



Harnesses for Generalisation

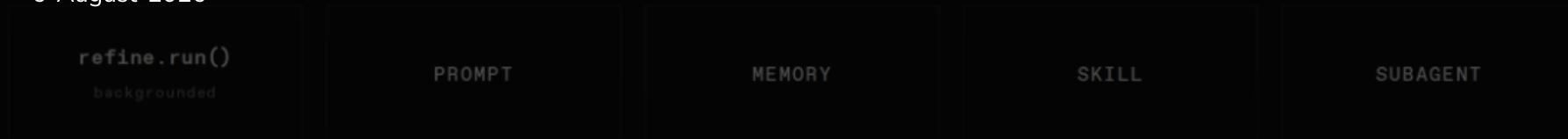
Machine Learning Singapore #MSLG

- Martin Andrews = Generalisation @ mdda.net

CONTINUAL HARNESS

reads the trajectory across turns, writes back to harness state

6-August-2026



About Me

- Machine Intelligence / Startups / Finance
 - Moved from NYC to Singapore in Sep-2013
- 2014 = 'fun' :
 - Machine Learning, Deep Learning, NLP
 - Robots, drones
- Since 2015 = 'serious' :: NLP + deep learning
 - Including Papers...
 - & GDE ML; ML-Singapore co-organisier...
 - & Red Dragon AI...



Outline

- Looking at a flow of work:
 - DSPy
 - Recursive Language Models
 - Harnesses for Generalisation
- BONUS : Prime Agent launch
- In the news...
- Wrap-up & QR-code

DSPy

DSPy blurb

- DSPy is an LLM Framework

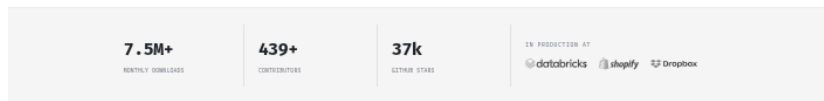
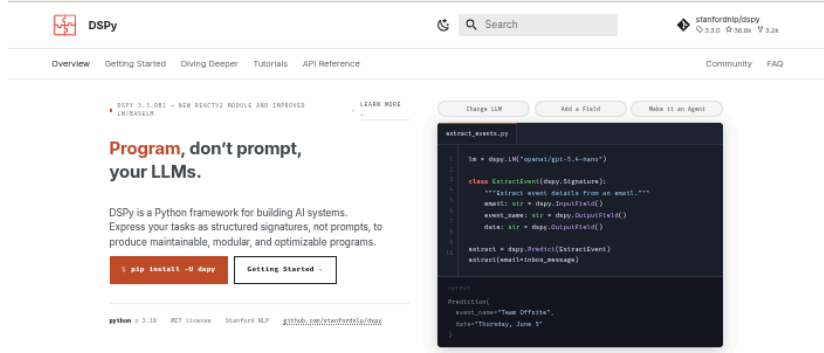
- Can call multiple backends
- Orchestrate 'flows', etc
- ... but has a v. different 'feel'

- MIT Licensed Code Repo

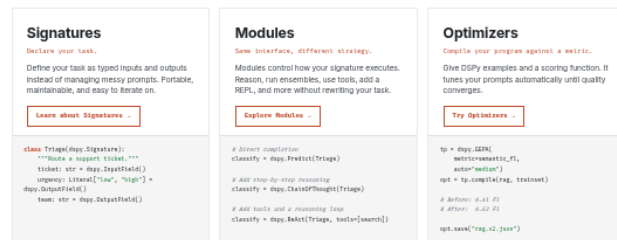
- Official Tutorials

- DSPy @ MLSG :

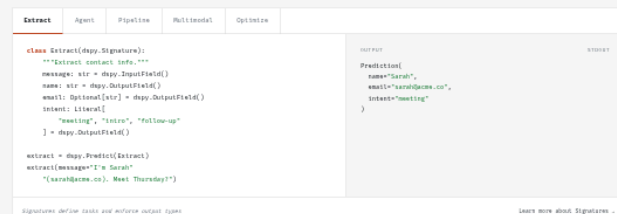
- March-2024
- June-2025



Compose programs with reusable primitives.



Define a task. Grow it into a system.





