# THE ERA OF EXPERIENCE

Minha Hwang

# AGENDA

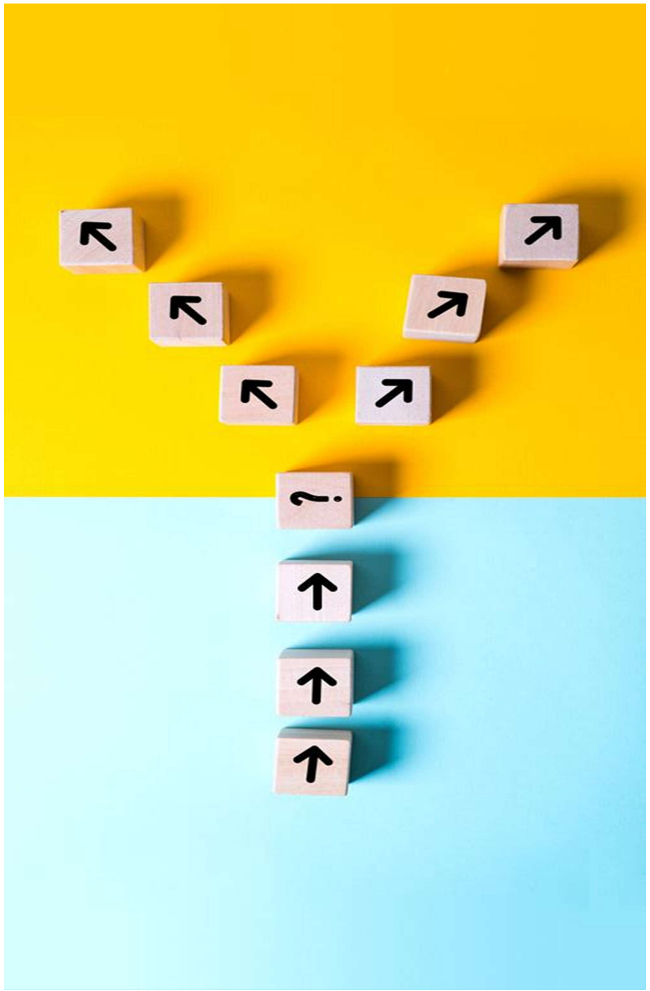
Three Era, LLM, RL

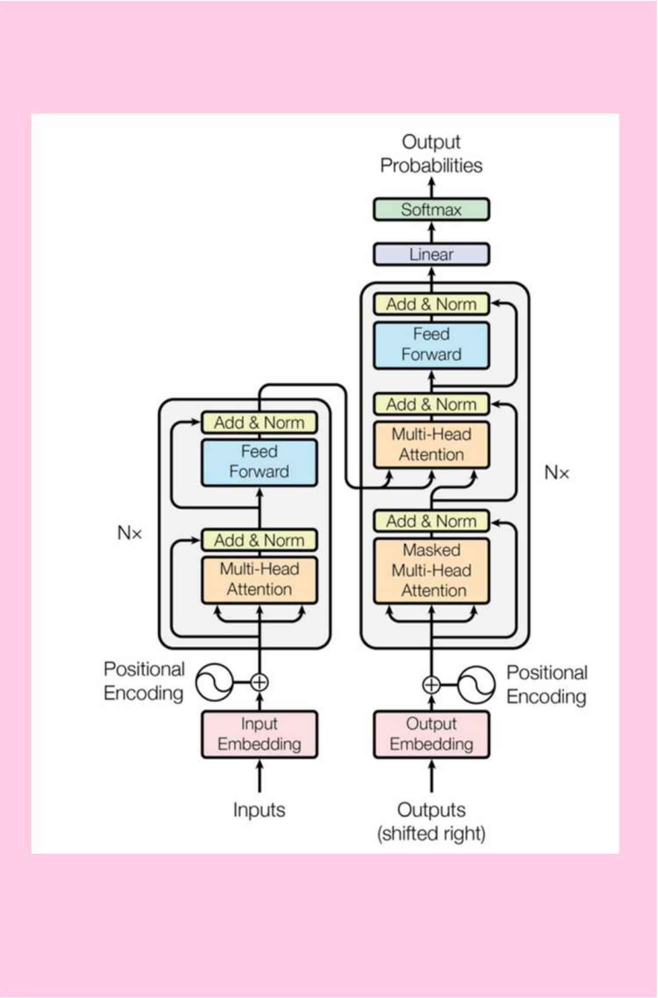
Data Scarcity

Key Points in the Paper

LLM 101

Connecting LLM and RL



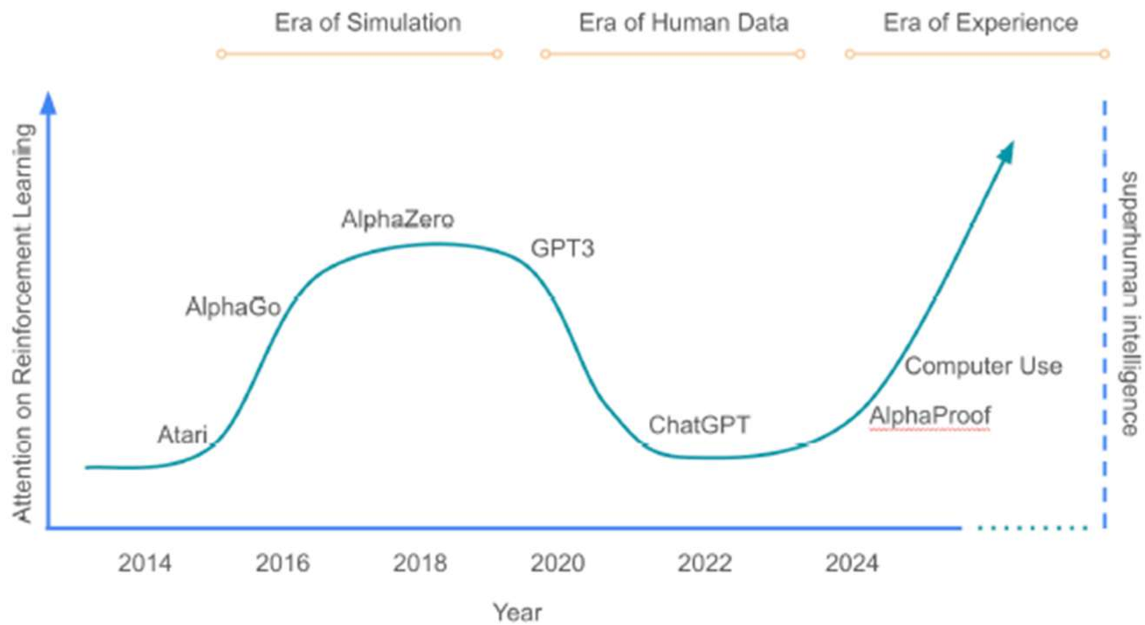


# THREE ERA, LLM, AND RL

# THE ERA OF EXPERIENCE

# (SILVER & SUTTON)

- Next major leap in AI capability: will come from agents that *learn predominantly from their own interactions with the world*
- Limit with static corpora of human-generated data



