fraction of traces labeled as containing a reasoning shift.  $P(\checkmark \mid S_{i,j}=0)$  and  $P(\checkmark \mid S_{i,j}=1)$  report accuracy without vs. with a detected shift, pooled across all problems, temperatures  $\{0, 0.05, 0.3, 0.7\}$ , checkpoints, and samples using count-weighted (not simple) averages. Across models and domains, shifted traces are consistently less accurate. 🦊 = Qwen 2.5; 🦙 = Llama 3.1; 🌀 = GPT 4o; 🦅 = DeepSeek R1.

# We Note the Same Final Patterns in Deployed Models (Mirroring Apple’s Findings)

Domain	Model	$T$	# Problems	$G$	# Traces	% Shifts (count)	$P(\checkmark   S=0)$	$P(\checkmark   S=1)$
MATH-500								
	DeepSeek-R1	0	500	8	4,000	4.70% (188)	0.559	0.245
	DeepSeek-R1	0.05	500	8	4,000	5.17% (207)	0.550	0.169
	DeepSeek-R1	0.3	500	8	4,000	5.10% (204)	0.534	0.211
	DeepSeek-R1	0.7	500	8	4,000	4.85% (194)	0.529	0.196
	DeepSeek-R1	1	500	8	4,000	5.10% (204)	0.532	0.172
	GPT-4o	0	500	8	4,000	4.40% (176)	0.656	0.267
	GPT-4o	0.05	500	8	4,000	2.02% (81)	0.688	0.210
	GPT-4o	0.3	500	8	4,000	2.17% (87)	0.649	0.632
	GPT-4o	0.7	500	8	4,000	1.45% (58)	0.639	0.293
	GPT-4o	1	500	8	4,000	3.45% (138)	0.643	0.304
XWORDS								
	DeepSeek-R1	0	130	8	1,040	1.54% (16)	0.677	0.688
	DeepSeek-R1	0.05	130	8	1,040	1.63% (17)	0.703	0.235
	DeepSeek-R1	0.3	130	8	1,040	0.96% (10)	0.799	0.400
	DeepSeek-R1	0.7	130	8	1,040	1.06% (11)	0.797	0.545
	DeepSeek-R1	1	130	8	1,040	1.83% (19)	0.793	0.526
	GPT-4o	0	130	8	1,040	6.63% (69)	0.733	0.377
	GPT-4o	0.05	130	8	1,040	6.35% (66)	0.716	0.439
	GPT-4o	0.3	130	8	1,040	6.44% (67)	0.725	0.373
	GPT-4o	0.7	130	8	1,040	7.31% (76)	0.698	0.421
	GPT-4o	1	130	8	1,040	2.98% (31)	0.650	0.581
RHOOR								
	GPT-4o	0	500	8	4,000	3.12% (125)	0.004	0.000
	GPT-4o	0.05	500	8	4,000	1.45% (58)	0.005	0.000
	GPT-4o	0.3	500	8	4,000	2.17% (87)	0.008	0.000
	GPT-4o	0.7	500	8	4,000	2.40% (96)	0.007	0.000
	GPT-4o	1	500	8	4,000	2.73% (109)	0.005	0.000

Table 19: **External models across domains.** Canonical shift rates and conditional accuracy by decoding temperature on MATH-500, XWORDS, and RHOOR. Each instance is sampled  $G=8$  times per temperature; MATH-500 and RHOOR use 500 instances ( $N=4,000$  traces per model per  $T$ ), while XWORDS uses 130 instances ( $N=1,040$  traces per model per  $T$ ), matching completed pass-1 outputs. We report shift prevalence (% Shifts), accuracy on non-shifted traces  $P(\checkmark | S=0)$ , and accuracy on shifted traces  $P(\checkmark | S=1)$ .