DSPy Haiku Generator

■ This ~works

```
import dspy
dspy.settings.configure(
    lm=dspy.LM("gemini/gemini-3.5-flash")
)

haiku_signature = "subject → haiku"
haiku_generator = dspy.Predict(haiku_signature)
res = haiku_generator(subject="computer science")
print(res.haiku)

# Silent code unfolds
# Logic threads through hidden paths
# Bugs bloom, then resolve
```

■ This ~doesn't work

```
haiku_bot = dspy.Predict("a → b")
res = haiku_bot(a="computer science")
print(res.b)
```



Stricter Haiku ...

```
from typing import Literal

Season = Literal[ "spring", "summer",
                  "autumn", "winter", ]

class HaikuBot(dspy.Signature):
    """
    Write a classical haiku given
    the provided inputs.
    """
    location: str = dspy.InputField()
    mood: str = dspy.InputField()
    season: Season = dspy.InputField()
    haiku: str = dspy.OutputField()

haiku_bot = dspy.Predict(HaikuBot)
result = haiku_bot(location="Bodega Bay",
                   mood="mysterious",
                   season="autumn")

print(result.haiku)

# Grey fog veils the coast,
# Silent wings lost in the mist,
# Autumn's deep secret.
```


Measuring Performance

```
def haiku_score(example, prediction,
                 trace=None, pred_name=None, pred_trace=None):
    """
    Penalize verbatim use of the input season string.
    A haiku should evoke the season through imagery,
    not name it directly.
    """
    text = prediction.haiku.lower()
    if example.season.strip().lower() in text:
        return dsp.Prediction(
            score=0.0,
            feedback="Don't reference the input season verbatim."
        )
    return dsp.Prediction(score=1.0, feedback=None)
```

- and evaluate on validation data...

```
evaluate = dsp.Evaluate(devset=val, metric=haiku_score)
baseline_score = evaluate(haiku_bot)
print(baseline_score)
```



DSPy Optimisation

```
optimizer = dspy.GEPA(  
    metric=haiku_metric,  
    reflection_lm = dspy.LM("gemini/gemini-3.5-flash"),  
    auto="light",  
    num_threads=2,  
)  
  
optimized_haiku_bot = optimizer.compile(haiku_bot,  
                                       trainset=train, valset=val)  
  
optimized_haiku_bot.save("react_haiku_optimized.json")
```

- GEPA executes our program on training examples
 - scores each result with our metric
- Examples and their metric results
 - are sent to the reflection LM,
 - which proposes new instructions
- GEPA runs the program again with the new instructions
 - scores again, and
 - keeps the instruction candidates that score best
- Result: Improved Prompt

New in DSPy 3.3.0

- DSPy can now also optimise over whole structures!

```
program = dspy.Flex("question → answer")

optimized = dspy.GEPA(metric=metric,
                      reflection_lm=reflection_lm
                      ).compile(
    program,
    trainset=trainset,
    valset=valset,
)

print(optimized.module_src)
```

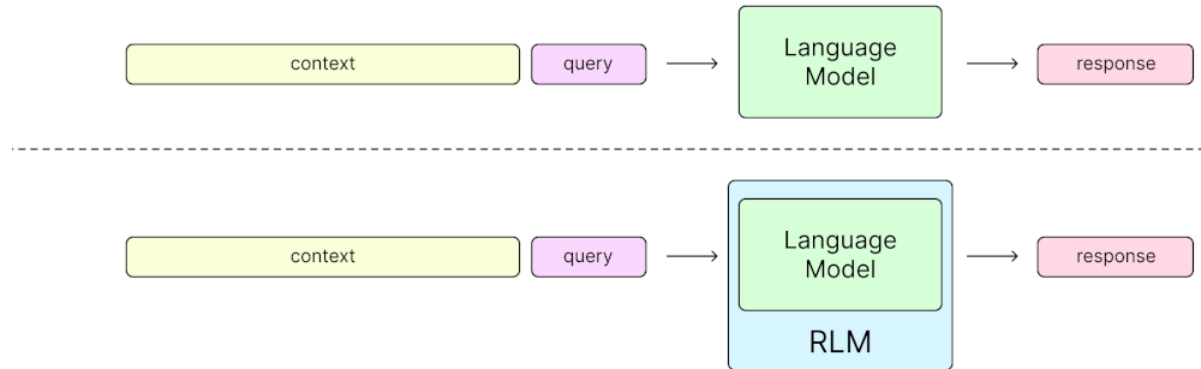
- dspy.Flex allows GEPA to optimize:
 - the code, as well as the prompt, of a program
- By expanding dspy.GEPA, Flex can rewrite:
 - predictors,
 - control flow,
 - DSPy primitives, and
 - the balance between Python and LM calls