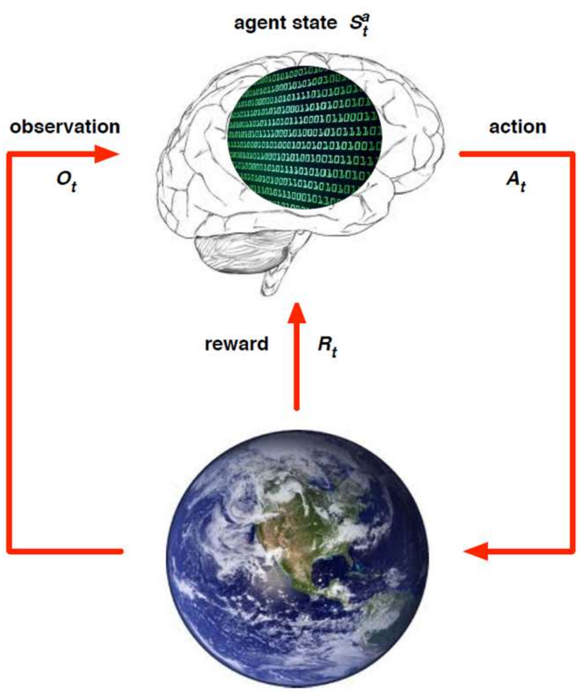

[The Era of Experience Paper.pdf]

# THREE ERA

| Era | Dominant data source | Typical paradigm | Limitation the authors highlight |
|---|---|---|---|
| Simulation | Synthetic data from game / physics simulators | Reinforcement learning (RL) self-play | **Narrow, closed-world tasks** |
| Human Data | Web-scale text & expert demonstrations | Supervised / RL-from-human-feedback | **Ceiling at "human-level" knowledge** |
| **Experience (proposed)** | **Agent-generated interaction streams** | **Continual RL with grounded rewards** | **Aims for open-ended, super-human discovery** |

[The Era of Experience Paper.pdf]

# REINFORCEMENT LEARNING: AGENT AND ENVIRONMENT




- At each step t, **the agent**:
  - Execute action A(t)
  - Receives observation O(t)
  - Receives scalar reward R(t)

- The **environment**:
  - Receives action A(t)
  - Emits observation O(t+1)
  - Emits scalar reward R(t+1)

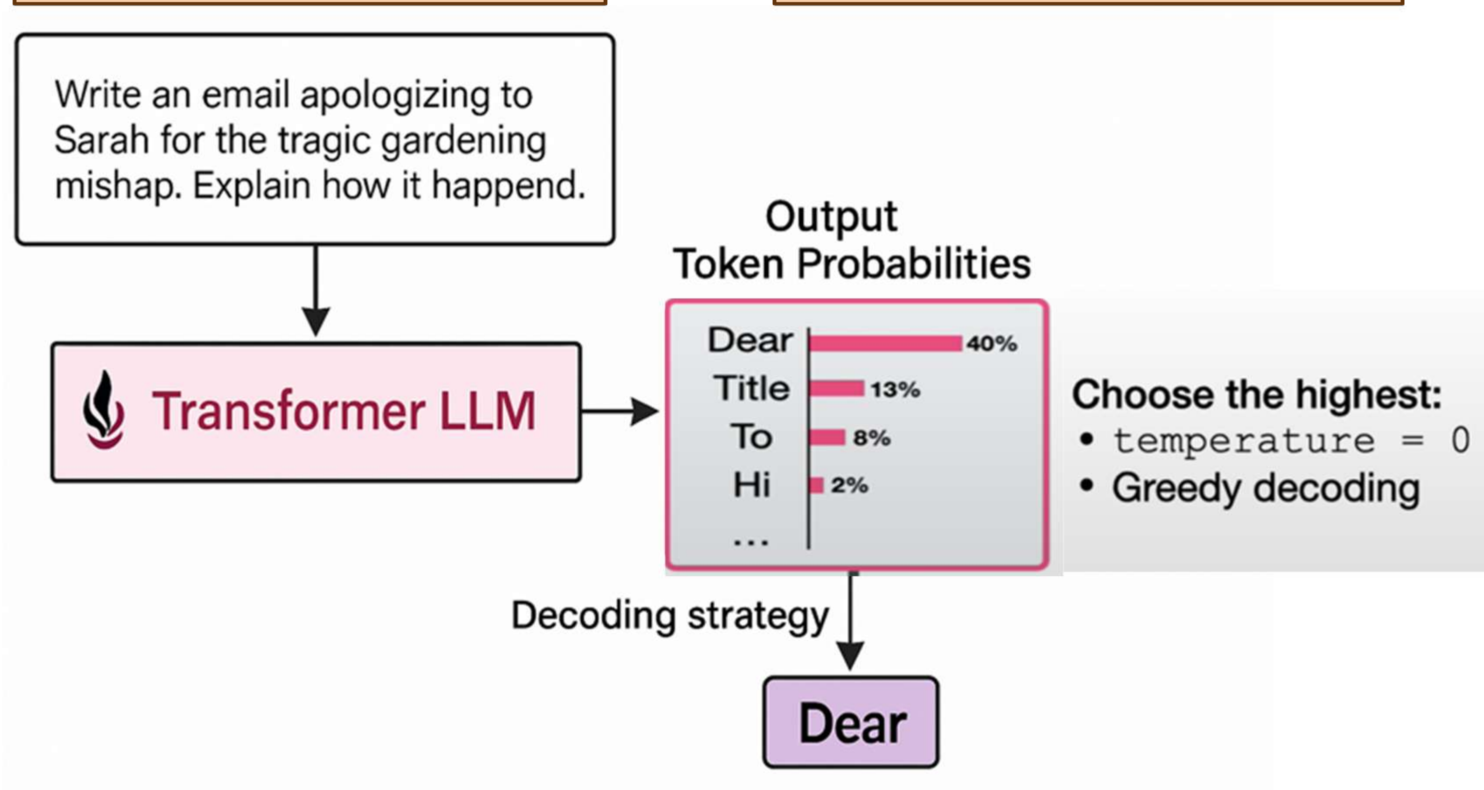
- **t increment** at environment step

- **Sequential Decision Making**
- Reward Hypothesis: All goals can be described by the maximization of **expected cumulative reward** **(scalar)**