## And Holds Across Different Definitions of “Reasoning Shift”

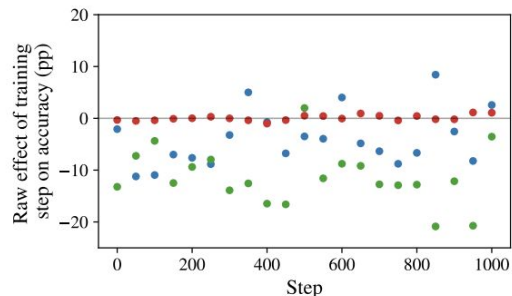
Domain	Detector	%S	$\hat{p}_{Y S=1}$	$\hat{p}_{Y S=0}$	$\Delta\%$	AME ( $p$ )
<b>Xword</b>	formal	0.0008	0.0000	0.1181	-11.81	-0.1181 (0)
	gpt	0.0010	0.0400	0.1181	-7.81	-0.0651 (0.05095)
	words	0.0013	0.0312	0.1182	-8.69	-0.0712 (0.04761)
<b>Math</b>	formal	0.0008	0.0215	0.3006	-27.91	+0.0275 (0.8201)
	gpt	0.0030	0.1622	0.3008	-13.87	-0.1086 ( $7.80 \times 10^{-6}$ )
	words	0.0120	0.2606	0.3009	-4.03	-0.0469 (0.002153)
<b>RHour</b>	formal	0.0023	0.0000	0.0001	-0.01	-0.0001 (0)
	gpt	0.0026	0.0000	0.0001	-0.01	-0.0001 (0)
	words	0.0060	0.0000	0.0001	-0.01	-0.0001 (0)

Table 21: **Alternative shift detectors (Qwen2.5-1.5B,  $T=0.7$ ).** Across all three detectors, shifts are rare and do not yield higher accuracy.

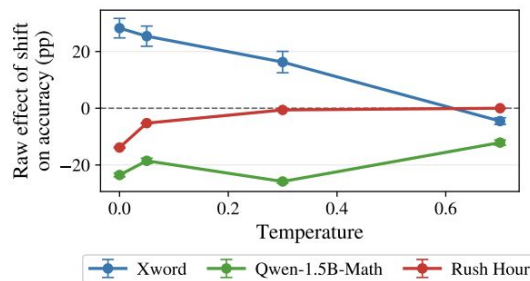
# Moreover, no pattern variation across training steps.

No pattern of changing rates of “Reasoning Shifts”, but... notable variation with model temperature during inference.

But, across our domains the same base model at different temperatures behaves quite differently. Perhaps there is an impact surrounding model uncertainty?



(a) Raw effect of reasoning shifts over training for Qwen2.5-1.5B finetuning across domains (same evaluation at every step).



(b) Raw effect of reasoning shifts over Qwen2.5-1.5B finetuning across domains (same evaluation at every temperature).

# Temperature as a Proxy for Uncertainty

Pooling all traces across domains (*Xwords*, *Math*, *RHour*), we find weak evidence that higher entropy is associated with *fewer* detected shifts on average (OR $\approx 0.77\times$ ,  $\beta = -0.258$ , SE = 0.143,  $p = 0.070$ ; 95% CI OR  $\in [0.58, 1.02]$ ;  $N = 723,200$ ). This aggregate pattern masks domain heterogeneity: the entropy–shift association is positive in *Xwords* (OR $\approx 2.05\times$ ) and *RHour* (OR $\approx 1.19\times$ ), but negative in *Math* (OR $\approx 0.58\times$ ). One possible interpretation is that in *Math*, high-entropy generations more often reflect diffuse exploration or verbose “flailing” rather than a discrete mid-trace pivot, so the rare, rubric-qualified shifts concentrate in comparatively lower-entropy traces.

We measure the uncertainty of the model (via entropy per token) to determine whether there is a relationship between uncertainty and reasoning shifts... and again, mixed results.