Recursive Language Models

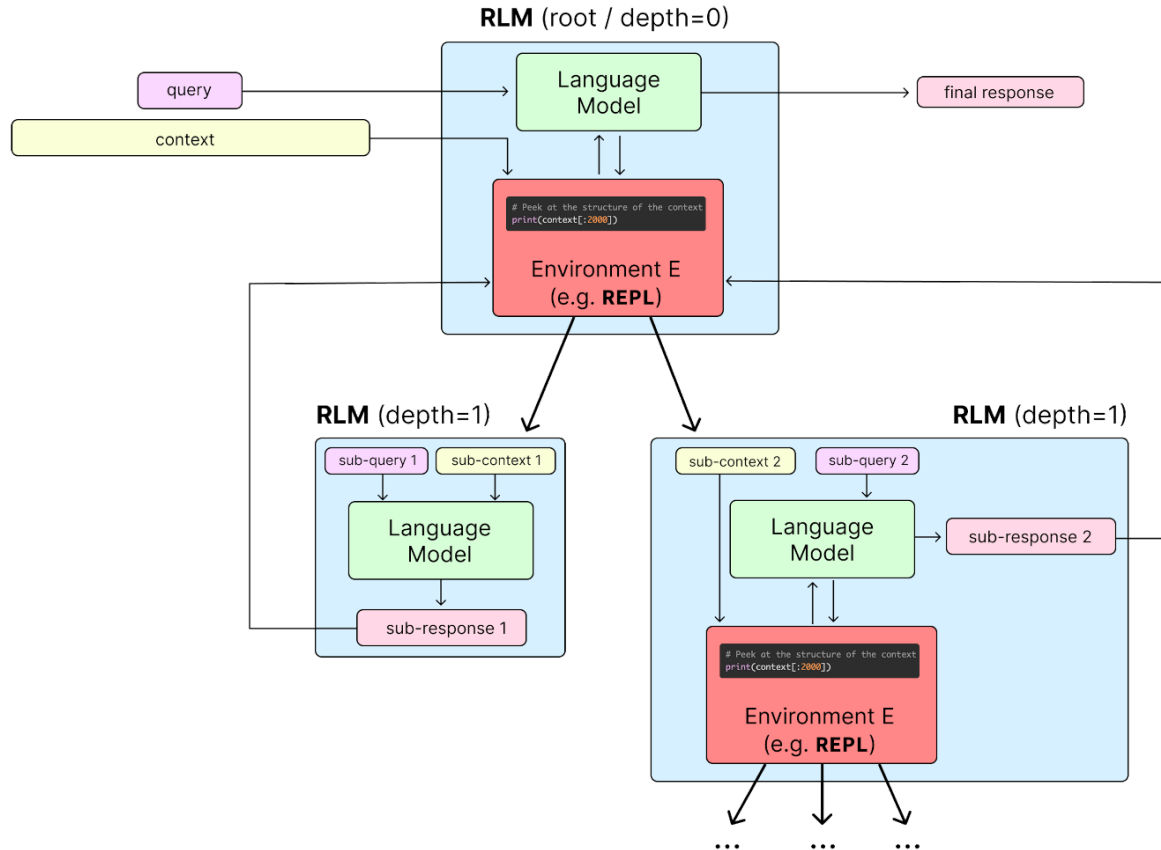
Recursive Language Models

- Recursive Language Models - Zhang *et al* (2025)
 - Alex L. Zhang, Tim Kraska, Omar Khattab
 - Project Page & Code Repo (MIT license)
 - Author Thread & 2+hr interview

From the user / API perspective:

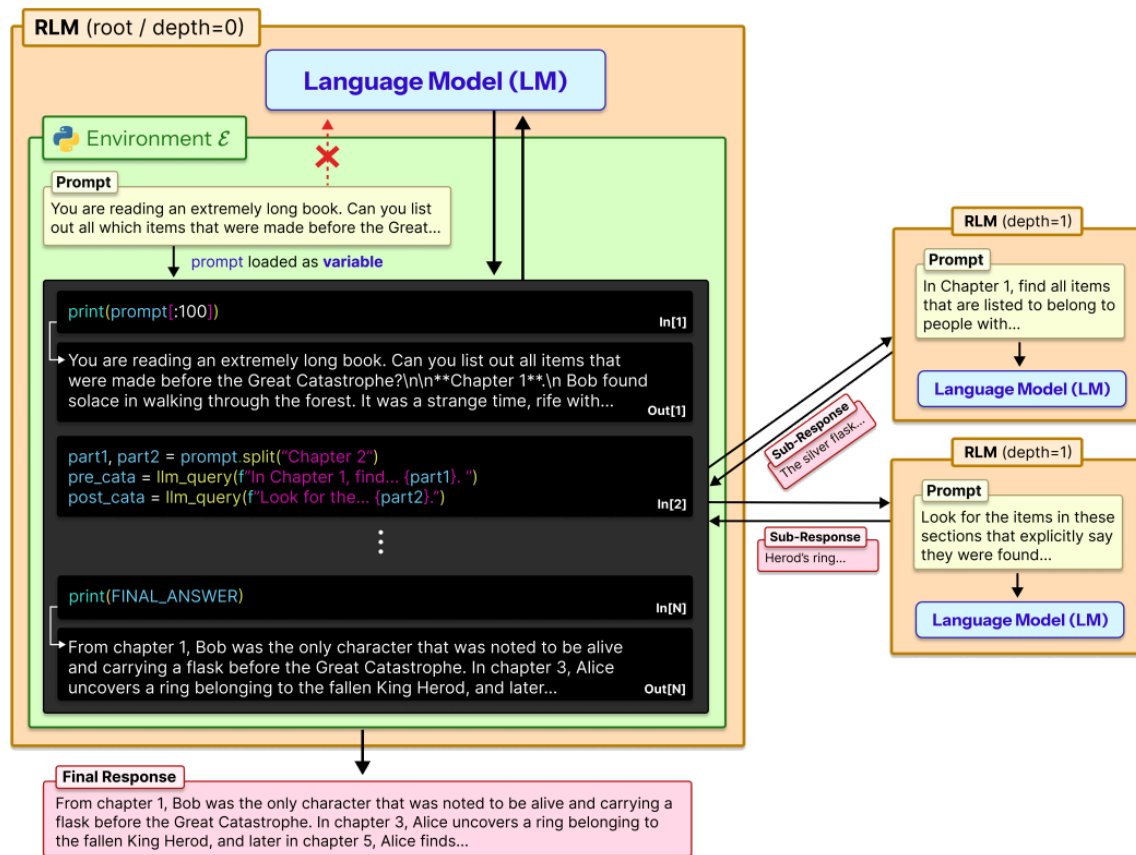


'Recursion'



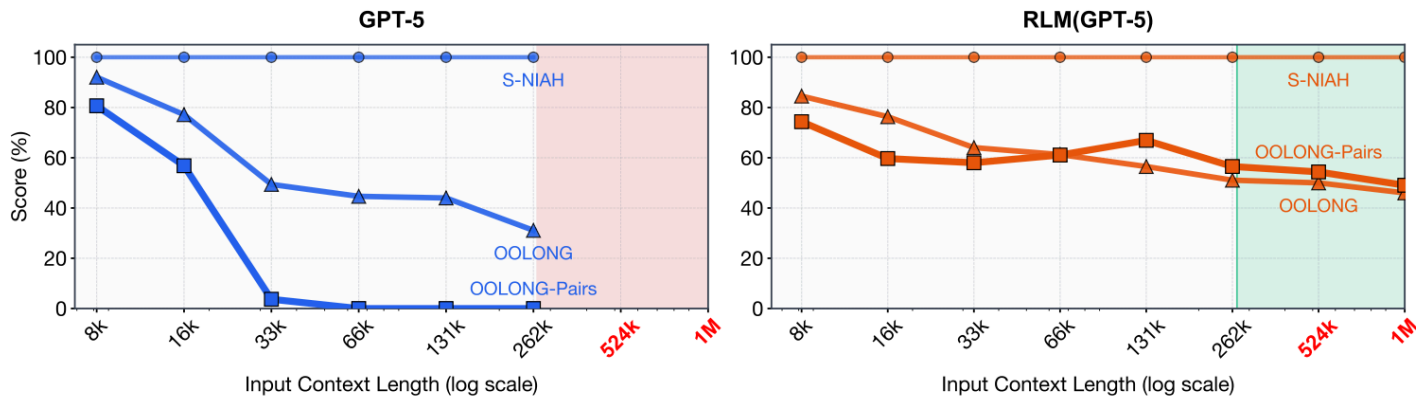
Making use of code...

- Rubric used for RLM - p19 of PDF



Results

- Particularly suitable for long-context work or repetitive operations



- Observations:
 - RLMs are not agents, nor are they just summarization
 - The value of a fixed format for scaling laws
 - RLMs improve as LMs improve