Source: David Silver – Reinforcement Learning

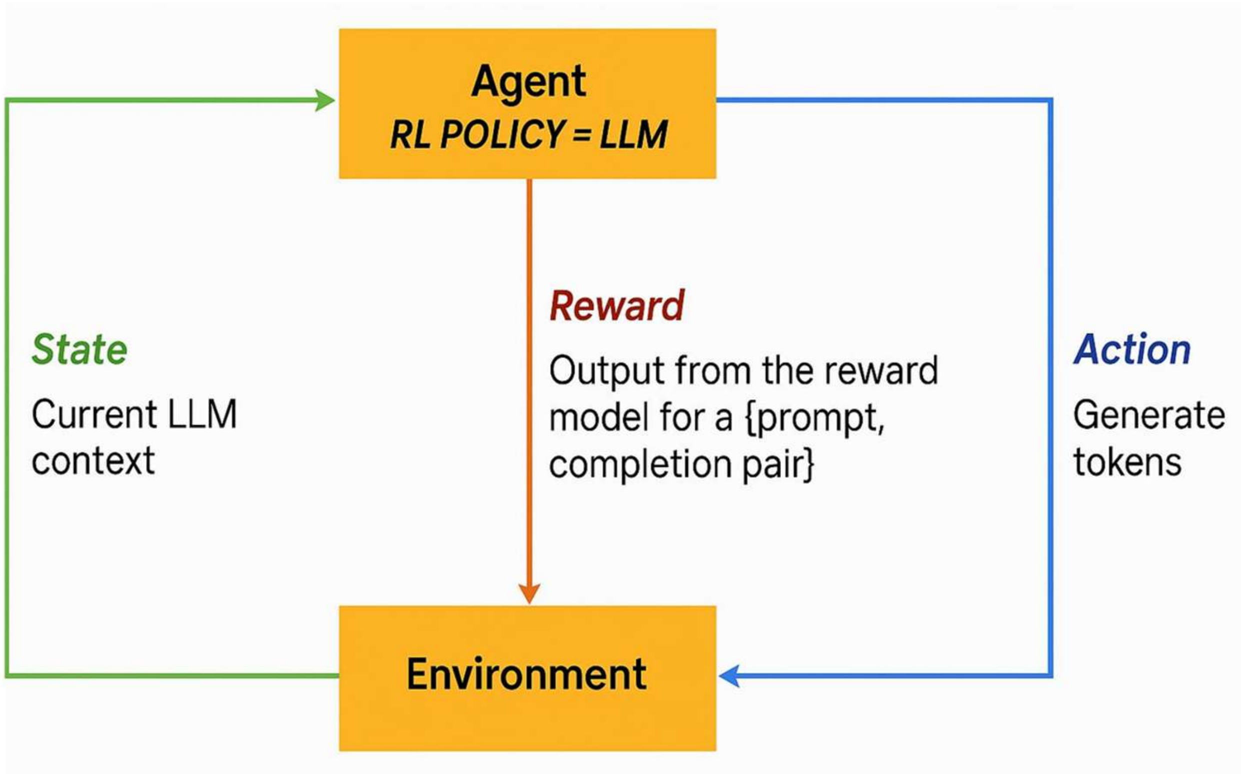
# TRANSFORMER LLM: INPUT AND OUTPUT

| Input | Output |
| --- | --- |




Source: [How Transformer LLMs Work - DeepLearning.AI]

# LLM AS RL AGENT




- **Proximal Policy Optimization (PPO)**
- **DeepSeek: GRPO**

Source: https://learn.deeplearning.ai/courses/reinforcement-learning-from-human-feedback

