SEMINAR



Jev : Machine Native Intelligence for Automation

TypeSafe AI | 2026. 10

01. Motivation

Why Did Diogo Almeida Leave OpenAI?

"When we launched InstructGPT, we thought, 'Did we just build AGI?'

But when you look at the actual adoption, 90% of it was just copywriting and sales emails—it completely missed true economic automation."



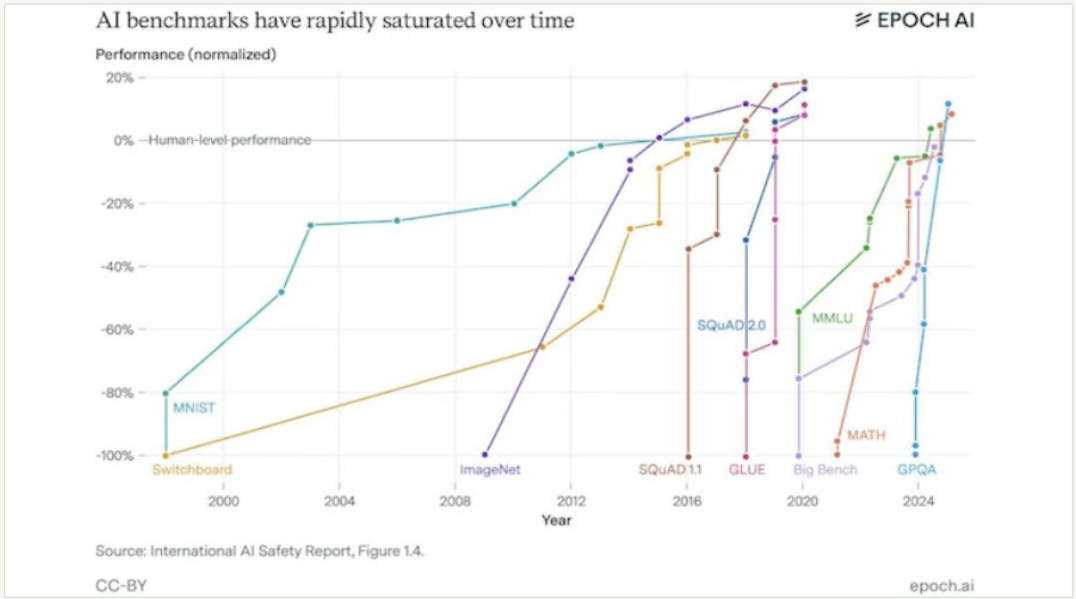
Diogo Almeida

Founder & CEO, TypeSafe

ex-OpenAI researcher

Co-inventor of RLHF, InstructGPT-4

AI is Overpromise



But Underdeliver

Microsoft CEO Admits That AI Is Generating Basically No Value

Exist LLMs failed to Automation

Too good to be true

- Instruction following
- ChatGPT
- Copilots
- Coding agents / Claude Code
- NotebookLM
- DeepResearch
- Voice agents
- Customer service w/o decisions
- Slop / spam

Assistant

Too bad to be useful

- Customer service
- Drive-thrus
- Accounting / taxes
- QA
- Insurance
- Data entry
- **Anything without a human in the loop**
- **Anything requiring reliability**