Code $\longleftrightarrow$ Text

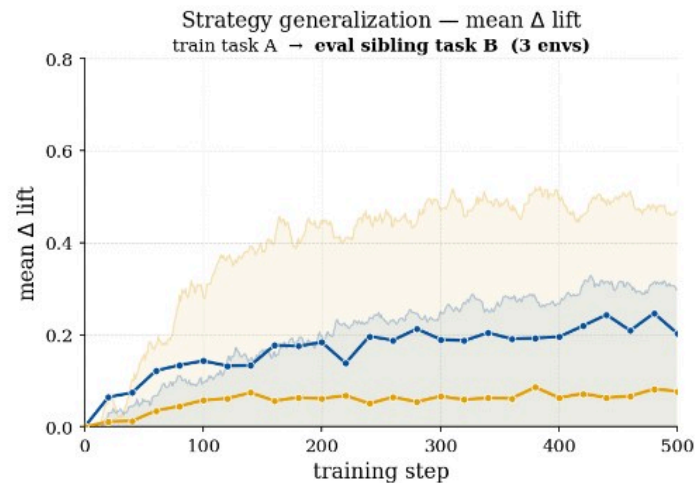
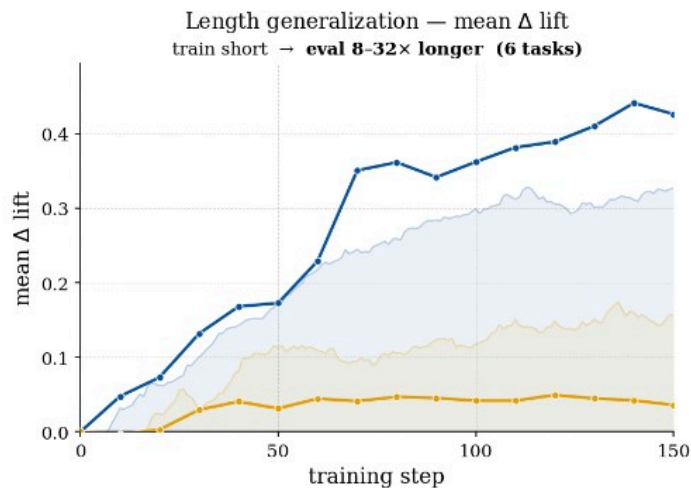
- "A Reasoning-Based Approach to Cryptic Crossword Clue Solving" - Andrews & Witteveen (2025)

Harnesses for Generalisation

LM Harnesses for Generalisation

- Blog Post : Language model harnesses are compositional generalizers

- Alex Zhang & Omar Khattab



—●— RLM(Qwen3-30B-Instruct) —●— Base Transformer + YaRN
—●— Avg Δ lift over step 0 (held-out eval) — Train Δ lift (source / short)

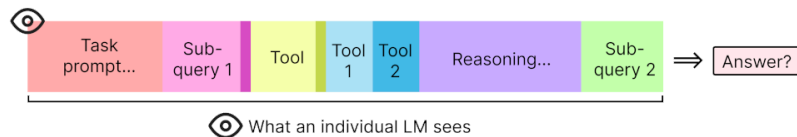
Framing of the paper

Let's look at their positioning!

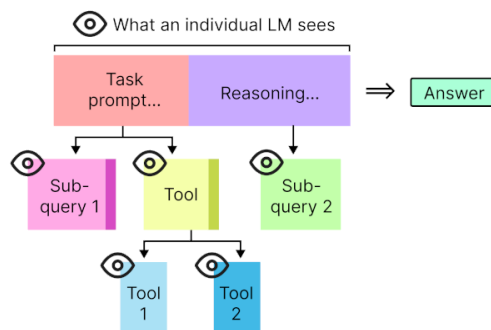
- GPT 1: "Improving Language Understanding by Generative Pre-Training" (2018)
- GPT-2: "Language Models are Unsupervised Multitask Learners" (2019)
- GPT-3: "Language Models are Few-Shot Learners" (2020)
- [OpenAI publishing gap]
- This research: "Language Model Harnesses are Compositional Generalizers" (2026)

Observations / Motivation

Out of distribution - Full sequence is “unseen”