# DATA SCARCITY

# THE CHALLENGE OF DATA SCARCITY

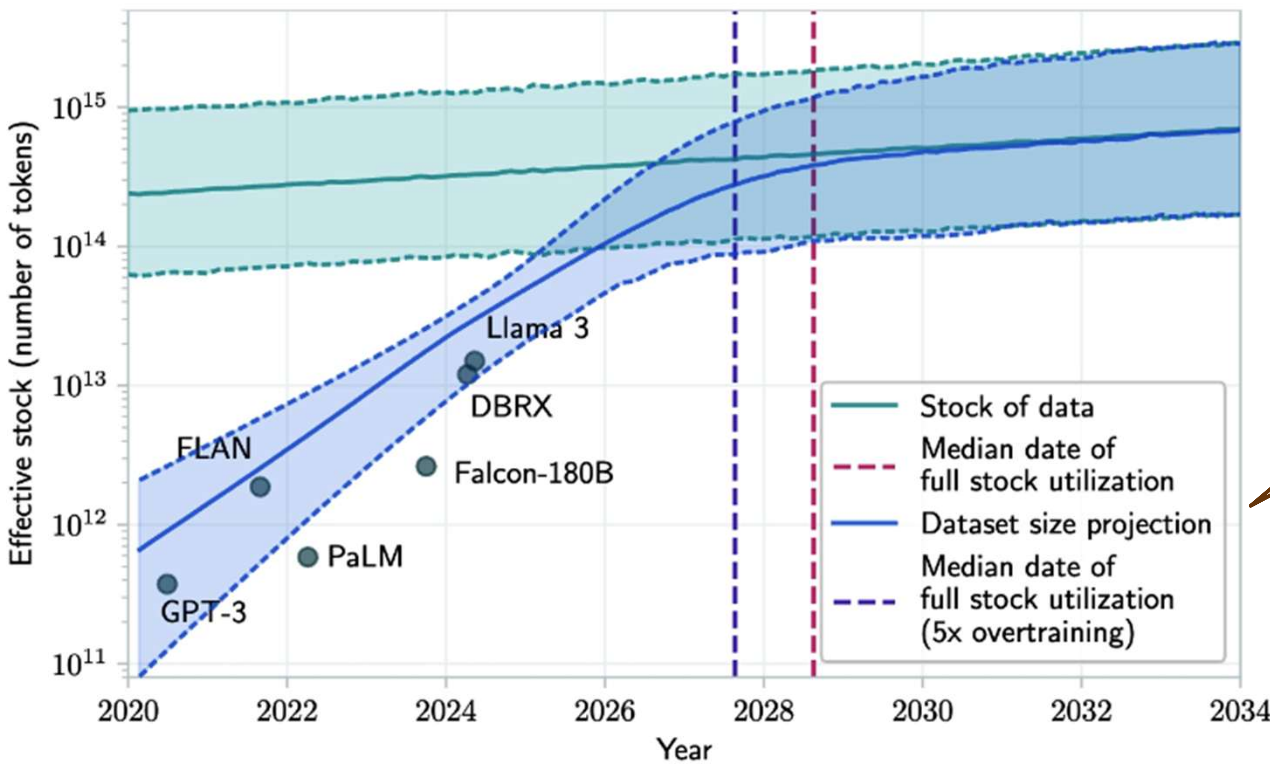



*Figure 2-9. Projection of historical trend of training dataset sizes and available data stock. Source: Villalobos et al., 2024.*

"**Pre-training** as we know it will **unquestionably end**…because we have **but one internet**," said OpenAI co-founder Ilya Sutskever at the NeurIPS 2024

- TRAINING DATASET LIMIT: 2026 AND 2032
- RECENT FOCUS ON **TEST TIME COMPUTE**: **REASONING** MODELS

[2211.04325] Will we run out of data? Limits of LLM scaling based on human-generated data

# PRE-TRAINING: CHINCHILLA SCALING LAW (DEEPMIND, 2022)

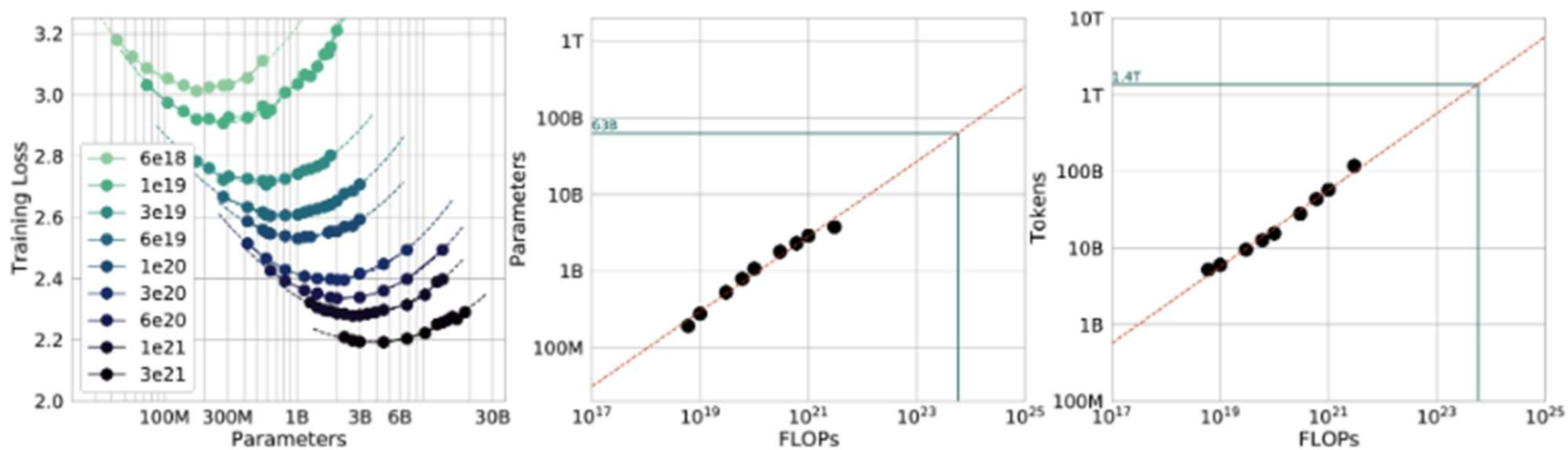



*Figure 2-8. Graphs that depict the relationships between training loss, a model's number of parameters, FLOPs, and number of training tokens. Source: "Training Compute-Optional Large Language Models" (DeepMind, 2022).*

**FLOPs (compute requirement for a task):** the number floating point operations performed for a certain task

Source: [2203.15556] Training Compute-Optimal Large Language Models

## NUMBER OF PARAMETERS VS. NUMBER OF TRAINING TOKENS

| Model | Size (# Parameters) | Training Tokens |
|---|---|---|
| LaMDA (Thoppilan et al., 2022) | 137 Billion | 168 Billion |
| GPT-3 (Brown et al., 2020) | 175 Billion | 300 Billion |
| Jurassic (Lieber et al., 2021) | 178 Billion | 300 Billion |
| *Gopher* (Rae et al., 2021) | 280 Billion | 300 Billion |
| MT-NLG 530B (Smith et al., 2022) | 530 Billion | 270 Billion |
| *Chinchilla* | 70 Billion | 1.4 Trillion |

- The number of **training tokens**: **20 times the model size**
- **30 trillion tokens**: 450 million books (5,400 times the size of Wikipedia)
- The number of **training tokens** = the **number of tokens in a model's dataset** x the **number of epoch**

Source: [2203.15556] Training Compute-Optimal Large Language Models

# KEY POINTS IN THE PAPER

# FOUR PILLARS OF THE "EXPERIENCE" PARADIGM

| Pillar | Author's key claim | Illustration / example |
| --- | --- | --- |
| **1. Streams of experience** | Agents should accumulate knowledge over *lifelong*, non-episodic interactions, enabling long-horizon goals and continual self-correction. | A health-coach model tracks months of wearable data to optimise long-term fitness. |
| **2. Rich actions & observations** | Agents must act through the same digital or physical interfaces humans use (mouse-clicks, code execution, lab robots), not merely language. | Recent "computer-use" agents that navigate UIs and call APIs autonomously. |
| **3. Grounded rewards** | Optimisation signals should come from *measurable consequences in the environment* (e.g., $CO_2$ levels, exam scores) rather than ex-ante human ratings. | AlphaProof generated 100 M new formal proofs via RL, surpassing pure imitation. |
| **4. Planning & reasoning over world models** | To avoid becoming an echo chamber of past human thought, agents should build predictive models of their environment and plan actions that maximise future grounded reward. | A science agent simulates material properties before lab synthesis. |