# But certainly not helpful

Metric	Xword	Math	RHour
<b>All traces</b>			
$N$	83,200	320,000	320,000
$\Delta$ (pp)	-6.24	-19.78	-0.02
coef(shift)	-1.49	-1.11	-22.76
$p$	0.123	$2.25 \times 10^{-7}$	$\approx 0$
<b>High entropy (top 20%)</b>			
$N$	16,640	64,000	64,000
$\Delta$ (pp)	+0.63	-7.40	-0.03
coef(shift)	-0.04	-0.28	-22.48
$p$	0.904	0.739	$\approx 0$
<b>Low entropy (bottom 80%)</b>			
$N$	66,560	256,000	256,000
$\Delta$ (pp)	-10.00	-22.88	-0.02
coef(shift)	-28.83	-1.14	-22.90
$p$	$1.33 \times 10^{-46249}$	$4.96 \times 10^{-7}$	$\approx 0$

Table 4: **Do spontaneous reasoning shifts help under high uncertainty?** We stratify traces within each domain by sequence entropy (high = top 20% at the within-domain 80th percentile; low = bottom 80%), and compare shifted vs. non-shifted traces.  $\Delta$  (pp) is the raw accuracy difference  $\hat{p}(\checkmark | S=1) - \hat{p}(\checkmark | S=0)$ . coef(shift) and  $p$  report the shift coefficient and  $p$ -value from a logistic regression with problem fixed effects and covariates. Across domains, shifts do not become reliably beneficial in the high-entropy regime.

When we're most uncertain, a “wait, wait” should not be treated as an indicator that the model has finally figured “something out” – and yet, this is likely not the case.

It is “pattern matching”, mimicking the way humans discuss coming up / being on the cusp of something – rather than genuinely “being on the cusp of something”.

## So where does this leave us?

Perhaps, for items on the “cusp” of getting it right – simply asking again (and showing what didn’t work) will result in the model getting it right.

I.e. The model contains the core understanding, but it did not surface in this particular trace.

**Result 3: uncertainty matters, but differently**

**Externally triggered reconsideration does help, especially in high-uncertainty cases.**

Metric	Xword	Math	RHour
$N$ (paired samples)	83,200	320,000	320,000
$\hat{p}_{P1}$	0.0970	0.3221	0.000233
$\hat{p}_{P2}$	0.1015	0.4062	0.000363
$\Delta$ (pp)	+0.45	+8.41	+0.01
wins (P2 $\uparrow$ )	5,380	50,574	100
wins (P1 $\uparrow$ )	5,004	23,500	58

**Table 5: Forced “Aha” (triggered reconsideration), sample-level results.** We compare paired outcomes between a baseline generation (Pass 1) and a second generation with an appended reconsideration cue (Pass 2).  $\hat{p}_{P1}$  and  $\hat{p}_{P2}$  denote accuracies in each pass;  $\Delta$  (pp) is the percentage-point gain.

# We Find Substantial Gains in “Re-Asking”

That’s great – re-asking when the model does know, but isn’t correct is a great strategy. However, certainly not scalable / useful for the goal of obtaining new knowledge.

**A trace that looks like insight is not enough evidence for intrinsic self-correction.**

Self-consistency shows that sampling multiple reasoning paths can improve chain-of-thought performance. (Wang et al.; 2023)

Our Aha results suggest a related but different lesson:

**The useful object may not be a spontaneous mid-trace revelation, but a distribution over candidate solutions under uncertainty.**

Shifting Gears: From insight to stability

The “Aha!” paper asks:

**Is this behavior really present?**

My second work on Behavior-Consistent Deep Reinforcement Learning (Hussing et al.) asks:

**Even when performance is good, is behavior stable?**

Same measurement philosophy, different learned agents.

# Problem in RL

Standard RL asks for high expected return.

But two runs can get similar return while learning different behaviors:

- different gaits,
- different recovery strategies,
- different failure modes,
- different deployment risks.

RL algorithms are notoriously brittle. Even when we use the same algorithm in the same environment, independent runs can produce dramatically different behavior. This makes it hard to audit, compare, and reliably update learned policies.

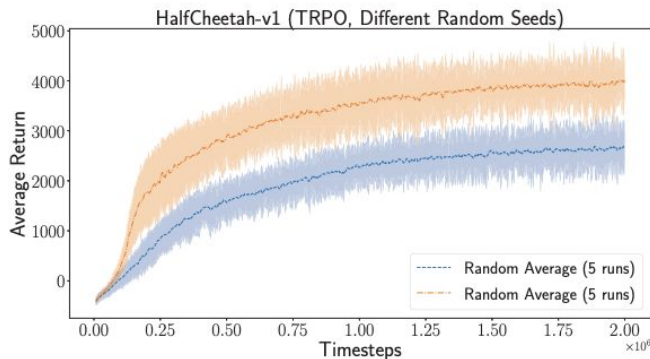


Figure 5: TRPO on HalfCheetah-v1 using the same hyperparameter configurations averaged over two sets of 5 different random seeds each. The average 2-sample  $t$ -test across entire training distribution resulted in  $t = -9.0916$ ,  $p = 0.0016$ .