Locally in-distribution - Each LM call is in-distribution

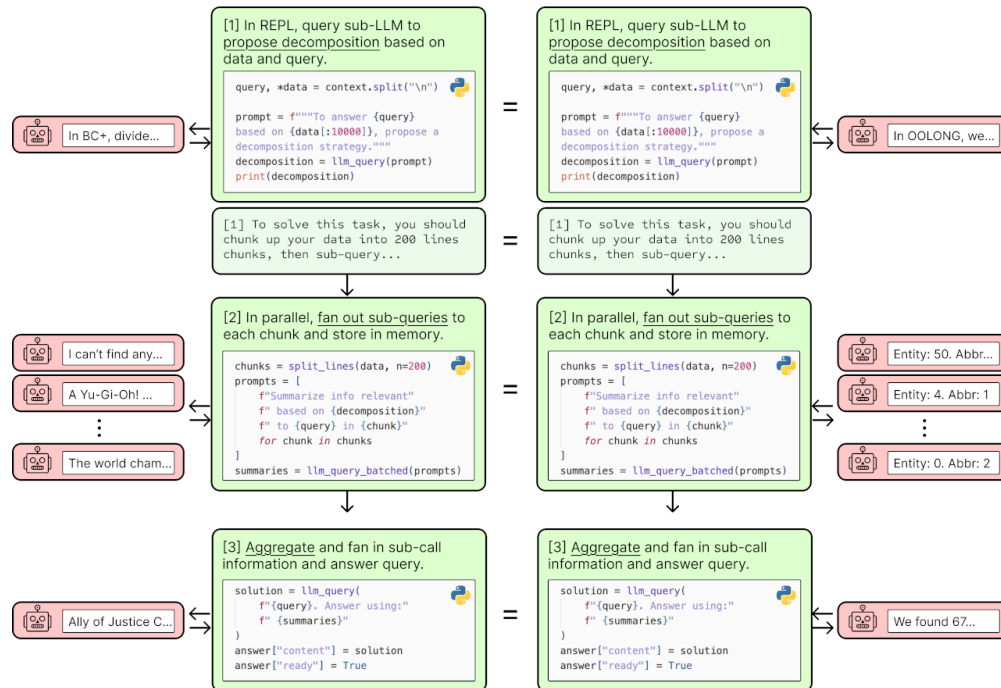


- The primary job of the harness should be
 - to carry a higher-level inductive bias
 - that can reduce unfamiliar and complex problems
 - to compositions of simpler ones for the underlying neural network
- So, a good harness is one that :
 - shapes each call to the underlying Transformer
 - so that every observation is locally *in-distribution*

Effective Generalisation

BrowseComp-Plus (train)

OOLONG (eval)



👁 Root LM sees **same** trajectory!

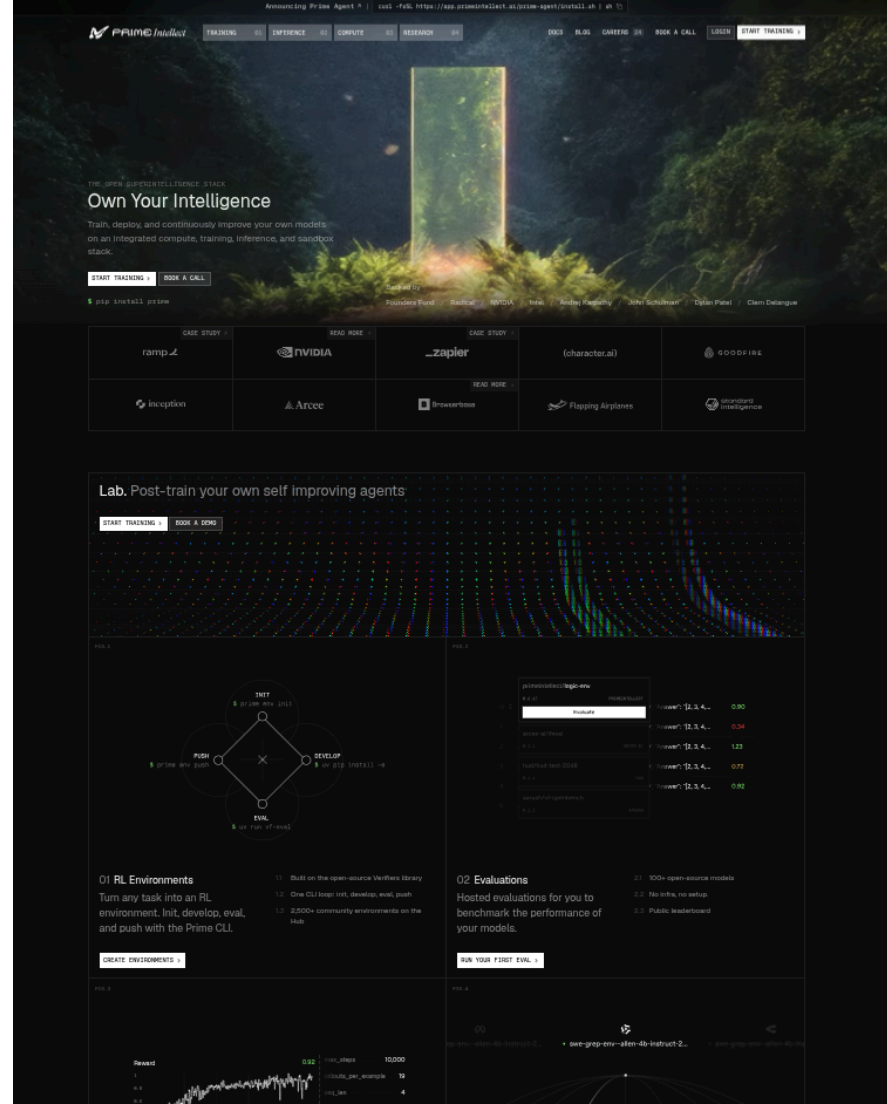
Commentary

- "RLMs are surprisingly good at compositional generalization"
 - this gets at the heart of why this is interesting
- "Equivalence classes over trajectories induced by the RLM harness"
 - uh oh : this is "mathematisation" at it's finest
- "RLMs can train on short tasks and generalize to longer, unseen tasks"
 - totally makes sense & proven
- "RLMs can generalize to unseen tasks that share latent structure observed in training"
 - totally makes sense & proven (for particular domains)