The Era of Experience Paper.pdf

## CORE REASONING

- **Human-data saturation** – High-quality human text/code is finite and largely consumed; incremental supervised scaling now yields diminishing returns, especially in domains like advanced mathematics or scientific invention.
- **Self-generated data scales with capability** – An agent that interacts, experiments, or plays against itself produces ever-harder training data as it improves, removing the external bottleneck. (AlphaZero, AlphaProof, DeepSeek-R1 cited as precedents.)
- **RL provides the algorithmic substrate** – Classic RL tools—value functions, exploration bonuses, world-model planning, temporal abstraction—are explicitly designed for continual, grounded interaction but were under-utilised in the human-data era (e.g., RLHF bypasses value estimation with human labels). The era of experience revives and extends these concepts.
- **Safety & alignment shifts** – Grounded rewards can *expose* misalignment early (because real-world metrics diverge) and can be *adaptively retuned* via a bi-level optimisation in which human feedback shapes the reward network itself, offering an incremental path to correct specification errors.
- **Societal impact** – Continuous-learning agents promise dramatic gains in personalised assistance and accelerated discovery, but also raise risks of autonomy, job displacement, and interpretability challenges; addressing these will require new governance and technical safeguards.

The Era of Experience Paper.pdf

# KEY TAKE-AWAYS

- **Design agents around long streams, not chat turns.** Architect memory, logging, and retraining pipelines to span months or years.
- **Expose agents to multimodal interfaces and execution feedback** so they can experiment and observe consequences.
- **Develop reward-learning modules** that flexibly combine environmental signals with lightweight human steering.
- **Revisit "classic" RL ideas** (e.g., optimistic exploration, options, Dyna-style model learning) in the context of LLM-scale function approximators and real-world data rates.
- **Prioritise continual evaluation & safety frameworks** that leverage the same streams of experience to detect and correct emergent misbehaviour.

By embracing these principles, the community can push beyond imitation toward systems that *discover* genuinely novel strategies, theories, and technologies—fulfilling the authors' vision of an AI era defined by experience rather than by static data

The Era of Experience Paper.pdf

# LLM 101

# LARGE LANGUAGE MODEL (LLM) TRAINING

## (1) Pre-Training: Base LLM (GPT3)

Predict **next word**, based on text training data

- **Self-supervised**

Once upon a time, there was a unicorn
that lived in a magical forest with
all her unicorn friends

What is the capital of France?
What is France's largest city?
What is France's population?
What is the currency of France?

## (2) Post-Training: Instruction/Preference Tuned LLM (ChatGPT)

Tries to follow instructions;Aligned with human preference

Fine-tune on instructions and good response pairs

Human Labeled Data: Instruction – Response Pair

- **Human Labeled Data**: Instruction – Response Pair
- **SFT**: Supervised Fine-Tuning
- **RLHF** (Reinforcement Learning with Human Feedback) or **DPO** (Direct Preference Optimization)

What is the capital of France?
The capital of France is Paris.

Source: ChatGPT Prompt Engineering for Developers - DeepLearning.AI

# AUTOREGRESSIVE LANGUAGE MODELS: PROBABILITY KERNEL

- A language model is a **probability kernel $\mu$** given a prefix of words: $\mu: X \rightarrow Pr(Y)$
  - Stochastic in nature: **A same prefix $X$** can give a **random output** sampled from a probability distribution $\mu_X$(i.e., generative) → A key reason for factual inaccuracy, inconsistency or hallucination (making stuff up)

- A language model calculates $Pr(s)$ given a sequence of words: $s = (w_1, w_2, \ldots\ldots., w_{T-1}, w_T)$

- An autoregressive language model calculates this **conditional on a previous sequence of words**:

$$Pr(s) = Pr(w_1, w_2, \ldots\ldots., w_{T-1}, w_T)$$
$$= \prod_{t=1}^{T} Pr(w_t | w_1, w_2, \ldots\ldots., w_{t-1})$$

  - **Next-word prediction**: Given a prefix $(w_1, w_2, \ldots\ldots., w_{t-1})$, calculate the probability of the next word $w_t$ (Conceptually same to time series with path dependence)

Source: Prof. Kyunghyun Cho

# AUTOREGRESSIVE LANGUAGE MODELS: SIMPLE EXAMPLE

- 4-word sentence example: "I am a student"

$$Pr(s) = Pr(w_1, w_2, w_3, w_4) = Pr(w_1) \times Pr(w_2|w_1) \times Pr(w_3|w_1, w_2) \times Pr(w_4|w_1, w_2, w_3)$$

- All you need is **"counting"** (if there are large amounts of data)

$$Pr(w_2|w_1) = \frac{count(w_1, w_2)}{count(w_1)} \longrightarrow \text{2-grams (Bigrams)}$$

$$Pr(w_3|w_1, w_2) = \frac{count(w_1, w_2, w_3)}{count(w_1, w_2)} \longrightarrow \text{3-grams (Trigrams)}$$

$$Pr(w_4|w_1, w_2, w_3) = \frac{count(w_1, w_2, w_3, w_4)}{count(w_1, w_2, w_3)} \longrightarrow \text{4-grams}$$

- Problems:
  - This requires **a lot of space (RAM)**
  - Count-based language models **cannot generalize**: A certain sentence **does not appear** in the corpus

Source: Prof. Kyunghyun Cho

# PRE-TRAINING: SELF-SUPERVISED LEARNING

**Input**

- A sequence of tokens (encoded words)