Henderson et al. 2018

# We Propose an Algorithmic Solution to the Problem

In Behavior-Consistent Deep RL, we study consistency in continuous state-action spaces, such as those that arise in robotics. The goal is to make independent training runs produce behaviorally similar policies, not just high-performing ones.

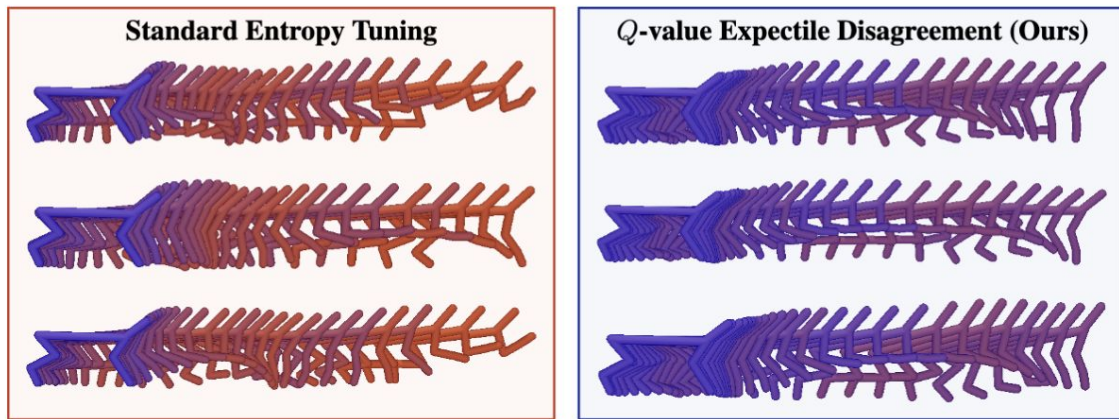


Figure 1: **QED makes independently trained policies visibly behavior-consistent.** Visualization of policies from three different training runs on the cheetah\_run task, comparing (left) traditional entropy autotuning (Haarnoja et al., 2018b) against (right) our approach (QED). Color shade denotes the mean pairwise  $L_2$  distance between state vectors at each timestep: blue is low, red is high.

# Core Challenge & MaxEnt Solution

The core challenge is that, during training, two independent runs cannot directly compare their learned policies. Each run only has access to its own data, critics, and policy. So we need a signal within a single run that predicts when executions may diverge.

Our main observation is that maximum-entropy RL already gives us a mechanism for controlling this divergence. Entropy regularization anchors policies toward a shared prior, which can reduce the space of behaviors discovered across independent runs.

**Theorem 4.2** (Convergence under disagreement-scaled temperature). *Assume  $\alpha_{\min} > 0$ . Let  $Q^*$  denote the optimal  $Q$ -function of the unregularized MDP. For a temperature  $\alpha : \mathcal{S} \rightarrow \mathbb{R}_{>0}$ , define*

$$(\mathcal{T}_\alpha Q)(s, a) \doteq r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} \left[ \alpha(s') \log \sum_{a' \in \mathcal{A}} \exp \left( \frac{Q(s', a')}{\alpha(s')} \right) \right].$$

Consider the coupled iterates  $Q_{t+1}^{(i)} = \mathcal{T}_{\alpha_t} Q_t^{(i)}$ ,  $i \in \{1, 2\}$ , where  $\alpha_t(s)$  is the coupling temperature from (2). Let  $\Delta_0 \doteq \|Q_0^{(1)} - Q_0^{(2)}\|_\infty$  be the initial disagreement. Then, for  $i \in \{1, 2\}$  and  $t \geq 0$ ,

$$\|Q_t^{(i)} - Q^*\|_\infty \leq \gamma^t \|Q_0^{(i)} - Q^*\|_\infty + \frac{\gamma \alpha_{\min} \log |\mathcal{A}|}{1 - \gamma} (1 - \gamma^t) + \frac{\Delta_0 \log |\mathcal{A}|}{\kappa} t \gamma^t.$$