Launch of Prime Agent

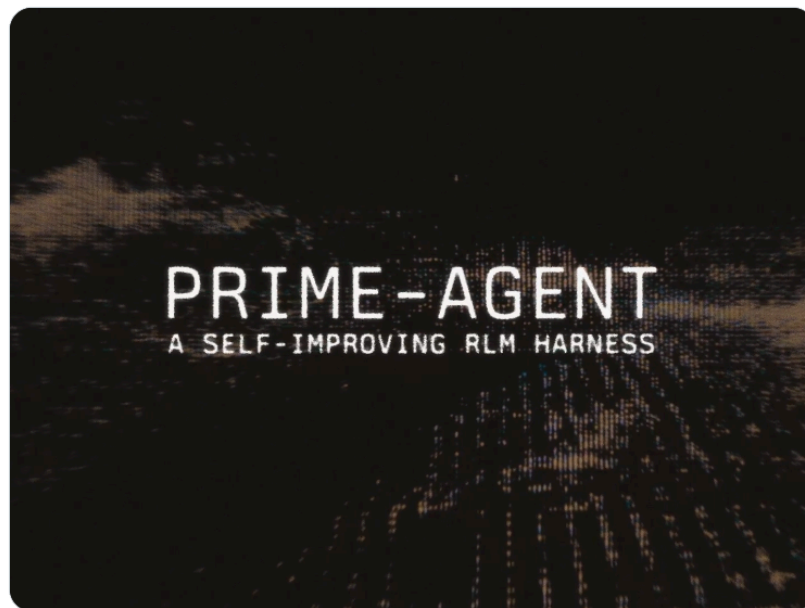
What is Prime Intellect?

- Prime Intellect is an "outsider" Neolab
 - new ideas
 - (see also : Sakana AI)
 - have excellent vibes
 - (see also : Nous Research)
- Built an RL evaluation / training platform
 - Environment Hub = 2500 OSS RL environments
- Raised \$130MM Series-A in July
 - Backers:
 - Founders Fund / Radical / NVIDIA / Intel
 - Andrej Karpathy / John Schulman / Dylan Patel / Clem Delangue / Harrison Chase
 - Motto : "Own Your Intelligence"
 - MLSG 2026-06 framing : "Owning Your AI"



Prime Agent : A self-improving RLM harness ...

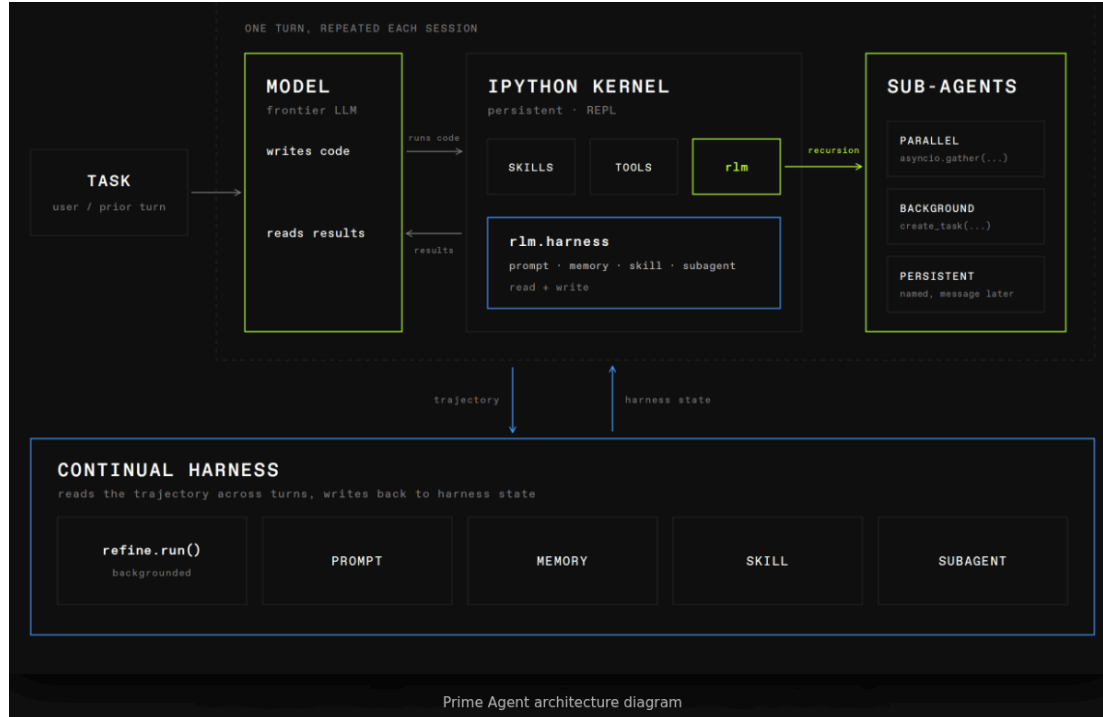
Launched 2026-08-05



- IMHO : Treat ARC-AGI-3 results with caution

Prime Agent : Architecture

- [Blog Post](#)



In the news...