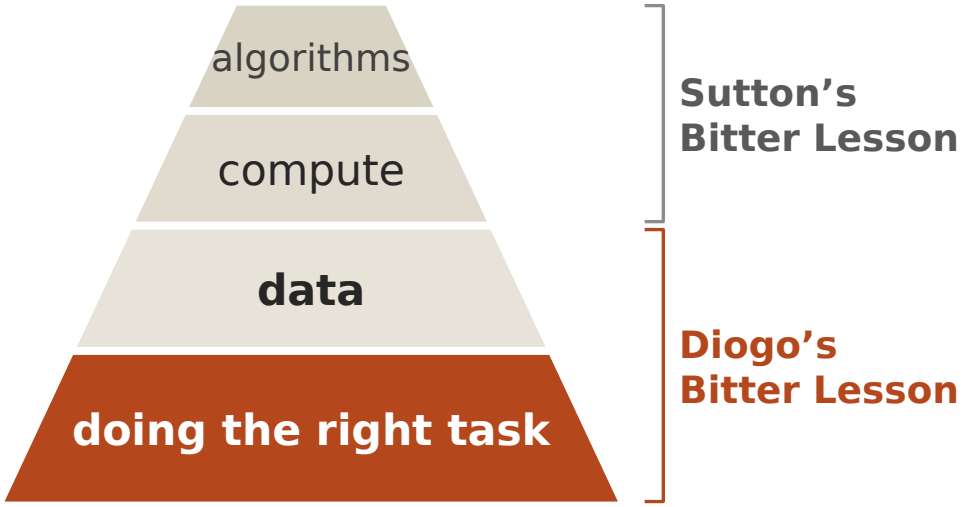
Automation

Why RLHF failed to Automation

**RLHF optimized models for human assistance,
but degraded their capacity for independent automation."**

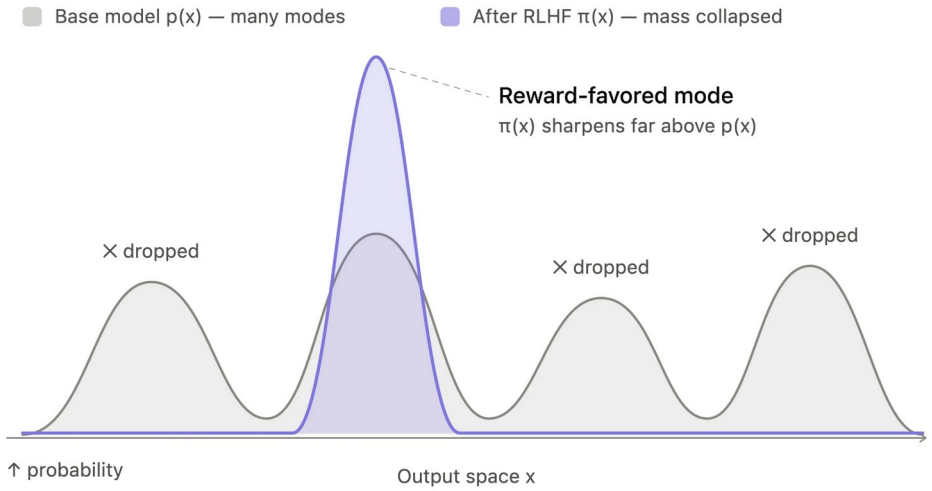
01 Optimizes for human preference

Rewards the answer people like,
not the logical decision



02 Mode collapse → broken calibration

p(x) no longer carries meaningful information



RLCD : RL for Calibrated Decision

- Confidence (q) : Estimated probability of answer correctness ($c = 1$).
  - Use Verbalized confidence in LLM :
    - Ask the LLM "How confident are you that your answer is correct?" and use its outputs.
- Calibration : Estimated probability is equal to t when $q = t$

$$\Pr(c = 1 | q = t) = t$$

- RL with Calibration :

$$\begin{aligned} R_{\text{RLCR}}(y, q, y^*) &= R_{\text{correctness}}(y, y^*) + R_{\text{Brier}}(y, q, y^*) \\ &= \mathbb{1}_{y \equiv y^*} - (q - \mathbb{1}_{y \equiv y^*})^2 . \end{aligned}$$

Scoring Rule is minimized when calibrated

Damani, M., Puri, I., Slocum, S., Shenfeld, I., Choshen, L., Kim, Y., & Andreas, J. (2026, April).
Beyond binary rewards: Training LMs to reason about their uncertainty. In International conference on learning representations (Vol. 2026, pp. 59456-59493).

Machine Native Intelligence for Automation

Paradigm shift

Human ↔ AI
Chat interface



AI ↔ Software & AI
Machine interface

Machine Native Intelligence

AI with software-like properties

System One	Fast, cheap judgment, run millions of times Speed · Low cost
Calibrated decisions	Knows when to act and when to escalate Reliability · Observability
Composable	Plugs into software and other AI through a stable interface Structure · Consistency · Testability

TypeSafe AI : Building Product, Not God

> **Our mission is to pave the shortest path to an AI-based economic revolution by making intelligence composable to catalyze a Cambrian explosion of intelligent software.**
>
> — Our mission, TypeSafe AI

System One Model

Highest possible intelligence-per-dollar

Ship the shape of machine-native, composable AI.

Calibrated decisions

Make our AI reliable

Reliable enough to transform the economy through real automation.

Composable

Stable enough to compose and layer upon

Higher-level intelligence abstractions for a collaborative, emergent future.