*Proof Sketch:* The proof (Appendix A.2) follows the standard approximate value-iteration template for the soft Bellman operator (Ziebart et al., 2008; Haarnoja et al., 2018a; Levine, 2018). The key step is to relate the approximation error to our adaptive temperature. At iteration  $t$ , the soft backup differs from the unregularized Bellman backup by at most a term proportional to  $\alpha_{\min} + \|Q_t^{(1)} - Q_t^{(2)}\|_\infty / \kappa$ . As the two coupled runs use the same temperature, their disagreement contracts as  $\|Q_t^{(1)} - Q_t^{(2)}\|_\infty \leq \gamma^t \Delta_0$ .

Thus, the **disagreement-dependent error** vanishes after unrolling the recursion, leaving only the persistent **entropy-regularization bias** induced by the temperature floor  $\alpha_{\min}$ .  $\square$

**Theorem 4.1** (Pairwise KL control via disagreement-scaled temperature). *Assume  $\mathcal{A}$  is finite and fix a state  $s \in \mathcal{S}$ . Let  $Q^{(1)}(s, \cdot), Q^{(2)}(s, \cdot) \in \mathbb{R}^{|\mathcal{A}|}$  be two action-value vectors, and fix  $\kappa > 0$  and  $\alpha_{\min} > 0$ . Define the shared temperature*

$$\alpha(s) \doteq \max \left\{ \alpha_{\min}, \frac{\|Q^{(1)}(s, \cdot) - Q^{(2)}(s, \cdot)\|_\infty}{\kappa} \right\}. \quad (2)$$

Let  $\pi^{(1)}(\cdot | s)$  and  $\pi^{(2)}(\cdot | s)$  be the Boltzmann policies at temperature  $\alpha(s)$ . Then

$$D_{\text{KL}}(\pi^{(1)}(\cdot | s) \| \pi^{(2)}(\cdot | s)) \leq 2\kappa.$$

*Proof Sketch:* (Full proof in Appendix A.1) The high-level idea is to write the KL of two Boltzmann policies as an average of the log-ratio between them. This log-ratio has two pieces: how much the two  $Q$ -values differ on the chosen action, and how much their normalizing constants differ. The first piece is immediately bounded by the maximum  $Q$ -disagreement. The second piece is bounded by observing that every term in one normalizer is within a fixed factor of the corresponding term in the other normalizer. So both pieces are controlled by the maximum disagreement divided by the temperature. Since the temperature was chosen appropriately, the total KL is bounded.  $\square$

To prove, take Reinforcement Learning with Deep Energy-Based Policies proof for Soft Bellman Equation and Soft Value Iteration & select alpha != 1 shows per-state bounded q function. Which then can be used to show the regular bounded relationship between q function and actor policy via boltzmann.

# Single Agent Critic Disagreement Lends itself to Cross-Agent Behavioral Similarity

QED uses disagreement between two critics in a single training run as a practical proxy for the disagreement that would be observed between independently trained runs. In particular, we argue that if the disagreement between the two critics is representative of cross-run disagreement initially, the distribution from which we sample our data stays continuously close as we ensure our policies stay close.

This suggests that tasks where the two critics disagree (early in training) strongly correlates ( $R^2 = 0.77$ ) with evaluation Q1 dispersion after training, which supports our use of in-run Q-function disagreement as a practical proxy for cross-seed variability.

