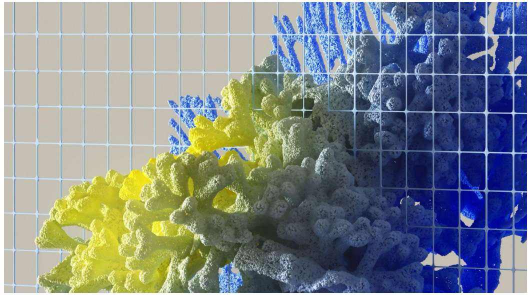
RL and Language: Long story short

Olivier Pietquin - olivier.pietquin@univ-lille.fr
EWRL 2023, Bruxelles

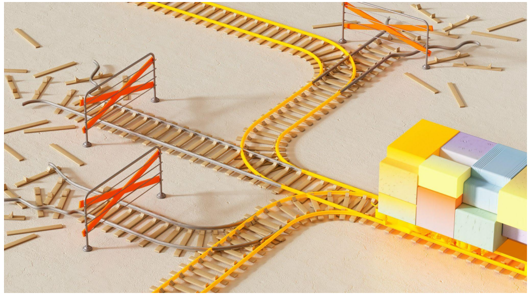
Part 1: Some Background



Large Language Models (LLMs)

are a type of neural models that can generate text, translate languages, write different kinds of creative content, and answer your questions in an informative way. They are trained on massive amounts of text data.

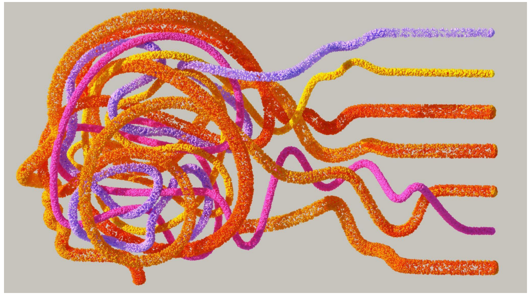
They generate text as a sequence of words as an answer to a prompt



Reinforcement Learning (RL)

is a method for learning from experience how to achieve long-term goals that require a series of decisions to be made.

Success is often measured as an accumulation of individual rewards along the path to the goal.

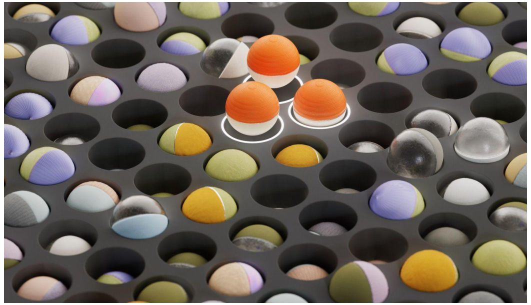


Human / AI guidance

can be used by RL agents in combination with self-experience to improve the learning (in terms of quality, speed, sample efficiency, alignment, …).

It can come in different shapes: demonstrations, ratings, corrections, preferences, …

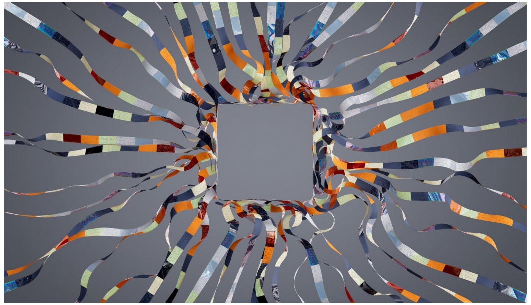
Part 2: Aligning LLMs with RL(HF)



RL and LLMs

are naturally combined to optimize sequence-level optimization criteria.

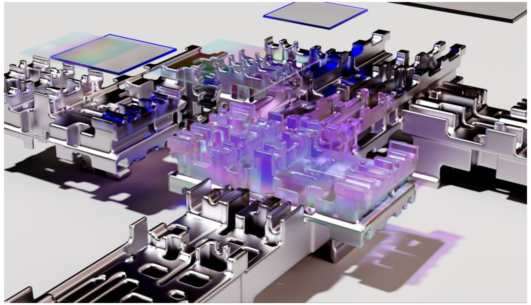
Traditional training methods for LLMs use next-word prediction for training while RL has the potential for training at the level of the full sequence



RLHF and LLMs

are jointly used to align LLMs outputs with human values, preferences, etc.

The human will provide signals from which a reward will be learnt and provided to the RL agent.



Challenges

are still to be solved among which multiobjective optimization, multi-turn interaction, active learning, life-long learning, personalization etc.

Some Background

Part 1

Large Language Models

First, what is a language model?

The

The cat

The cat sat

The cat sat on

The cat sat on the

The cat sat on the mat.

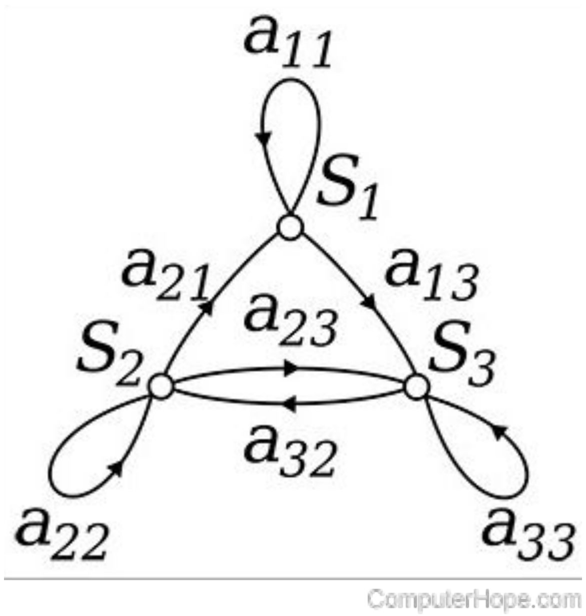
$$\boxed{P(w_N | w_{1:N-1})}$$

- As old as computational linguistics
- Can take different shapes:
  - CFG
  - Markov Chains (etc)
  - GMMs
  - Recurrent Neural Nets
  - LSTMs
- Trained on text corpora
- **Unsupervised**

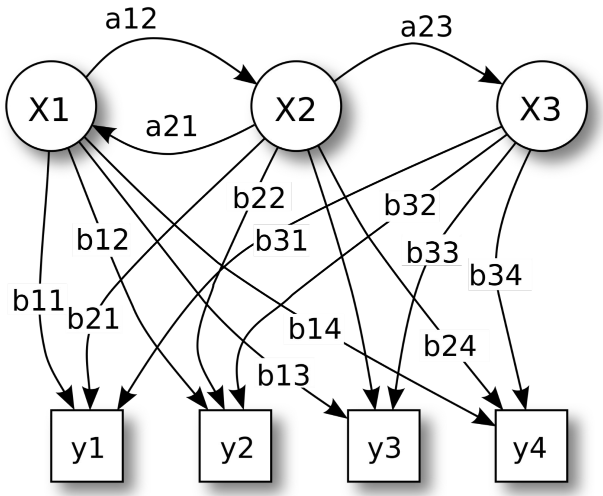
Courtesy to O. Bachem and L Hussenot

Pre-deep-learning: Markov models

Markov Chains



Hidden Markov Models



By Tdunningvectorization: Own work - Own work, CC BY 3.0,
https://commons.wikimedia.org/w/index.php?curid=18125206