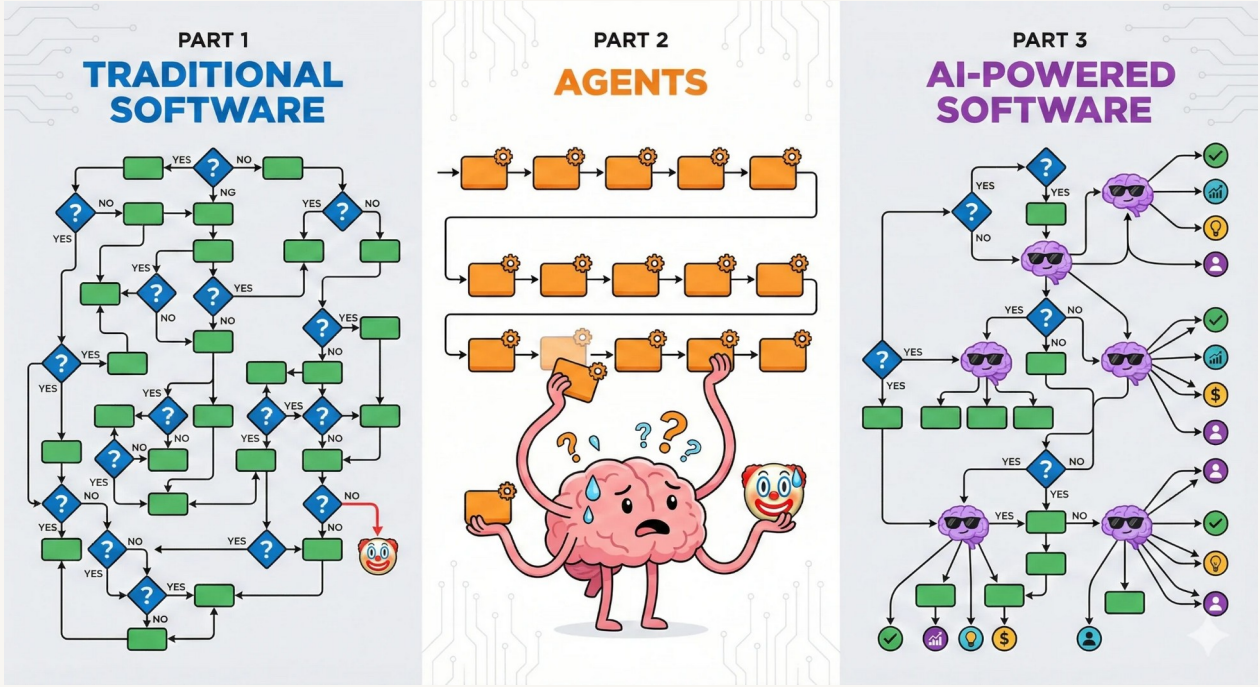
Jev : The First System One Model

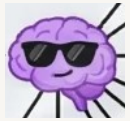
Existed LLMs		System One Model
		Fast, structured decisions software can use directly
 ChatGPT  Claude Code		 Jev
Human ↔ AI	INTERACTION	AI ↔ Software & AI
Human preference / Verifiable rewards	TRAINING OBJECTIVE	Calibrated decisions
RLHF / RLVR	OPTIMIZATION	RLCD
Natural Language	OUTPUT	Structured Output + confidence

02. How to use Jev

Composable AI with Jev

Build a normal software workflow, and insert Jev only where AI is needed.





: Jev

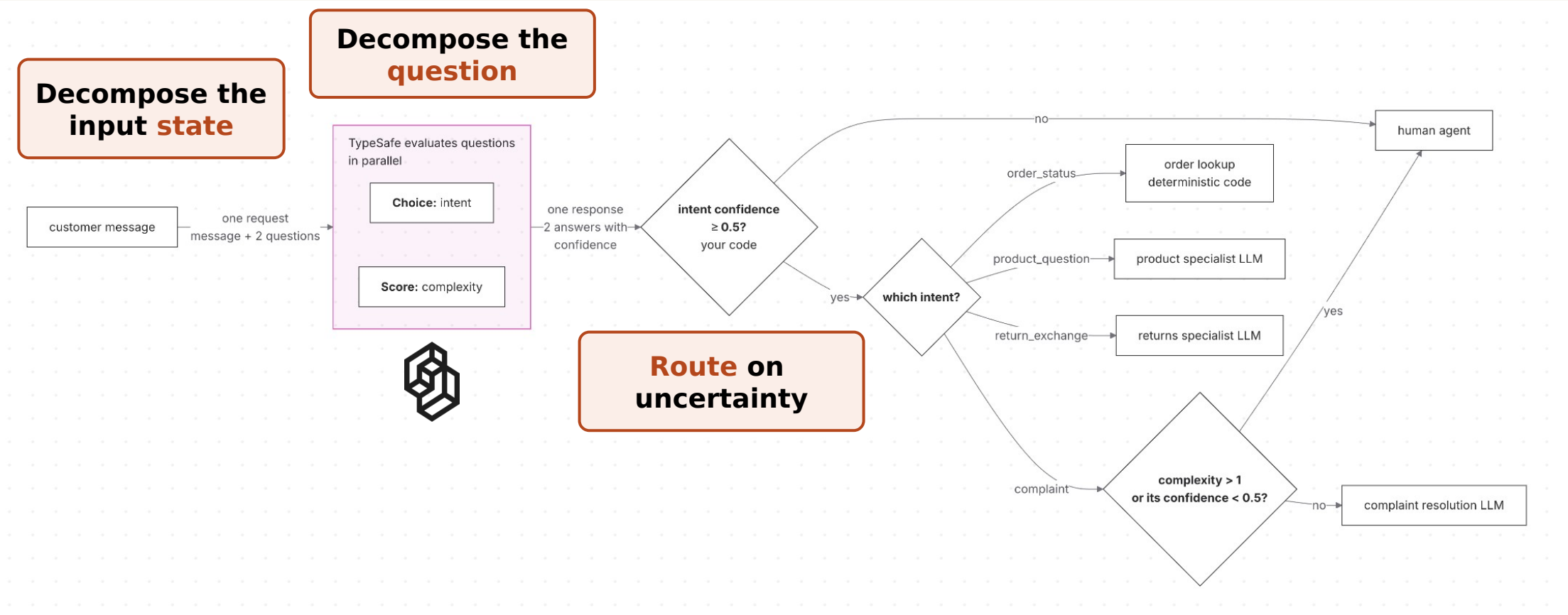
Traditional SW
Reliable primitives compose into higher-level abstractions