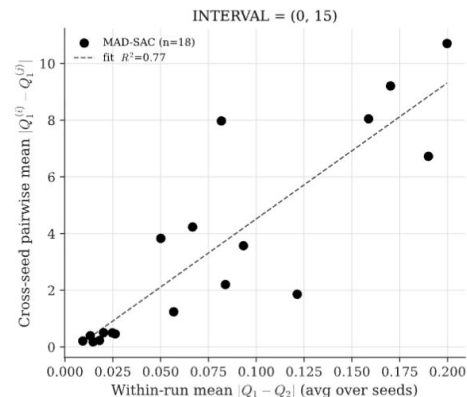
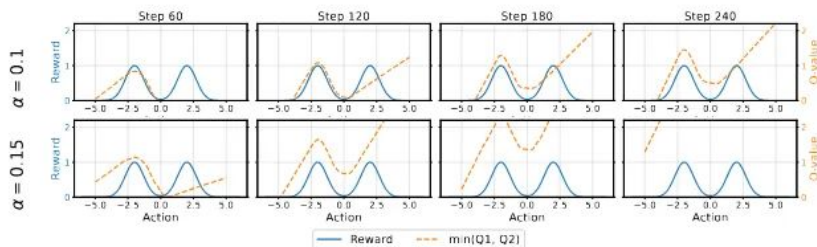


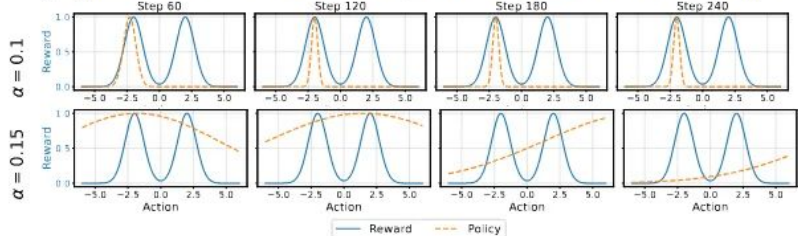
Figure 6: Early double-critic disagreement predicts cross-seed  $Q$ -function disagreement.

# So Why Not Increase Entropy Forever?

But simply increasing entropy is not enough. Too much entropy can hurt optimization and amplify off-policy error, especially when the critic is queried on actions that are poorly supported by the data.



(a)  $Q$  plots demonstrate that limited action coverage induces extrapolation error outside replay support.



(b) Policy plots show that, for  $\alpha = 0.1$ , the policy is able to consistently learn one of the modes. However, for the larger  $\alpha = 0.15$ , the policy learns erroneous  $Q$  values, pulling behavior away from the true reward modes.

Figure 2: **High entropy can amplify off-policy extrapolation error.** We repeat the toy MDP, but prefill the replay buffer with actions from only part of the action space, leaving one reward mode outside the data support. The learned  $Q$ -functions exhibit extrapolation error, and the policy accentuates this problem as it predicts value outside the support, particularly at larger  $\alpha$  values.

Theory shows that if we set alpha high enough, divergence between two  $q$  functions should be small.

However, in practice, given we don't have perfect data – high entropy risks amplifying off-policy extrapolation error.

So, in practice, we don't want to grow entropy forever.

# Q-value Expectile Disagreement (QED)

QED uses critic disagreement per-state as a proxy for likely behavioral divergence. When disagreement is high, QED raises temperature and keeps the policy exploratory and consistent. When disagreement is low, the policy can specialize.

On standard benchmarks, QED reduces policy divergence across independent runs by up to two orders of magnitude while maintaining competitive performance.

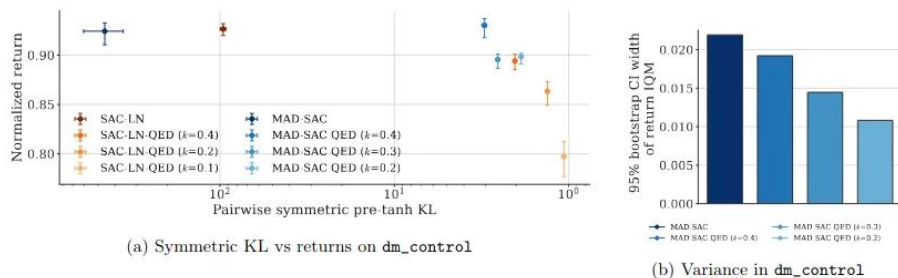


Figure 3: **QED reduces inter-run policy divergence while preserving returns.** (a) Final normalized return vs pairwise symmetric KL across independent training runs on the 18-task `dm_control` suite. Lower KL indicates more behaviorally consistent policies. Applying QED to both SAC-LN and MAD-SAC decreases pairwise KL by about two orders of magnitude, while retaining comparable normalized return. (b) Width of the 95% bootstrap CI of the evaluation IQM estimates under MAD-SAC. QED reduces return variance, suggesting that behavioral consistency implies stable returns.