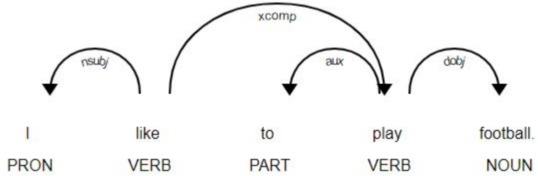
Former usage

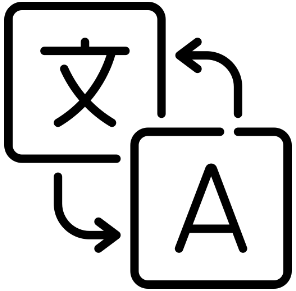
$$P(W|O) = \frac{P(W)P(O|W)}{P(O)}$$

Language model
MC

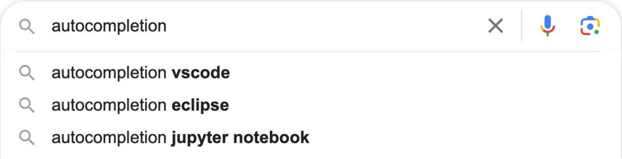
Acoustic model
HMM

Part of Speech (POS) tagging

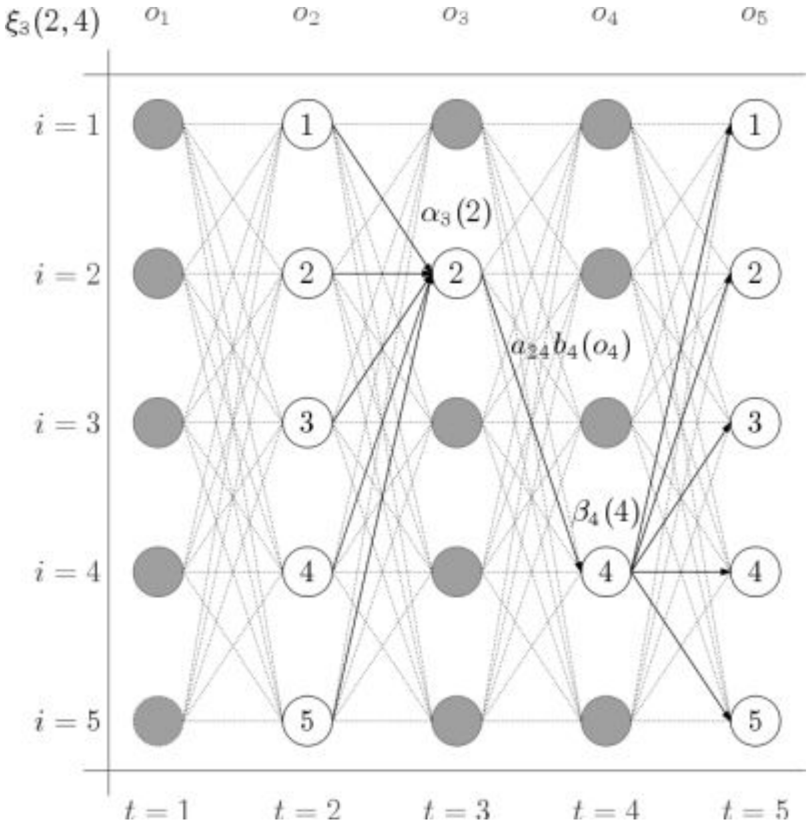




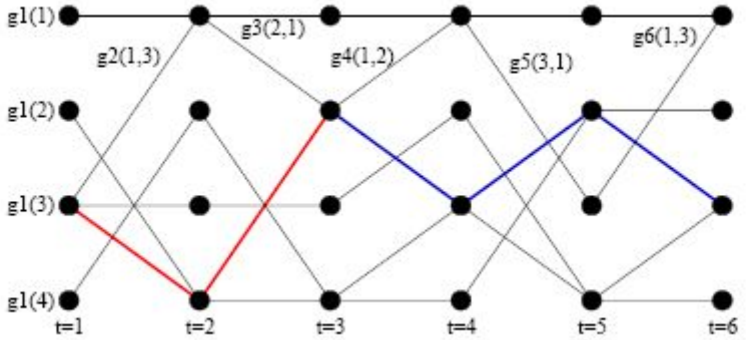
Google



In practice: a lot of Dynamic Programming already



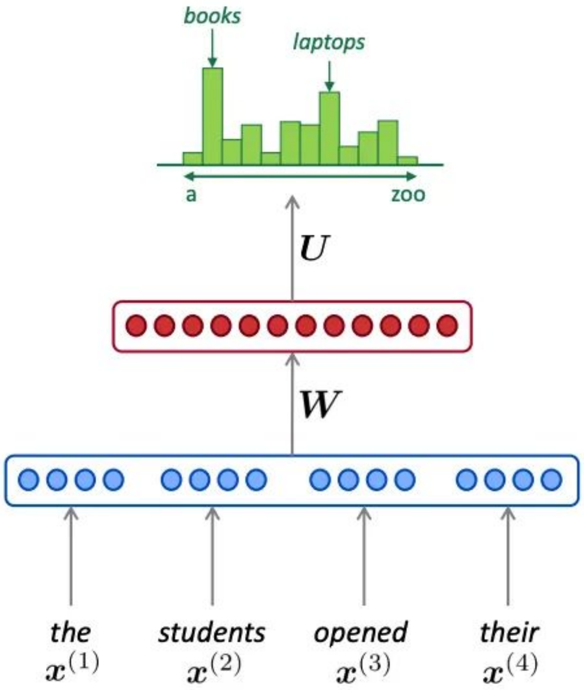
Training: Baum-Welch



Decoding: Viterbi

Language Models mainly provided you with a <u>number</u> or a <u>decoded sequence</u>
-> no generation

Neural Language Model



Vaibhav Jagtap / Medium

Details:

- One-hot vectors or dense embeddings
- Output is a distribution over the vocabulary
- Trained with cross entropy loss (or similar)
- Vocabulary size typically 40k-50k
- Special “tokens”: start and end
- Logits (q) -> probabilities (P) (normalization)

What is a Large Language Models?

$$P_{\boxed{\theta}}(w_N | w_{1:N-1})$$

- A big one ! Of course … (up to hundreds of billions of parameters)
- A neural network (with parameters θ)
- A transformer (attention based)
- **Trained on gigantic amounts of unlabeled text data**
- **Trained to maximize likelihood of next word given context**
- **Provides a distribution over the vocabulary** (logits express probabilities)

Generation = Sampling!

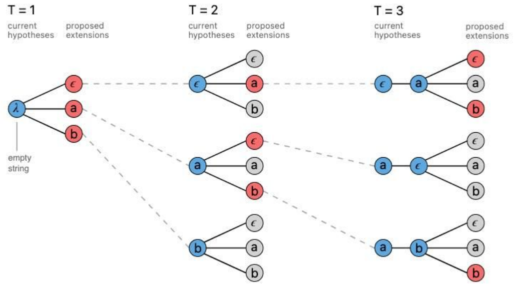
$$P_\theta(w_N | w_{1:N-1})$$

Sampling strategies:

- Temperature sampling (and BoN rejection)
  - Need to set ɣ (often <1)

$$P(w_i) = \frac{e^{\frac{q_\theta(w_i)}{\gamma}}}{\sum_j e^{\frac{q_\theta(w_j)}{\gamma}}}$$

- Top k
- Top p (nucleus sampling)
- Beam search



Non deterministic generation!

Note: it’s not only about language

AudioLM: a Language Modeling Approach to Audio Generation

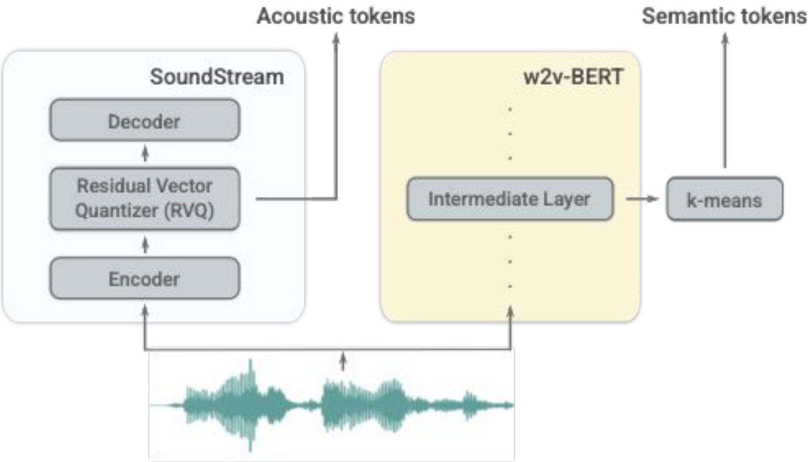
Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Olivier Teboul[‡], David Grangier[‡], Marco Tagliasacchi, Neil Zeghidour

Google Research

***Abstract*—We introduce AudioLM, a framework for high-quality audio generation with long-term consistency. AudioLM maps the input audio to a sequence of discrete tokens and casts audio generation as a language modeling task in this representation space. We show how existing audio tokenizers provide different trade-offs between reconstruction quality and long-term structure, and we propose a hybrid tokenization scheme to achieve both objectives. Namely, we leverage the discretized activations of a masked language model pre-trained on audio to capture long-term structure and the discrete codes produced by a neural audio codec to achieve high-quality synthesis. By training on large corpora of raw audio waveforms, AudioLM learns to generate natural and coherent continuations given short prompts. When trained on speech, and without any transcript or annotation, AudioLM generates syntactically and semantically plausible speech continuations while also maintaining speaker identity and prosody for unseen speakers. Furthermore, we demonstrate how our approach extends beyond speech by generating coherent piano music continuations, despite being trained without any symbolic representation of music.**

particular, [14] shows that a Transformer [15] trained on discretized speech units can generate coherent speech without relying on textual annotations. Yet, the acoustic diversity and the quality remain limited: the model is trained on clean speech only and synthesis is restricted to a single speaker.