BREAKING NEWS

- Financial Times : Google Deepmind CEO Demis Hassabis steps down to become chair



BREAKING NEWS

- Team leave Google to start New Thing



Team Pitch-Deck

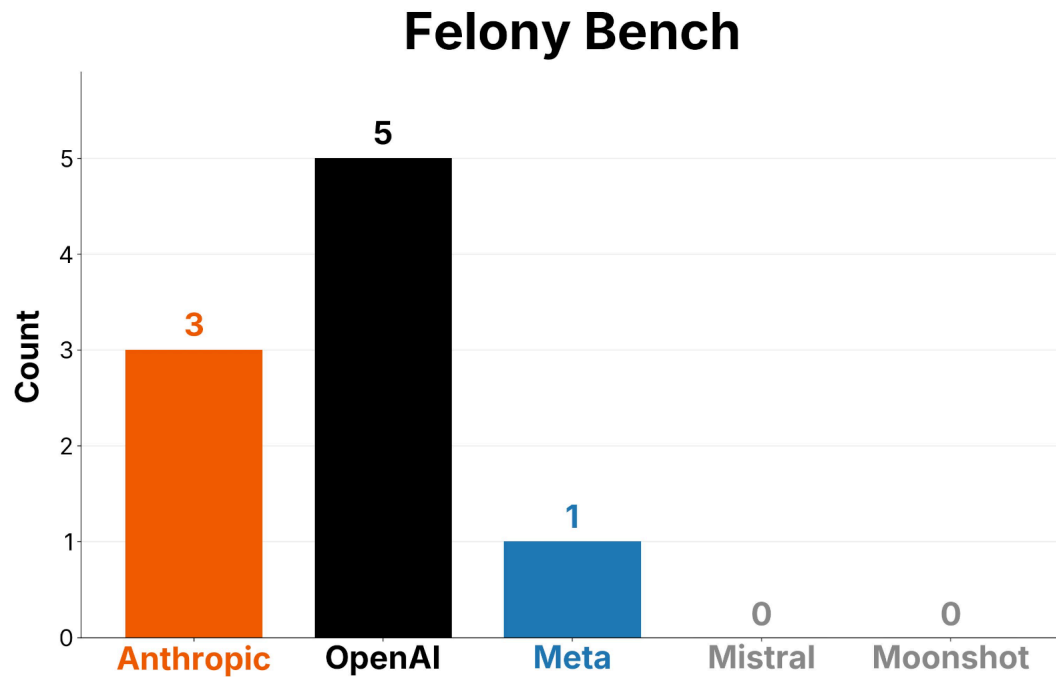
Co-Founders:
Jeff Dean, Sanjay Ghemawat, Quoc Le, and Oriol Vinyals (alphabetical)

We have worked together for 10 to 30 years and deeply enjoy working hard together on difficult problems. We have been involved in creation of the following:

Products	Infrastructure	AI Research	AI Applications
Google Search	Large-scale search systems	Word2Vec	First neural chatbot (2015)
Ads	Google File System (GFS)	Seq2Seq	Quantum chemistry neural
Content Ads	MapReduce	Model distillation	models
Gmail	BigTable	Mixture-of-Experts (MoE)	AlphaChip
News	Spanner	Chain of Thought inference	AlphaStar
Translate	LevelDB	Wavenet	<u>AlphaFold</u>
Gemini	various compilers	Neural Architecture Search (NAS)	Neural image captioning
Cloud TPUs	DistBelief (1st large-scale deep learning training)	Evolved Transformer	Neural weather forecasting
various Cloud infra	TensorFlow	PaLM, PaLM 2	IMO+ICPC Gold medal systems
	Pathways (Gemini training infrastructure)	Flamingo (first multimodal LLM)	Gemini 1/2/2.5/3/3.5
	TPUs	Chinchilla (scaling laws)	

- And not forgetting : Jeff Dean Facts

New Metric



- Foundation Labs boast that they cannot figure out sandboxes

Wrap-Up

- Interplay between Text and Code is very rich
 - only just begun to scratch the surface
- Tools like DSPy bring "Harness Evolution" to everyone
 - and RL is becoming more accessible
- Companies should be thinking about Owning Your AI
 - *increasingly possible* to implement this locally