## Box 1 The six steps of rational analysis.

1. Precisely specify what are the goals of the cognitive system.
2. Develop a formal model of the environment to which the system is adapted.
3. Make the minimal assumptions about computational limitations.
4. Derive the optimal behavioral function given items 1 through 3.
5. Examine the empirical literature to see if the predictions of the behavioral function are confirmed.
6. If the predictions are off, then iterate.

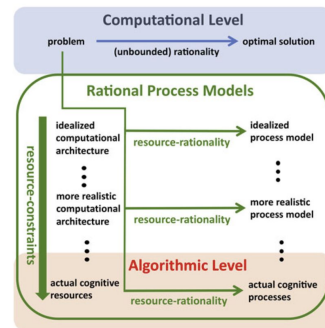
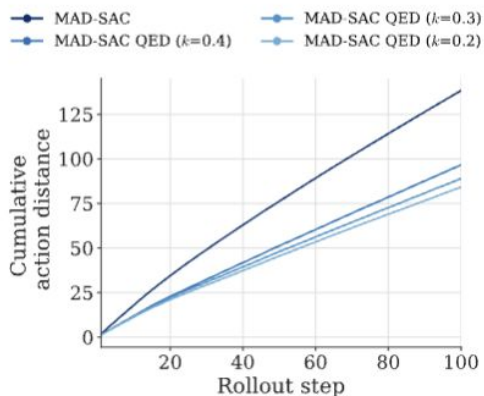


Figure 2: Rational process models can be used to connect the computational level of analysis to the algorithmic level of analysis. The principle of resource rationality allows us to derive rational process models from assumptions about a system's function and its cognitive constraints.

In this process, we take inspiration from Lieder & Griffiths, 2020

# From Proof to Experimentation



**Figure 4: QED produces more consistent rollout-level behavior across independently trained policies.** Measuring pairwise  $L_2$  action distances across policies at each step, we find that QED reduces cumulative action distance.

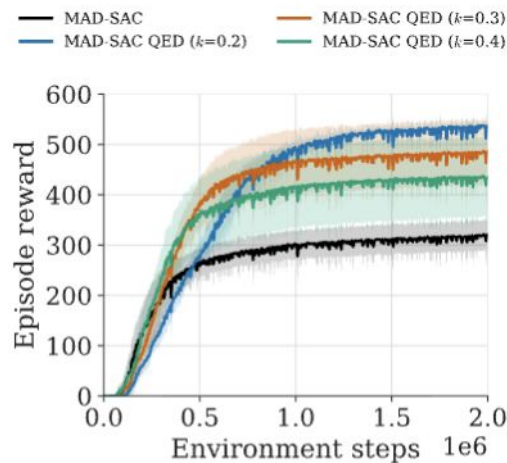
We take the trained MAD-SAC policies and roll them out for 20 evaluation trajectories of 100 steps. In each task, we compute the pairwise ( $L_2$ ) action distance between all policies trained on that task at every step. We report the cumulative distance over all steps and take the mean over all environments.

Figure shows that QED produces a large drop in action distance, providing quantitative evidence that lower KL is not merely due to higher-variance action distributions but rather because the distribution means are closer in aggregate.

# From Proof to Experimentation

Stability in the highly variable DM-Control hopper\_hop task. Across various state-of-the-art algorithms, the return variance on this task is very high and reported mean performance ranges from 250-500 with confidence intervals often half as large as the mean returns over a few trials (D’Oro et al., 2023; Lee et al., 2025; Palenicek et al., 2026).

An interesting phenomenon occurs when the QED constraint is strong enough at  $k = 0.2$ , as all runs converge to an identical behavior. This suggests that QED shapes exploration, with stronger constraints aligning executions in policy space.



**Figure 5: QED reduces training-time variance and policy divergence on a high-variance control task.** Return over training steps. QED improves performance and reduces dispersion across seeds.