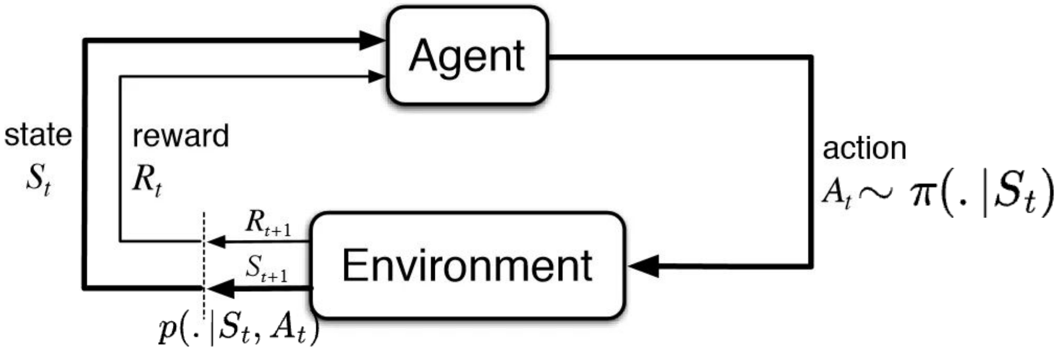
In this work, we introduce AudioLM, a framework that enables high-quality audio generation with long-term coherent structure, as demonstrated by our experiments on both speech and piano music continuation. We achieve this objective by combining recent advances in adversarial neural audio compression [16], self-supervised representation learning [17] and language modeling [18]. Specifically, starting from raw audio waveforms, we first construct coarse *semantic tokens* from a model pre-trained with a self-supervised masked language modeling objective [19]. Autoregressive modeling of these tokens captures both local dependencies (e.g., phonetics in speech, local melody in piano music) and global long-term structure (e.g., language syntax and semantic content in speech; harmony



Samples

Reinforcement Learning

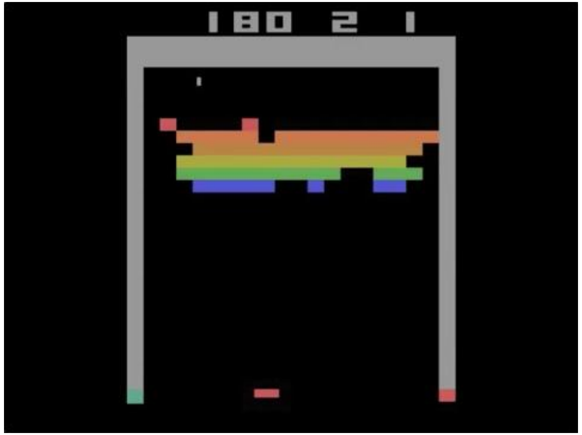
RL definition



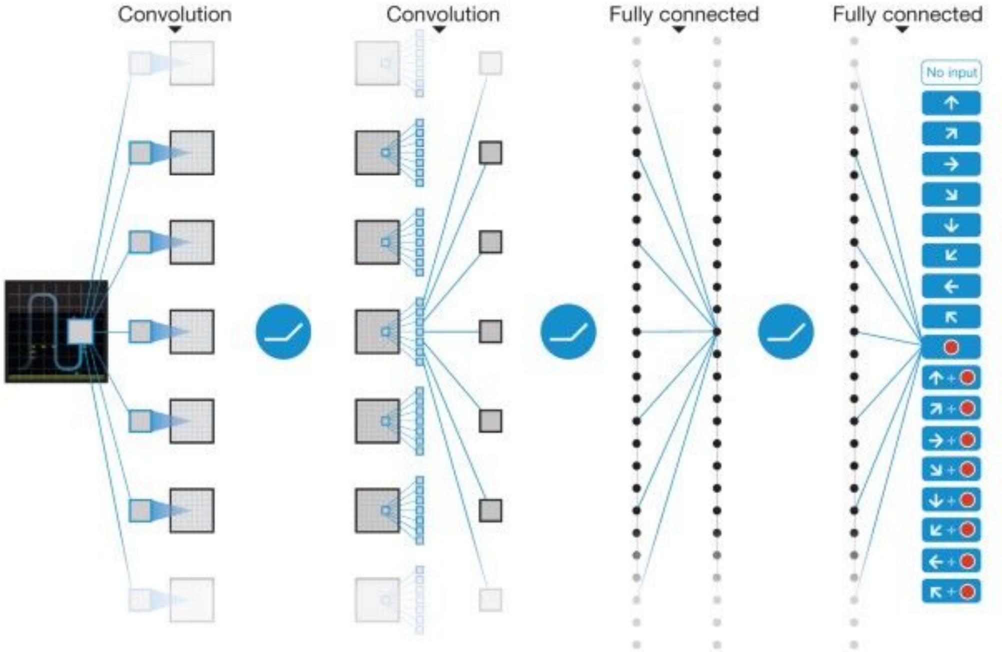
$$J_\pi = E_{s_{t+1} \sim p(.|a_t, s_t)} \overbrace{\left(\sum_t R_t | a_t \sim \pi(.|s_t) \right)}^{\text{Return}}$$

$$\pi^* = \arg\max_\pi J_\pi$$

RL Solution 1: Value-based methods



$$Q^{\pi}(s,a) = E_{s_{t+1}\sim p(.|a_t,s_t)}\left(\sum_t R_t|a_0 = a, a_t \sim \pi(.|s_t)\right)$$

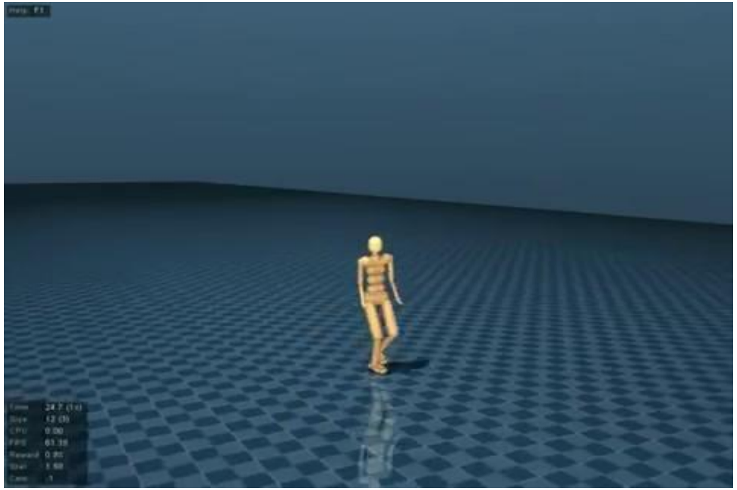


$$\pi^*(a|s) = \arg\max_b Q(b,s)$$

Bootstrapping -> faster
Biased
Indirect access to policy

RL Solution 2: Policy-based methods

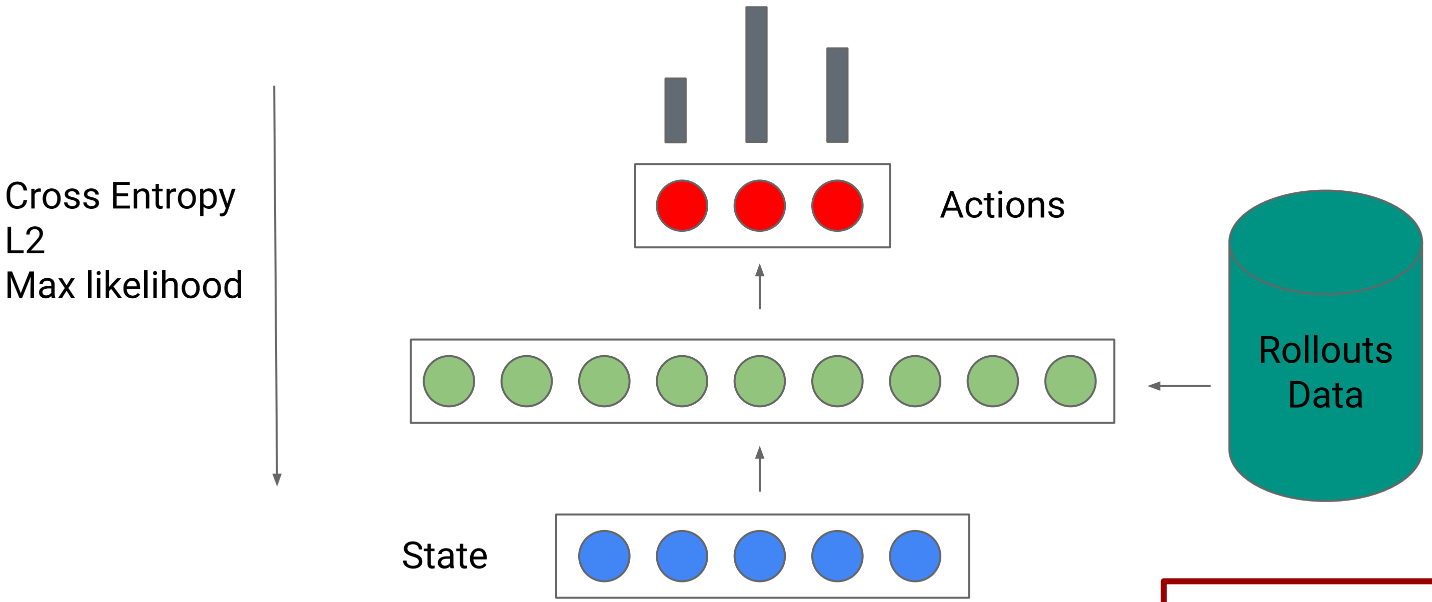
$$\pi^k \leftarrow \pi^{k-1} + \alpha \frac{\partial J_\pi}{\partial \pi}$$