Agents
Model picks each next step;
every loop can go off the rails

AI-Powered SW
Code owns control flow; Jev handles atomic, constrained judgments

Source: How to build with TypeSafe

AI-Powered Workflow with Jev



Three primitives of Jev

Code owns the flow. **Jev** answers typed questions about the state.

3 PRIMITIVES & CONFIDENCE

Choice

ASKS

What does the sender want from us?

RETURNS **lead 0.89**

customer	0.04
lead	**0.89**
vendor	0.04
fraud	0.01
other	0.02

CONFIDENCE

$$\frac{p_{max} - 1/n}{1 - 1/n}$$

Score

ASKS

How well is this claim supported?

RETURNS **1.43**

0	0.00
1	**0.57**
2	0.43

Score = $E[X]$

$0 \times 0 + 1 \times 0.57 + 2 \times 0.43 =$ **1.43**

CONFIDENCE

$$D = E[|X - m|]$$
$$\text{where } m = \text{mode}(X)$$

Noul

ASKS

payment or a login?

RETURNS **P(yes) 0.93**



likely no unsure likely yes

0.5 means it can’t tell, not medium risk. Need levels? Use a Score.

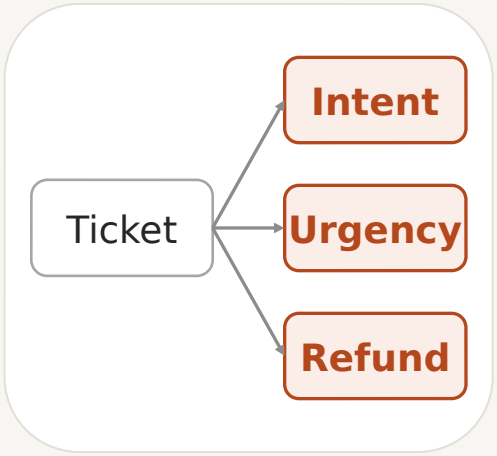
CONFIDENCE

$$|2\,P(yes) - 1|$$

Four patterns for building with Jev

Think in **atomic decisions** that compose into system behavior.

ASK MANY AT ONCE



Speculative Fan-Out

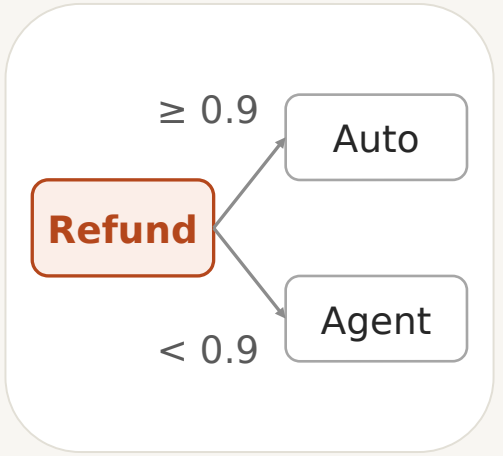
Ask a ticket's intent, urgency and refund eligibility in one call