**Loss Function**

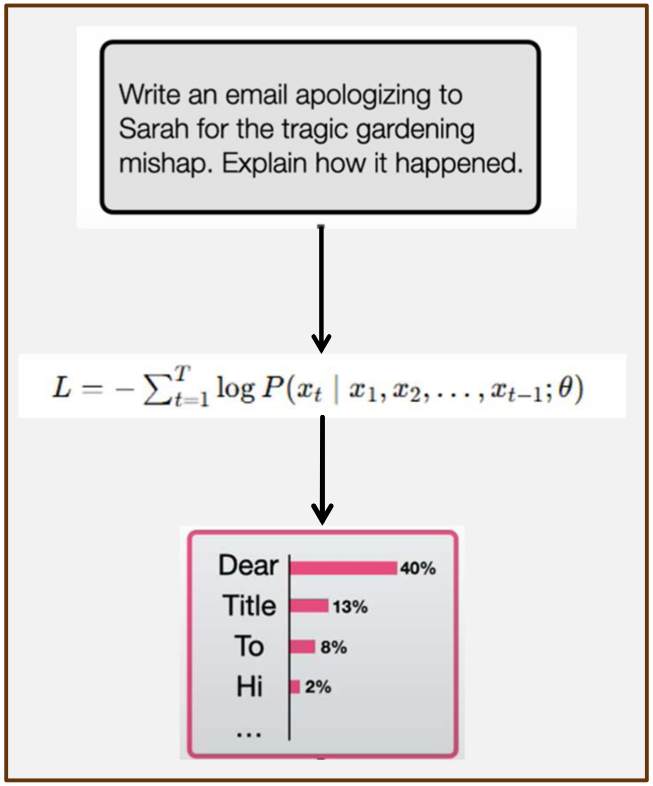
- Minimize the **negative log-likelihood** of the predicted token given the preceding tokens: **Cross-entropy Loss**
- Mathematically, the loss L for a sequence of tokens $(x_1, x_2, \ldots, x_T)$ is:

**Output**

- **Probability distribution** over the **vocabulary** (~30,000)
- Deterministic

**Dataset Size**

- Neural scaling law: The dataset size (D) should **scale proportionally** with the model size (i.e., linear)
  - e.g., GPT-3: 175B parameters, 300B tokens




Source: DeepLearning.AI – How Transformer LLM Works; Chinchilla Scaling: Hoffmann, et al, 2022

# PRE-TRAINED LLM EVAL - PERPLEXITY: PREDICTIVE ACCURACY

**Example Calculation**: "The cat sat on the mat"

- P("The") = 0.2
- P("cat"|"The") = 0.1
- P("sat"|"The cat") = 0.15
- P("on"|"The cat sat") = 0.3
- P("the"|"The cat sat on") = 0.25
- P("mat"|"The cat sat on the") = 0.05

$$\text{Perplexity}(PP) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i \mid w_{<i})\right)$$

First, calculate the average negative log probability:

$$-\frac{1}{6}\left(\log(0.2) + \log(0.1) + \log(0.15) + \log(0.3) + \log(0.25) + \log(0.05)\right) \approx 1.8992$$

Then, exponentiate to find perplexity:

$$PP = \exp(1.8992) \approx 6.68$$

This means the model, on average, considers about 6.68 possible next words, indicating its uncertainty in prediction.

# POST-TRAINING: SUPERVISED FINE TUNING

**Input**

- Prompt-Desired Response Pair Dataset:
  {Prompt 1, Desired Response 1},
  {Prompt 2, Desired Response 2}

**Loss Function**

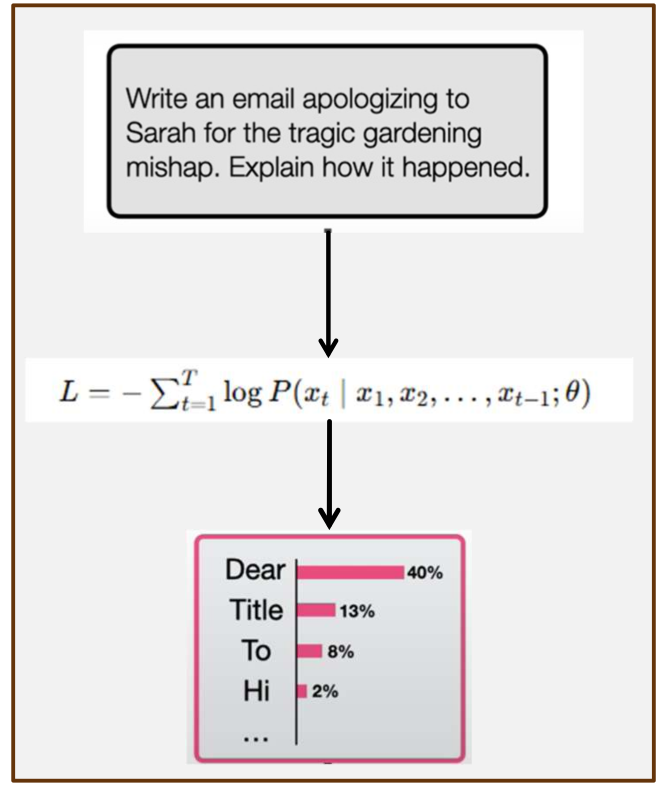
- **Cross-entropy Loss**
- Generated output is compared to the desired response using a loss function

**Output**

- **Modified Probability Distribution** over the **vocabulary** (~30,000)

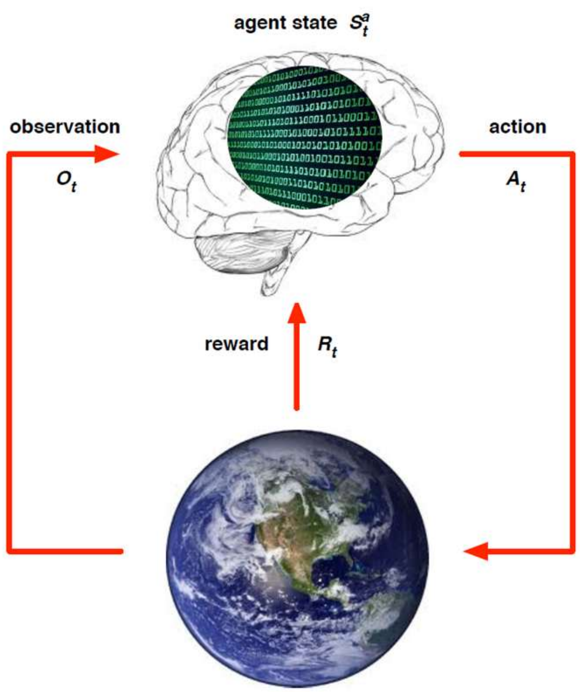
**Dataset Size**

- Task-specific models: **Less than 0.5%** of the original dataset: **Generalize to other data**, not just in fine-tuning data
- InstructGPT: 13,000 prompt-response pairs human labelers




Source: DeepLearning.AI – How Transformer LLM Works; Chinchilla Scaling: Hoffmann, et al, 2022