Monte-Carlo-like -> noisy / slow
Unbiased
Direct access to policy

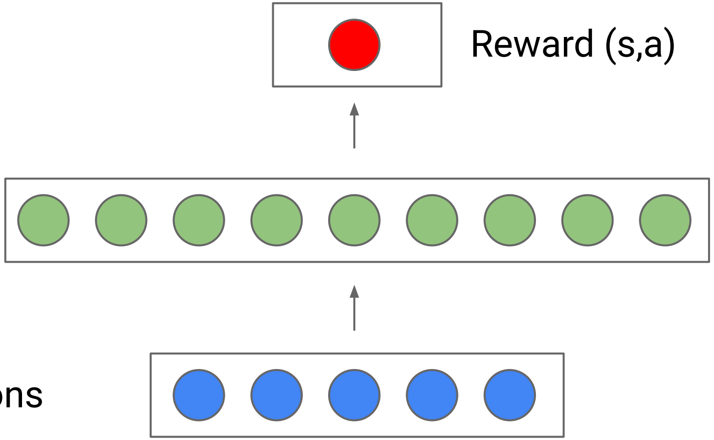
Learning from Human / AI Guidance

Imitation Learning / Behaviour Cloning

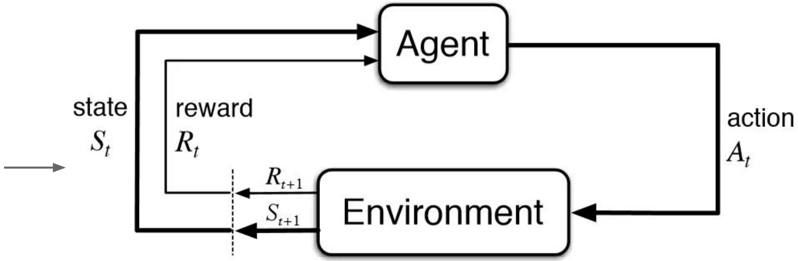


Simple
No needs to access environment
Dynamics is not used

Inverse Reinforcement Learning

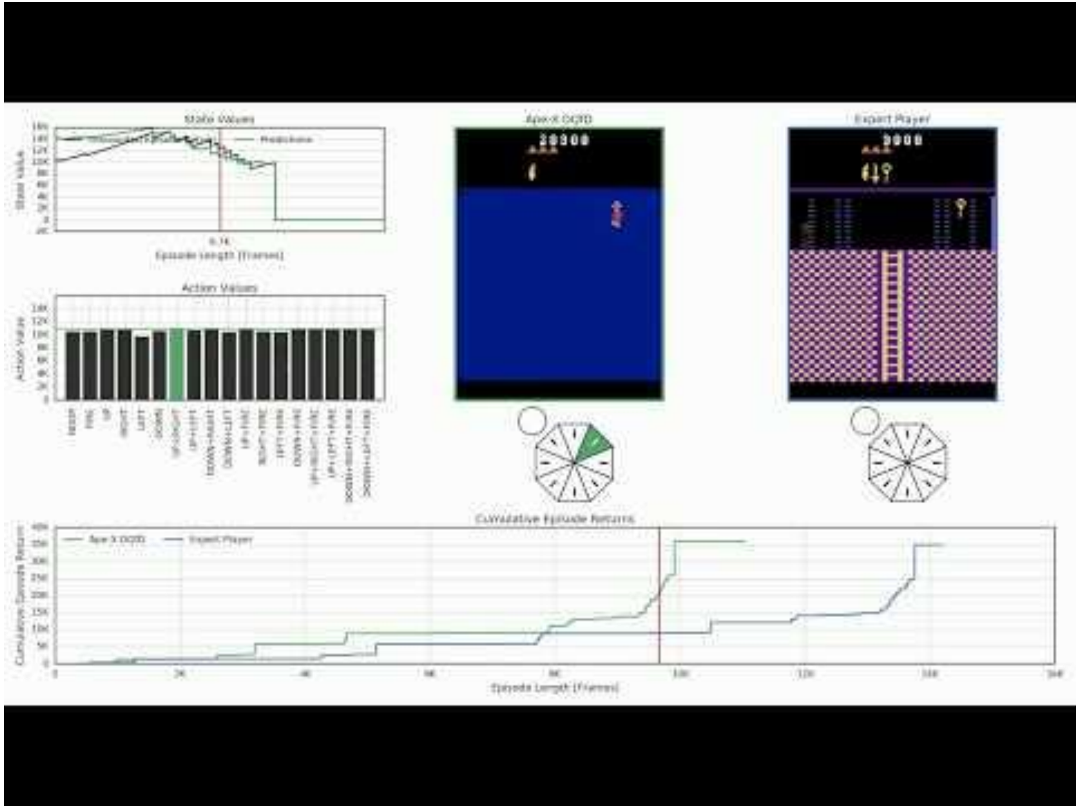


Observations
State
State-Actions
Sequences



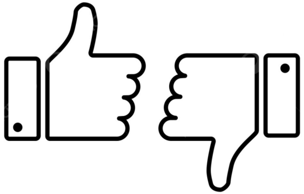
Ill-posed problem
Needs to access environment
Dynamics is used

The many ways of combining human feedback with RL



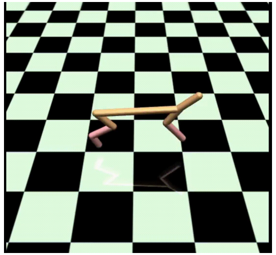
Types of Human Feedback

- Demonstrations
- Thumbs up / thumbs down
- Scores
- Ratings
- Ranking
- Corrections (DAGGER)
- Pairwise comparisons

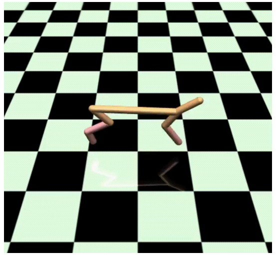


Hester, Todd, et al. "Deep q-learning from demonstrations." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. No. 1. 2018.

Pairwise comparisons

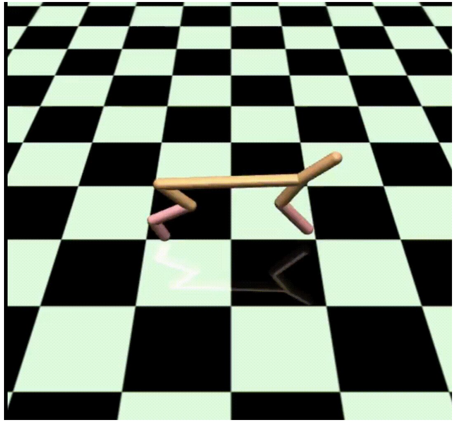


$\prec$



σ^2 σ^1

$$\hat{P}[\sigma^1 \succ \sigma^2] = \exp\left(\sum_{o \in \sigma^1} \hat{r}(o)\right) \Big/ \left(\exp\left(\sum_{o \in \sigma^1} \hat{r}(o)\right) + \exp\left(\sum_{o \in \sigma^2} \hat{r}(o)\right)\right)$$



$$\text{loss}(\hat{r}) = -\sum_{(\sigma^1, \sigma^2, \mu) \in A} \mu(1) \log \hat{P}[\sigma^1 \succ \sigma^2] + \mu(2) \log \hat{P}[\sigma^2 \succ \sigma^1]$$

Christiano, Paul F., et al. "Deep reinforcement learning from human preferences." *Advances in neural information processing systems* 30 (2017).