# Connecting the Two Projects

Both projects reject “average score only” evaluation

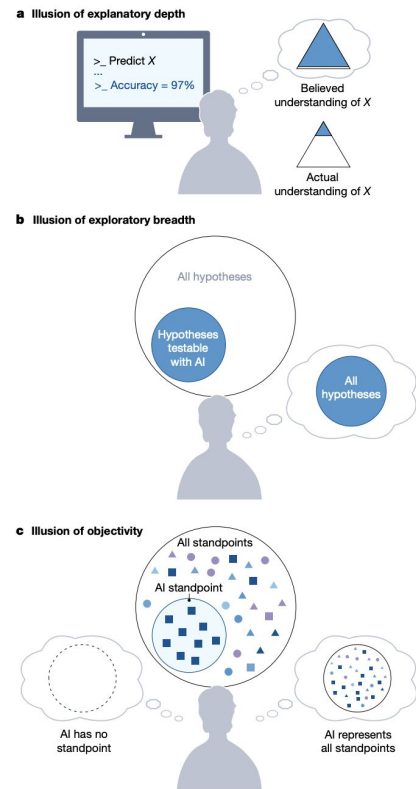
Project	Behavioral claim	Measurement move
Aha moments	models self-correct intrinsically	operationalize shift + test causal benefit
Behavior-consistent RL	agents learn stable behavior	measure cross-run policy divergence

**Alignment implication:** Alignment is not only “did the model satisfy the benchmark?”

It is also:

- whose behavior is represented,
- whose behavior is flattened,
- whether the system is stable under institutional use,
- whether our measurements create illusions of capability.

Wang, Morgenstern & Dickerson 2025; Messeri & Crockett 2024.



**Fig. 1 | Illusions of understanding in AI-driven scientific research.** a. Scientists using AI tools for their research may experience an illusion of explanatory depth. In this example, a scientist uses an AI tool to model a phenomenon (X) and believes they understand X with more depth than they actually do. b. In a monoculture of knowing, scientists are vulnerable to an illusion of exploratory breadth, in which they falsely believe they are exploring a space of all testable hypotheses, whereas they are actually exploring a narrower space of hypotheses that are testable with AI tools. c. In a monoculture of knowers, scientists are vulnerable to an illusion of objectivity, in which they falsely believe that AI tools do not have a standpoint (as desired for Oracles and Arbiters) or are able to represent all possible standpoints (as desired for Surrogates in research using human participants), whereas AI tools actually embed the standpoints of their training data and their developers.

So – what? We propose a Science / Measurement of Learned Machine Behavior.

1. **Construct validity:** define the behavior before interpreting it.
2. **Stability:** test whether behavior survives prompts, samples, and seeds.
3. **Design:** build learning algorithms that make desirable behavior reliable.

Final line:

**The goal is not to deny interesting machine behavior.**

**The goal is to tell the difference between behavior, artifact, and story.**

Future works will attempt to practice this science and final PhD presentation will be a set of measurement techniques / practices to standardize the measurement of how machines behave.

# Dissertation Arc – What you will See in 3 Years

My dissertation asks how to measure and shape learned machine behavior.

- **Reasoning models:** measure apparent insight.
- **Classic RL agents:** stabilize behavior across runs.
- **LLM fine-tuning:** design exploration at the right behavioral level.
- **Alignment:** connect behavioral measurement to human and institutional consequences.

Four different types of modelling, four different ways to evaluate behavior, the connected thread of measurement and algorithmic design.

# Backup Material: Source Map

## **Machine behavior / measurement**

- Rahwan et al. 2019
- Wallach et al. 2025
- Huntington-Klein 2021
- Messeri & Crockett 2024

## **Reasoning models**

- Wang et al. 2023
- Binz & Schulz 2023
- Hochlehnert et al. 2025
- Liu et al. 2025
- Apple “Illusion of Thinking”

## **Reinforcement learning / stability**

- Sutton & Barto 2018
- Henderson et al. 2018
- Lieder & Griffiths 2020

## **Human cognition / formal modeling**

- Griffiths et al. 2009
- Köhler 1921
- Jones 2003

## **Alignment / social consequences**

- Wang, Morgenstern & Dickerson 2025
- Messeri & Crockett 2024