# REINFORCEMENT LEARNING: AGENT AND ENVIRONMENT

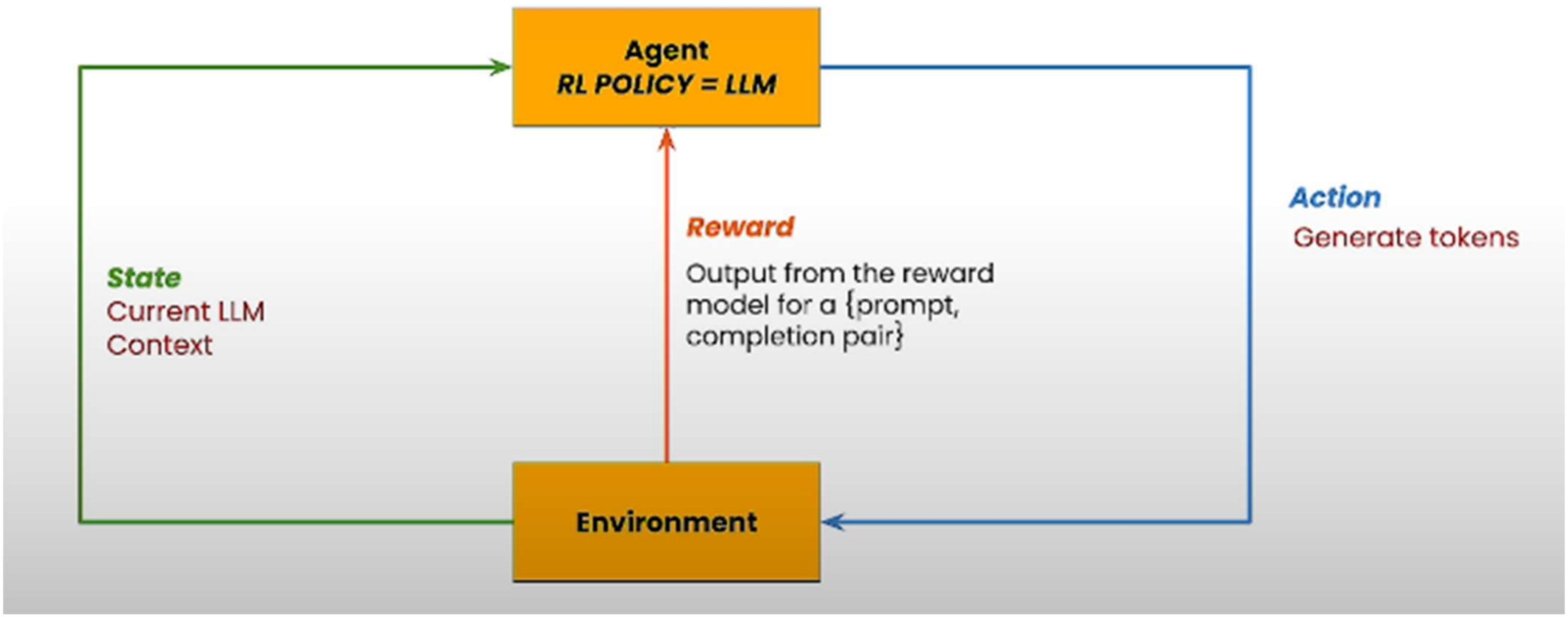



- At each step t, **the agent**:
  - Execute action A(t)
  - Receives observation O(t)
  - Receives scalar reward R(t)
- The **environment**:
  - Receives action A(t)
  - Emits observation O(t+1)
  - Emits scalar reward R(t+1)
- **t increment** at environment step

- **Sequential Decision Making**
- Reward Hypothesis: All goals can be described by the maximization of **expected cumulative reward** **(scalar)**

Source: David Silver – Reinforcement Learning

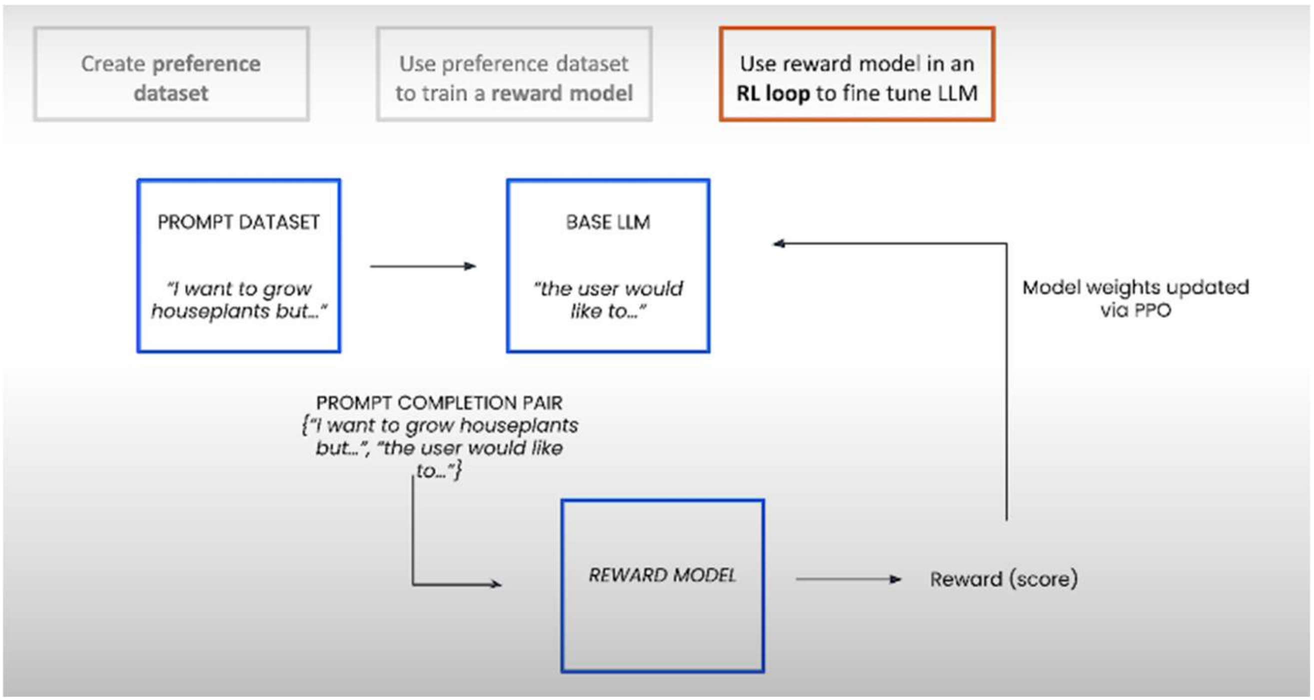
# POST-TRAINING: RLHF TO FINE TUNE LLM




- **Proximal Policy Optimization (PPO)**
- **DeepSeek: GRPO**

Source: https://learn.deeplearning.ai/courses/reinforcement-learning-from-human-feedbac