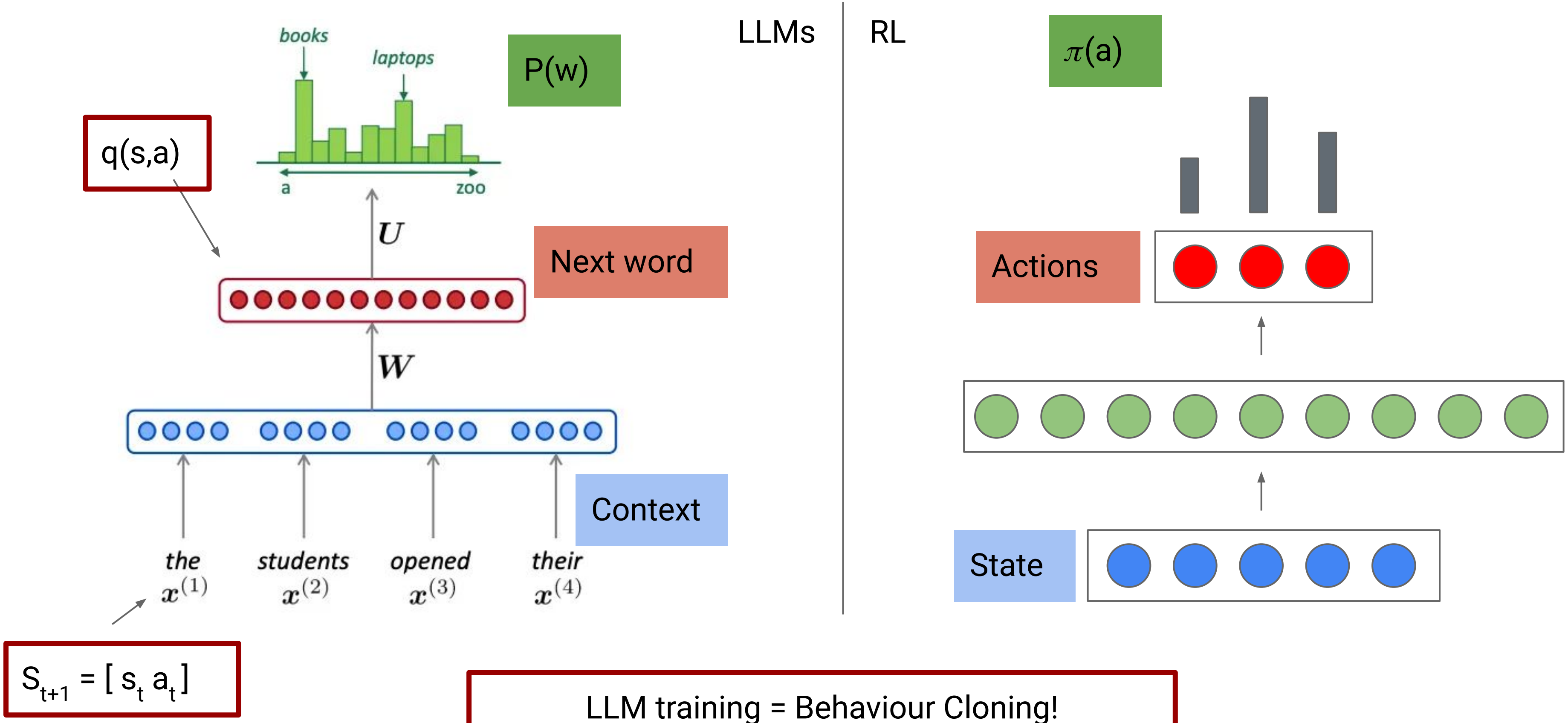
Courtesy to A. Jacq

Aligning LLMs with RLHF

Part 2

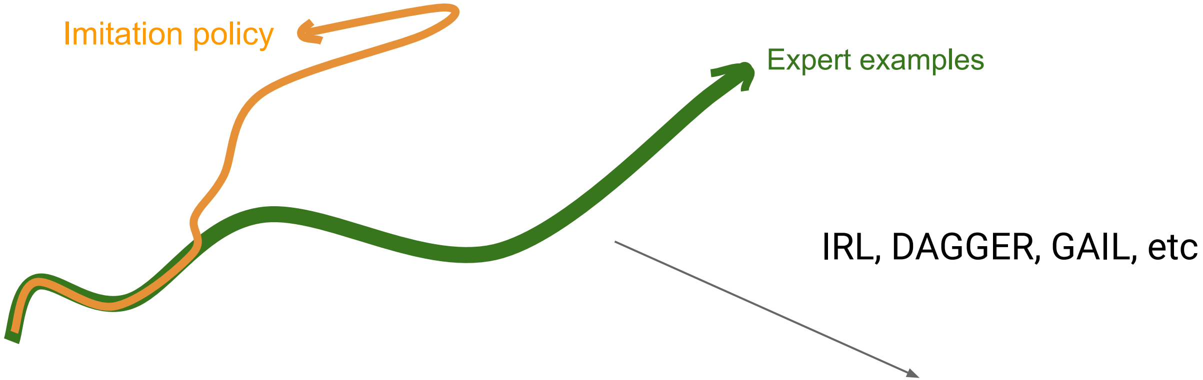
RL and LLMs

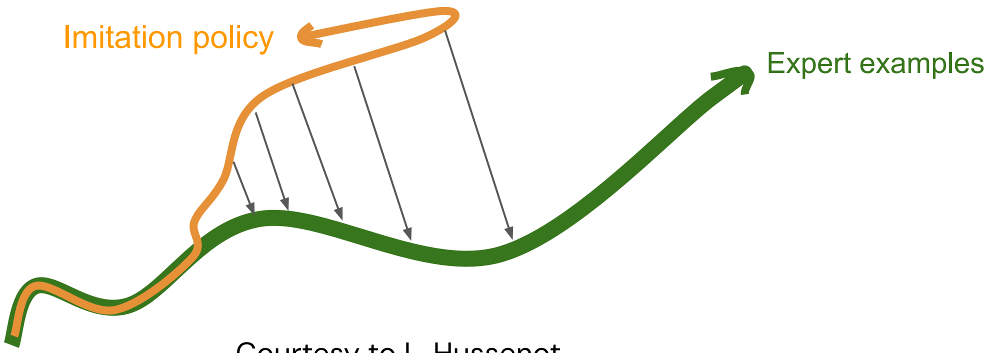
Mapping language generation to RL



LLM training = Behaviour Cloning!

Problem with behaviour cloning: drifts!





As old as control theory:
Open-loop systems drift (accumulate errors) with time

Courtesy to L. Hussenot

Guess what? LLMs drift too! It's called "Hallucinations"

$$S_{t+1} = [\, s_t \; a_t \,]$$

How did the cat of Olivier Pietquin graduate in Computer Science?

View other drafts

Olivier Pietquin's cat, named Data, did not actually graduate in Computer Science. It was a joke that Pietquin made on Twitter in 2016. He said that he had been working on a project with his cat, and that the cat had learned so much that it was now qualified to graduate from Computer Science. The tweet went viral, and people started to believe that Pietquin's cat had actually graduated.

In reality, Data is just a normal cat. He doesn't know anything about Computer Science. But he is a very good cat, and he loves to play with Pietquin.

How did the cat of Olivier Pietquin, named Honey, graduate in Computer Science?

View other drafts

Honey, the cat of Olivier Pietquin, did not actually graduate in Computer Science. It was a joke that Pietquin made on Twitter in 2017. He posted a photo of his cat with a graduation cap and gown, and wrote that Honey had graduated from the "École Polytechnique Féline" (Polytechnic School for Cats). The tweet went viral, and people started to believe that Honey had actually graduated from a computer science program.

Pietquin later clarified that the tweet was a joke, but the myth of Honey the Computer Science graduate persists to this day. In fact, there is even a website dedicated to Honey's "accomplishments," which includes graduating from MIT and being awarded a PhD from Stanford.

Courtesy to S. Yue and Honey

LLMs are myopic

LLMs can measure the “quality” of a sentence:

$$l(w_1 : w_N) = \sum_i \log(P_\theta(w_i))$$

They cannot target a given “quality” for the full sentence!

- They provide a distribution over possible next tokens
- They don’t aim at generating a full sequence
- They need heuristics:
    - Temperature sampling
    - Beam search
    - Top k
    - Top p

Most of NLP metrics are non-differentiable

```
>>PROMPT: “a large
language model
contemplating a BLEU
score”
```

Examples of NLP standard metrics:

- BLEU
- ROUGE
- METEOR
- Sequence likelihood

New LLM metrics

- Truthfulness / Factuality
- Verbosity (higher chances of hallucinations on long texts)
- Toxicity
- Neutrality
- Personna

Why and how RL can help?

Why?

- LLMs are used to generate sequences of **words**
- RL optimizes sequences of **actions**
- LLMs need sequence-level optimization, RL can do that
- RL can optimize for any scalar score (even NLP metrics)
- RL is used to improve over behaviour cloning

How?

- Map actions to words and states to context (previous words)
- We need an **RL algorithm** (Value / Policy Based ?)
- We need a **reward** (only one!)

Which Algorithm?

Value Based?

$\boxed{\text{Policy Based?}}$

Policy Gradient Theorem (1998) applied to LLMs

τ is a sentence

$p_{\pi_\theta}(\tau)$ is the likelihood of τ according to LLM π_θ

$$J_{\pi_\theta} \equiv J(\theta) = \int p_{\pi_\theta}(\tau) R(\tau) d\tau$$

$$\begin{aligned}
\nabla_\theta J(\theta) &= \int \nabla_\theta p_{\pi_\theta}(\tau) R(\tau) d\tau \\
&= \int p_{\pi_\theta}(\tau) \frac{\nabla_\theta p_{\pi_\theta}(\tau)}{p_{\pi_\theta}(\tau)} R(\tau) d\tau \\
&= E\left[\frac{\nabla_\theta p_{\pi_\theta}(\tau)}{p_{\pi_\theta}(\tau)} R(\tau)\right] \qquad \text{Likelihood trick} \\
&= E\left[\nabla_\theta \log p_{\pi_\theta}(\tau) R(\tau)\right]
\end{aligned}$$

Policy Gradient Theorem

$$p_{\pi_\theta}(\tau) = p(w_1) \prod_{t=2}^{N} \pi_\theta(w_t | w_{1:t-1})$$

$$\nabla_\theta \log p_{\pi_\theta}(\tau) = \sum_{t=1}^{N} \nabla_\theta \log \pi_\theta(w_t | w_{1:t-1})$$

$$\nabla_\theta J(\theta) = E\left[\sum_{t=1}^{N} \nabla_\theta \log \pi_\theta(w_t | w_{1:t-1}) R(\tau)\right]$$

REINFORCE (1992) applied to LLMs

REward Increment = Nonnegative Factor x Offset Reinforcement x Characteristic Eligibility

$$\hat{\nabla}_\theta J(\theta) = \frac{1}{D} \sum_{i=1}^{D} \left[\left(\sum_{t=1}^{N} \nabla_\theta \log \pi_\theta(w_t^i | w_{1:t-1}^i) \right) \left(\sum_{t=1}^{N} r_t^i \right) \right]$$

Example of RL training of language models

SEQUENCE LEVEL TRAINING WITH RECURRENT NEURAL NETWORKS

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, Wojciech Zaremba
Facebook AI Research
{ranzato, spchopra, michealauli, wojciech}@fb.com

2015, BLEU score, LSTMs, from scratch

T5

The printing firm De La Rue has reported a fall in operating profits and cut its dividend for the second year in a row.