Composite Scoring

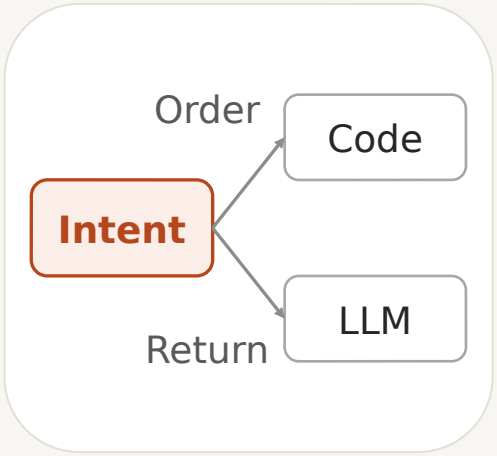
Rank leads by budget fit, urgency and company size, all scored at once

ROUTE ON A DECISION



Confidence-Gated Routing

Auto-approve a refund at 0.92 confidence; send 0.55 to an agent



Intent Routing

Send “Where’s my order?” to code and “It arrived broken” to a returns LLM

Orange nodes are Jev decisions. Source: TypeSafe docs, Patterns

02. How to use Jev

Eight use cases, two ways to build with Jev

ASK MANY AT ONCE

Speculative Fan-Out

Verify an agent finished AGENT
Does the result meet each acceptance criterion?
Missing or stale evidence keeps the task open.

Screen LLM input or output SAFETY
One question per hazard.
Code maps the answers to pass, review or block.

Triage a support inbox DATA
Queue, urgency and escalation asked
as three questions over the same ticket.

Composite Scoring

Combine scored dimensions DATA
Urgency, risk, value and fit scored separately,
then weighted in code.

ROUTE ON A DECISION

Intent Routing

Route to model, code or human AGENT
Picks the path a request needs;
a difficulty score keeps hard requests off the cheap path.

Check a citation against a claim RETRIEVAL
Support, contradict or not address?
Code still checks that the quote and URL are real.

Confidence-Gated Routing

Filter RAG passages RETRIEVAL
Does this passage answer the question, or only keywords? Runs after normal retrieval.

Apply a fixed evaluation rubric SAFETY
Rates an answer on one narrow criterion
as a typed verdict code can act on.

Implementation patterns, not measured deployments. Source: Eric Kang, “18 Practical JEV Use Cases for AI Agents,” Hugging Face, Sep 2026

03. Research question

03. Research question

Research Around Jev

Researched by ChatGPT

초기 리뷰에서 관련 arXiv 논문 **28 편** 검토 · 대상 기간 2026.09.19–09.24 · Jev 공개 2026.09.15

01 Evaluating Jev

성능 · 효율 검증

핵심 질문

기존 LLM· 분류기보다 정확하고 저렴한가 ?

대표 연구

Evaluating and Benchmarking Jev

Fast Models, Slow Evidence

02 Understanding Failures

편향 · 견고성 분석

핵심 질문

선택지 표현이나 입력 조작에 얼마나 흔들리는가 ?

대표 연구

JevAdvBench

Ordinal-Scale Bias

03 Building Decision Models

대안 모델 · 학습법 개발

핵심 질문

기존 모델을 어떻게 판단 모델로 바꿀 수 있는가 ?

대표 연구

LLM2Jev

SBERT2S1

04 Using Decisions in Systems

에이전트 · 응용 확장

핵심 질문

판단 모델을 어디에 넣어 전체 작업을 개선할까 ?

대표 연구

LAVOIR

Just Ask Jev

ChatGPT 가 조사 · 정리한 내용으로 , 원문 확인 전입니다 . 28 편은 해당 리뷰의 수집 범위이며 전체 등재 수가 아님 .
집계 출처 : Typed Decision Models: An Early Evidence Audit and Evaluation Checklist

Idea…

- Jev를 활용해서 trajectory에 대한 failure 분석 개선
  - Trajectory 자체가 너무 길기 때문에, 여러 스텝에 걸친 failure 원인 분석 어려움 -→ Jev를 통해 탐색 범위 효율적 조정 후 분석
- Jev에 어떻게 State, Question를 분해해서 줄 것인가?
  - Trajectory 기반으로 Question을 생성 (meta-jev)
  - 단순 thresholding에 의존하는데, 이것이 성립하지 않는 경우가 생길 수 있음
- **에이전트의 연산을 통제하는 판단 모델을 어떻게 구성할 것인가?**
  - Jev prior confidence : $p_\theta(a \mid x, q, \mathcal{A}) \neq$ Harness (H) given confidence $p_\theta(a \mid x, q, \mathcal{A}, H)$
    when action is not independent to H