# POST-TRAINING: RLHF




- **Sequential Decision Making**
- Reward Hypothesis: All goals can be described by the maximization of **expected cumulative reward** **(scalar)**

Source: David Silver – Reinforcement Learning

# POST-TRAINING: RLHF – REWARD MODEL – TRAINING (1/3)

**Input**

- Preference Dataset - Pairwise {Prompt, Winning Candidate, Losing Candidate, Choice}
- Annotated by Human (Subjective)

**Loss Function**

- Minimize Pairwise Loss

$$\mathcal{L}(\theta) = -\frac{1}{\binom{K}{2}} \sum_{(x,y_w,y_l)} \log\left(\sigma\left(r_\theta(x,y_w) - r_\theta(x,y_l)\right)\right)$$

Here:

- $x$ is the prompt.
- $y_w$ and $y_l$ are the preferred and less preferred responses, respectively.
- $r_\theta(x,y)$ is the reward model's score for a given prompt-response pair.
- $\sigma$ denotes the sigmoid function.
- $K$ is the number of responses ranked by human annotators for each prompt.

**Dataset Size**

- 10K – 100K range
- InstructGPT:
  - Reward Model Dataset: ~ 33,000 examples. Human labelers ranked multiple responses to the same prompt

Source: DeepLearning.AI – How Transformer LLM Works; Chinchilla Scaling: Hoffmann, et al, 2022

# POST-TRAINING: RLHF – REWARD MODEL – INFERENCE (2/3)

**Input**

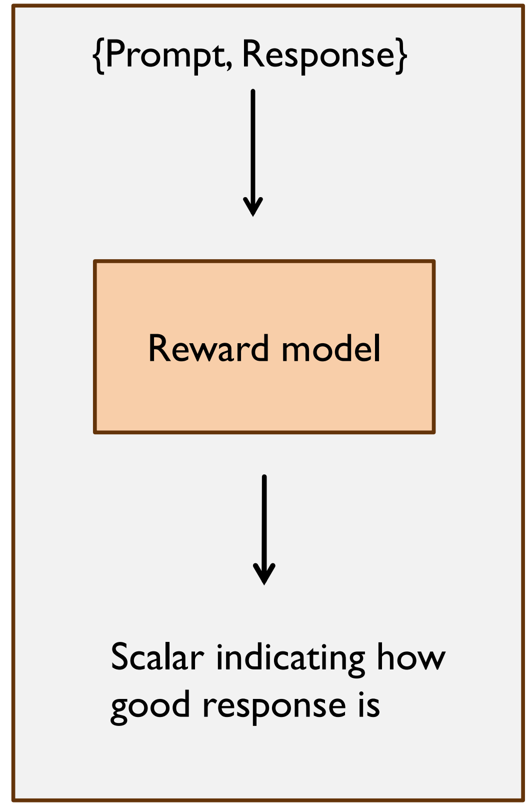
- Prompt Dataset - {Prompt, Response}

**Model**

- Reward Model (LLM), trained with Preference Dataset: {Prompt, Winning Candidate, Losing Candidate, Choice}

**Output**

- Scalar indicating how good response is




Source: DeepLearning.AI – How Transformer LLM Works; Chinchilla Scaling: Hoffmann, et al, 2022

# POST-TRAINING: RLHF – PROXIMAL POLICY OPTIMIZATION (3/3)

Input

- **Proximal Policy Optimization (PPO) Dataset**: InstructGPT ~31,000 prompts used to generate responses without human intervention during training, {Prompt}

Loss Function

**Policy Optimization**: The language model is fine-tuned using reinforcement learning to maximize the rewards predicted by the reward model. A common approach is to use Proximal Policy Optimization (PPO) with a loss function that balances achieving high reward and maintaining the model's output distribution close to the original model to prevent divergence:

$$L(\phi) = \mathbb{E}_{(x,y)\sim D_{\pi_\phi}} \left[ r_\theta(x,y) - \beta \log \left( \frac{\pi_\phi(y|x)}{\pi_{\text{SFT}}(y|x)} \right) \right]$$

In this equation:

- $\pi_\phi$ is the policy of the fine-tuned model.
- $\pi_{\text{SFT}}$ is the policy of the supervised fine-tuned model before reinforcement learning.
- $\beta$ is a scaling factor that controls the strength of the penalty for deviating from the original policy.

Source: DeepLearning.AI – How Transformer LLM Works; Chinchilla Scaling: Hoffmann, et al, 2022

# CONNECTING LLM AND RL

# CONNECTING RL AND LLMS (1/2)

Connection in LLM

| | |
|---|---|
| Sequential Decision Making | • Next token prediction to maximize prediction accuracy (pre-training) and rewards (post-training) over vocabulary (discrete set) |
| State | • Input prompt and previous generated tokens |
| Action | • Token chosen from the vocabulary |
| Reward | • Prediction accuracy + rewards from aligning with preference (weighted) |

$$r(s_t, y_t) = \lambda_1 \log \pi(y_t \mid s_t) + \lambda_2 \,\mathrm{Metric}(s_t, y_t),$$

Source: [Large Language Models as Reinforcement Learning Agents in Token Space: A Theoretical Framework by Miquel Noguer I Alonso :: SSRN]

# CONNECTING RL AND LLMS (2/2): APPLICABILITY OF RL RESEARCH

| | Description |
|---|---|
| Hierarchical Decision | • Motivated by AlphaStar's multi-scale decision-making<br>• High-level planning and detailed token generation |
| Self-dialogue | • Motivated by self-play training in AlphaGo, AlphaStar<br>• Self-dialogue: Models can engage in dialogue, critiquing and improving each other's outputs |
| Adaptive decoding | • Develop decoders that balance exploration and exploitation based on state uncertainty |
| Hybrid Models | • Combine maximum likelihood training with RL-based fine-tuning |

Source: Large Language Models as Reinforcement Learning Agents in Token Space: A Theoretical Framework by Miquel Noguer I Alonso :: SSRN