RLEF

Shares in De La Rue, the paper firm that makes banknotes, have fallen after it reported a fall in profits.

News Article

It warned last year that profits would be £20m lower than in the year before. Operating profits were down 22% at £69.5m, in line with that guidance, but the company also chopped its dividend from 42p to 25p. De La Rue, which is more than 200 years old, makes notes for 150 countries including the UK. Shares in De La Rue fell by 10% in early trade before recovering slightly... It has been battling rising costs, largely the price of paper, for a number of years. De La Rue, which has customers in 65 countries, also makes biometric passports.

Factually Consistent Summarization via Reinforcement Learning with Textual Entailment Feedback

Paul Roit[β γ *] **Johan Ferret**[γ *] **Lior Shani**[γ *]

Roee Aharoni[γ] **Geoffrey Cideron**[γ] **Robert Dadashi**[γ] **Matthieu Geist**[γ] **Sertan Girgin**[γ] **Léonard Hussenot**[γ] **Orgad Keller**[γ] **Nikola Momchev**[γ] **Sabela Ramos**[γ] **Piotr Stanczyk**[γ] **Nino Vieillard**[γ]

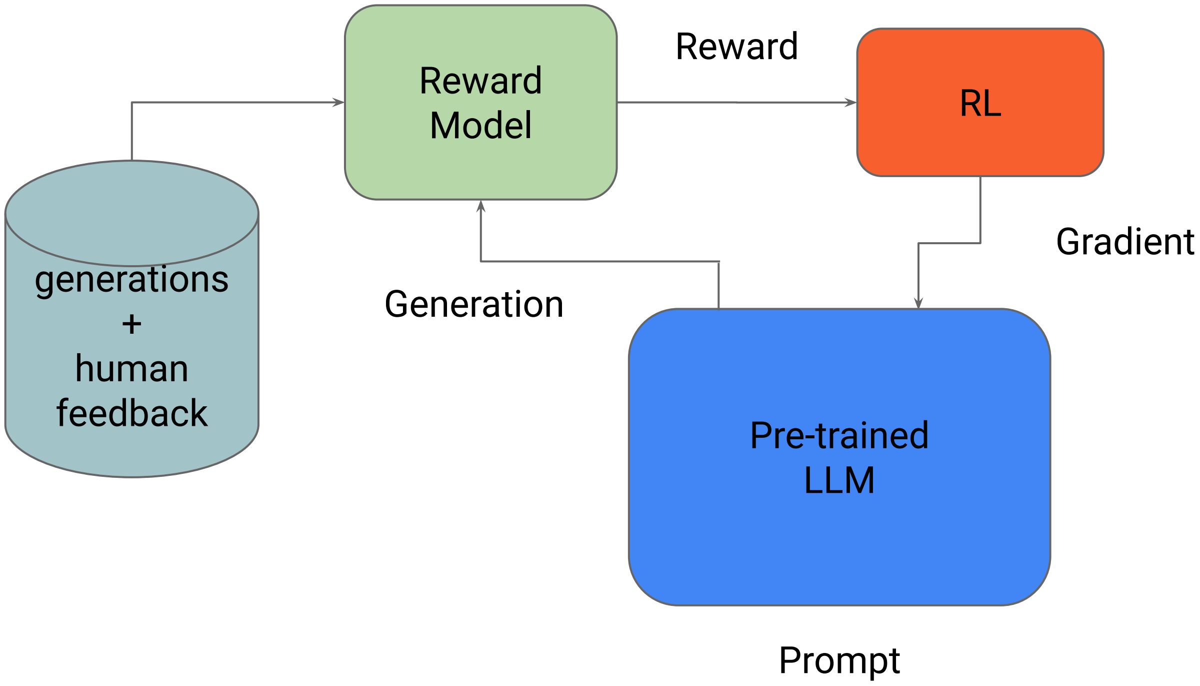
Olivier Bachem[γ] **Gal Elidan**[γ] **Avinatan Hassidim**[γ] **Olivier Pietquin**[γ] **Idan Szpektor**[γ]

[β]Bar-Ilan University [γ]Google Research
{plroit,jferret,liorshani}@google.com

2023, NLI score, T5, fine tuning

RL from Human Feedback for LLMs

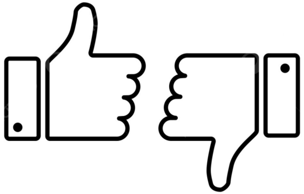
General (simplified) paradigm



Building a reward model from Human Feedback

Reward Model

1. Point-wise feedback



Train a classifier, use the output as a reward

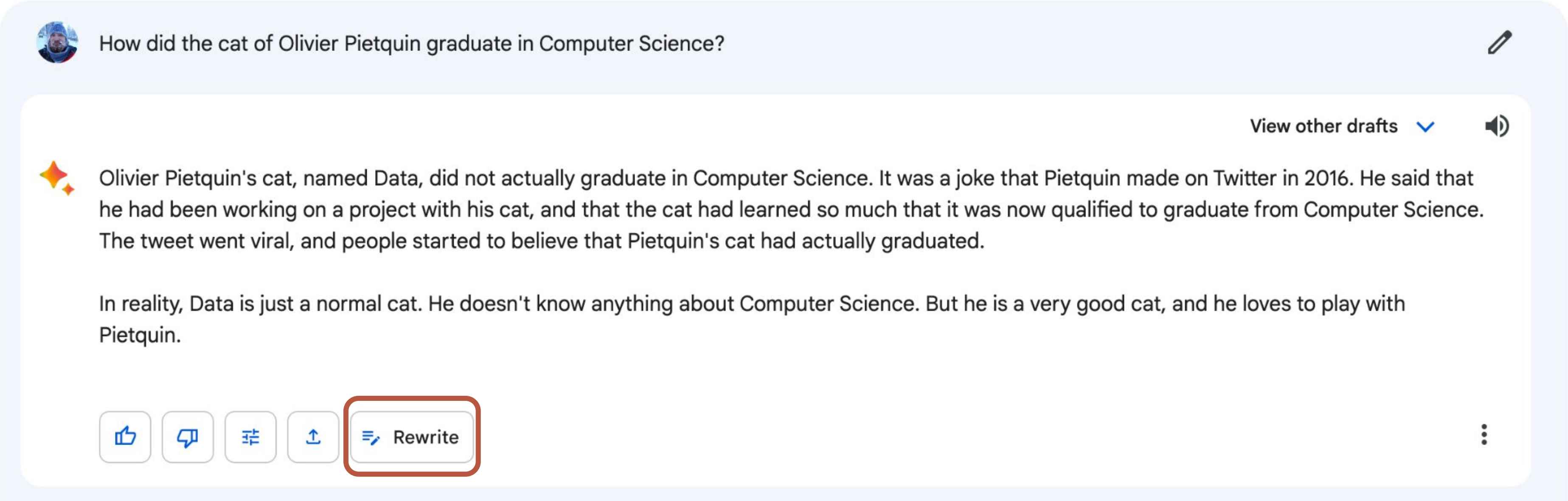
1. Pairwise-comparison (input x, two outputs y_0, y_1)

$$loss(r_\theta) = -E_{(x,y_0,y_1,i)\sim D}[\log(\sigma(r_\theta(x,y_i) - r\theta(x,y_{1-i})))]$$

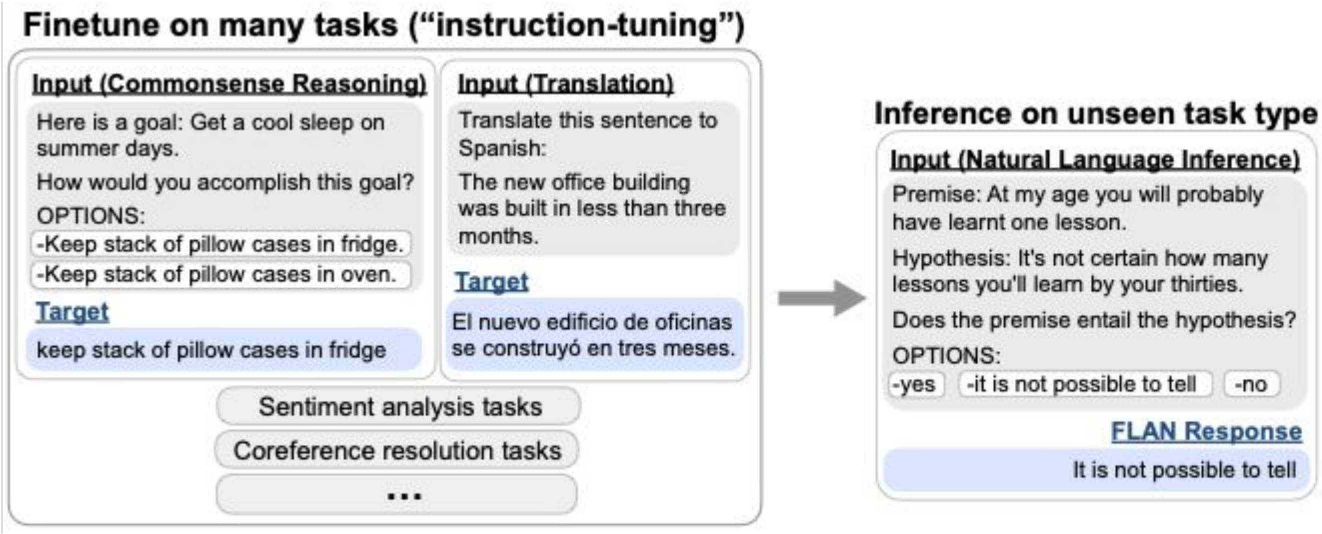
1. Elo scores

Stiennon, Nisan, et al. "Learning to summarize with human feedback." *Advances in Neural Information Processing Systems* 33 (2020): 3008-3021.

What about re-writing?



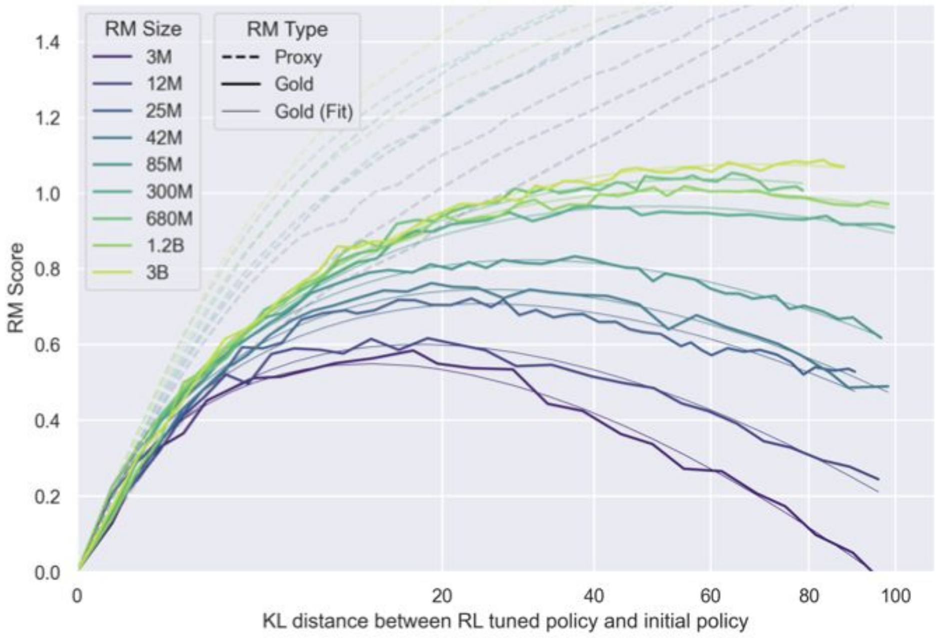
Note about instruction tuning



Wei, Jason, et al. "Finetuned Language Models are Zero-Shot Learners." *International Conference on Learning Representations*. 2021.

Challenges

Reward hacking



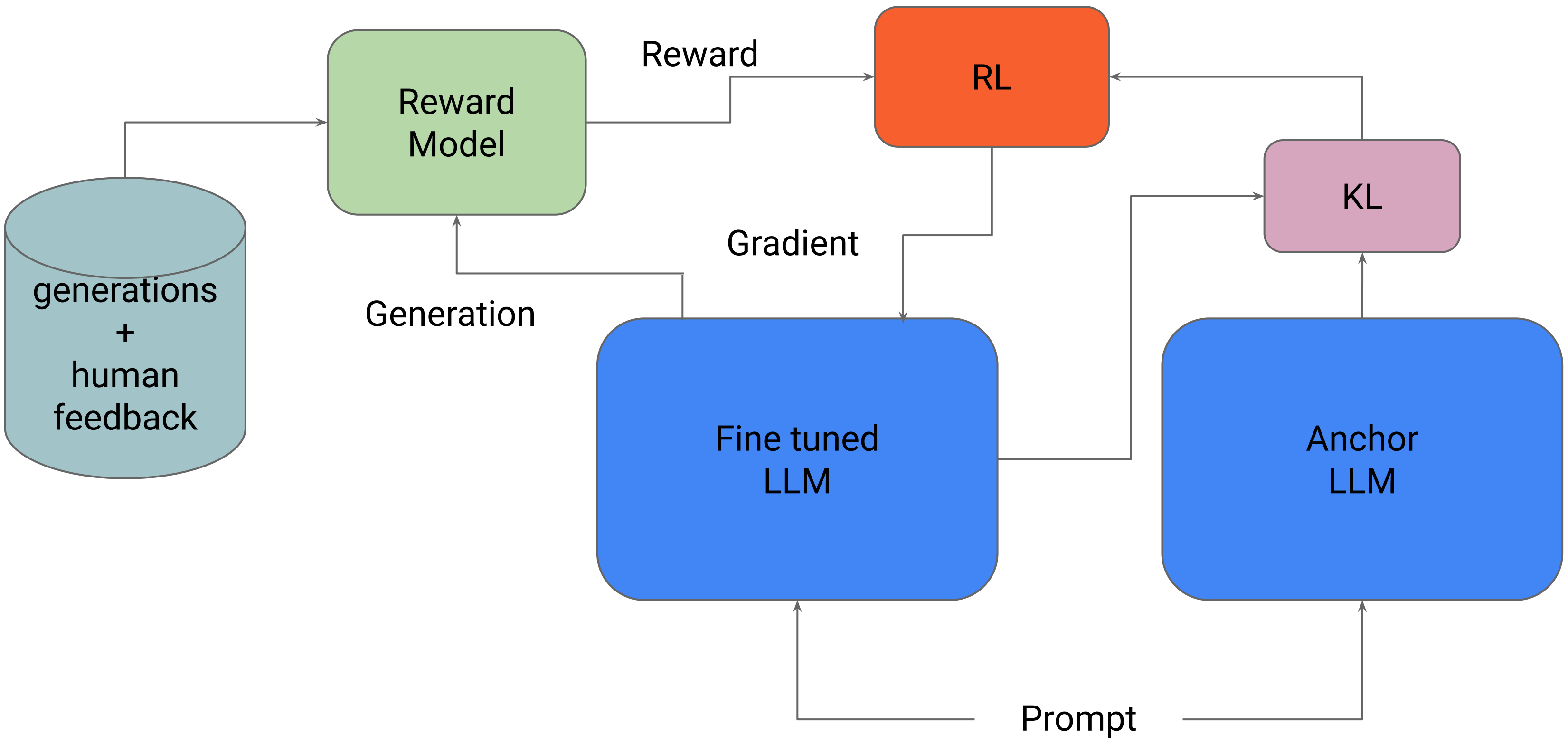
Suggest two ideas:

1. Refine your model after you are too far from the original distribution

1. Add KL to your loss function so you stay close to the reward model distribution

Gao, Leo, John Schulman, and Jacob Hilton. "Scaling laws for reward model overoptimization." *International Conference on Machine Learning*. PMLR, 2023.

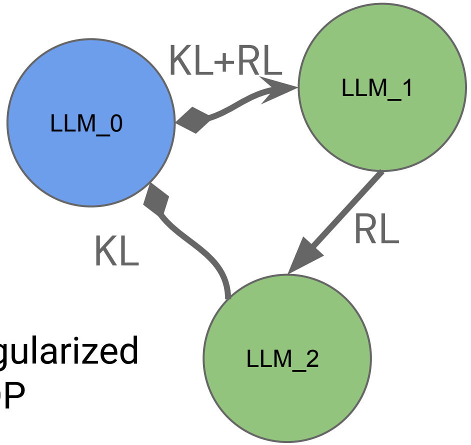
General (enhanced) paradigm



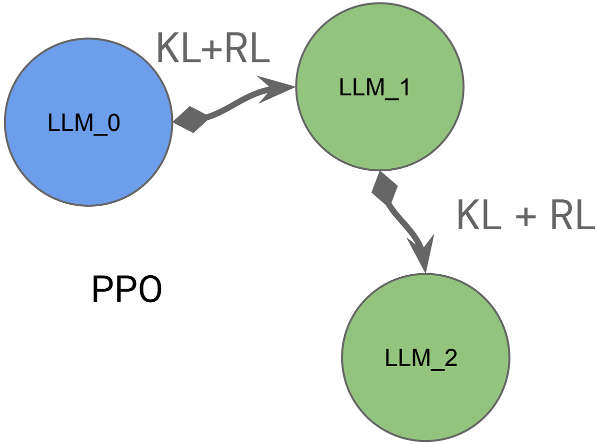
How to use KL?

- Anchoring to the original model is desired!
- KL divergence is generally used in 2 ways:
  - Anchor to the original model
    - "Easy" (just add -KL to the reward)
    - Expensive: requires a copy
  - Anchor to the current model
    - Cheaper (no copy needed)
    - Less safe (can move away)

Note: it also prevents babbling



Regularized MDP



PPO

Variance reduction

REINFORCE is noisy

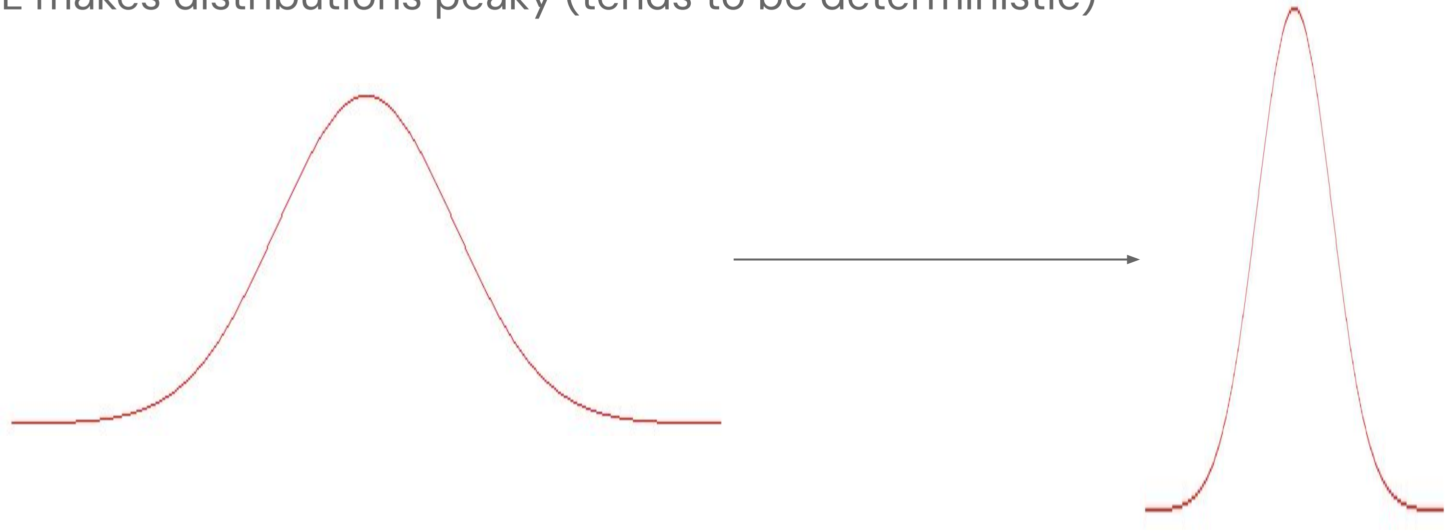
$$\hat{\nabla}_\theta J(\theta) = \frac{1}{D}\sum_{i=1}^{D}\left[\left(\sum_{t=1}^{N}\nabla_\theta \log \pi_\theta(w_t^i | w_{1:t-1}^i)\right)\left(\sum_{t=1}^{N} r_t^i\right)\right]$$

$$\hat{\nabla}_\theta J(\theta) = \frac{1}{D}\sum_{i=1}^{D}\left[\left(\sum_{t=1}^{N}\nabla_\theta \log \pi_\theta(w_t^i | w_{1:t-1}^i)\right)\left(\sum_{t=1}^{N} r_t^i \boxed{- b}\right)\right]$$

Better RL algorithms? How about efficiency?

Diversity

RL makes distributions peaky (tends to be deterministic)



What if several modes? Mode selection?

Multi-objective RL

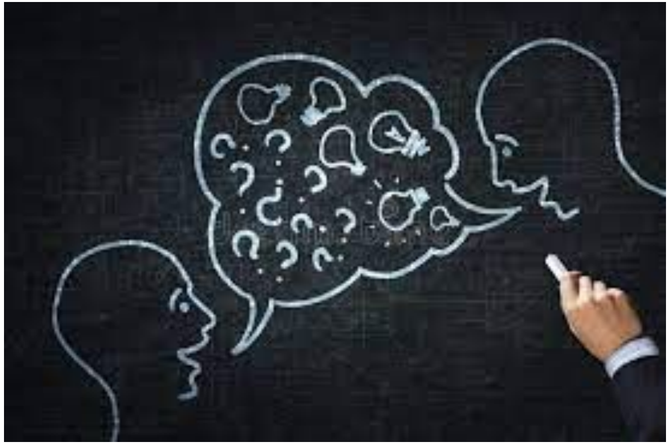
- Sentence Likelihood
- BLEU, ROUGE etc
- Factuality (related to Truthfulness)
- Verbosity (higher chances of hallucinations on long texts)
- Persona (not too much human-like, own personality)
- Toxicity (filtering)
- Neutrality (no strong opinion)
- …
- HF, which ones?

- Remember PARADISE (1997)
- Pareto front?

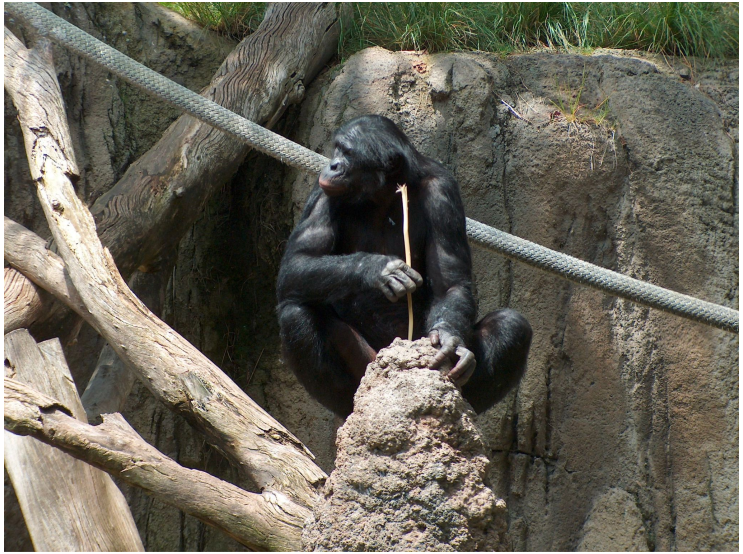
What’s next?

Multiple turns

Dialogue



Tool use



Wikipedia, no owner provided

Not a new problem

Using Markov Decision Process for Learning Dialogue Strategies

Esther Levin, Roberto Pieraccini, Wieland Eckert

AT&T Labs-Research,
180 Park Avenue, Florham Park, NJ 07932-0971, USA
(esther | roberto | eckert)@research.att.com

ICASSP, 1998

A Neural Conversational Model

Oriol Vinyals VINYALS@GOOGLE.COM
Google

Quoc V. Le QVL@GOOGLE.COM
Google

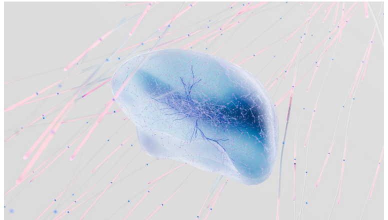
ICML, 2015: neural reset

In practice

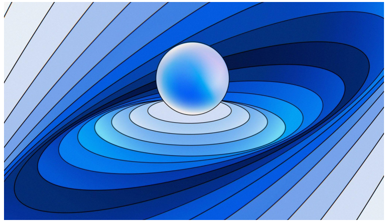
- Attention has its limits
  - Size of the mask
  - Necessary context may not be full context
- RL fine-tuning is not about conversational goals yet
  - It's about 1 turn ahead
- Technically more difficult:
  - Need to interact with outside world
  - Need sample efficiency
  - Non deterministic
  - Non stationary

RL IS BACK

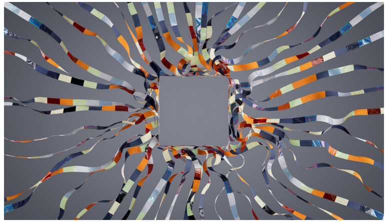
Other challenges:



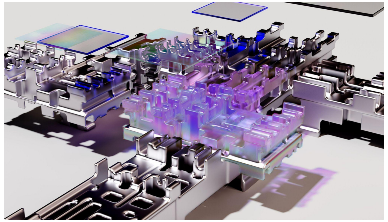
Personalization
How to adapt to 1 user or 1 group of users via RL?



Lifelong learning
How to learn from human long term behaviour?



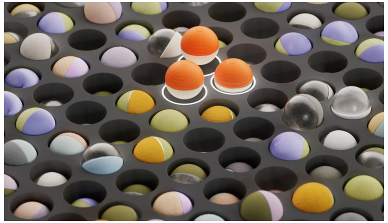
Can we teach machines
RL is about searching outside the SL data



More feedback
Rewriting, clicking?

Raters and users
are different

Human feedback is
Noisy, inconsistent, depends on the context



Benchmarks – metrics
How to navigate in huge models and datasets to assess properly performances

Data reuse
Most of the training data is thrown away, can we do offline RL (is it useful)?

Is it even RL?

I'd say yes :)

>>PROMPT: "a (not so) senior researcher questioning what drove their career"

Questions?