SSL, SFT, and RLHF: The ML Problems Behind LLMs

Sharan Sahu | Stats and Data Sci. PhD | Cornell University

Recent Adoption of LLMs

A multimodal generative AI copilot for human pathology

https://doi.org/10.1038/s41586-024-07618-3
Received: 11 December 2023
Accepted: 28 May 2024

Ming Y. Lu[1,2,3,4,11], Bowen Chen[1,2,11], Drew F. K. Williamson[1,2,3,11], Richard J. Chen[1,2,3], Melissa Zhao[1,2], Aaron K. Chow[5], Kenji Ikemura[1,2], Ahrong Kim[1,6], Dimitra Pouli[1,2], Ankush Patel[7], Amr Soliman[5], Chengkuan Chen[1], Tong Ding[1,8], Judy J. Wang[1], Georg Gerber[1], Ivy Liang[1,8], Long Phi Le[2], Anil V. Parwani[5], Luca L. Weishaupt[1,9] & Faisal Mahmood[1,2,3,10] ✉

Extracting structured information from unstructured histopathology reports using generative pre-trained transformer 4 (GPT-4)

Daniel Truhn[1], Chiara ML Loeffler[2,3,4], Gustav Müller-Franzes[1], Sven Nebelung[1], Katherine J Hewitt[2,4], Sebastian Brandner[5], Keno K Bressem[6], Sebastian Foersch[7] and Jakob Nikolas Kather[2,3,8,9]*

Health system-scale language models are
prediction engines

6160-y

Lavender Yao Jiang[1,2], Xujin Chris Liu[1,3], Nima Pou
Duo Wang[5], Anas Abidin[4], Kevin Eaton[6], Howard
Madeline Miceli[6], Nora C. Kim[1], Cordelia Orillac[1],
Hannah Weiss[1], David Kurland[1], Sean Neifert[1], Yo
Alexander T. M. Cheung[1], Grace Yang[1,2], Ming Ca
Yindalon Aphinyanaphongs[5,7], Kyunghyun Cho[2,

Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES): A method for populating Knowledge Bases using zero-shot learning

J. Harry Caufield[1], Harshad Hegde[1], Vincent Emonet[2], Nomi L. Harris[1], Marcin Joachimiak[1], Nicolas Matentzoglu[3], HyeongSik Kim[4], Sierra Moxon[1], Justin T. Reese[1], Melissa A. Haendel[5], Peter N. Robinson[6], and Christopher J. Mungall[1]

Validation of large language models for detecting pathologic complete response in breast cancer using population-based pathology reports

Ken Cheligeer[1,2], Guosong Wu[1,3], Alison Laws[4,5], May Lynn Quan[3,4,5], Andrea Li[1], Anne-Marie Brisson[6], Jason Xie[1] and Yuan Xu[1,3,4,5*]

Large multimodal model-based standardisation of pathology reports with confidence and its prognostic significance

, Gabriele Pergola[1], Harriet Evans[2,3], David Snead[1,2,3] and Fayyaz Minhas[1]

Curated Oncology Reports to
ge Model Inference

essa E. Kennedy, M.D.,[2] Divneet Mandair, M.D.,[2] Brenda Y. Miao, B.A.,[1]
Travis Zack, M.D., Ph.D.,[1,2] and Atul J. Butte, M.D., Ph.D.[1,2,3,4]

Received: September 1, 2023; Revised: December 27, 2023; Accepted: January 21, 2024; Published: March 13, 2024

Large language models for extracting histopathologic diagnoses from electronic health records

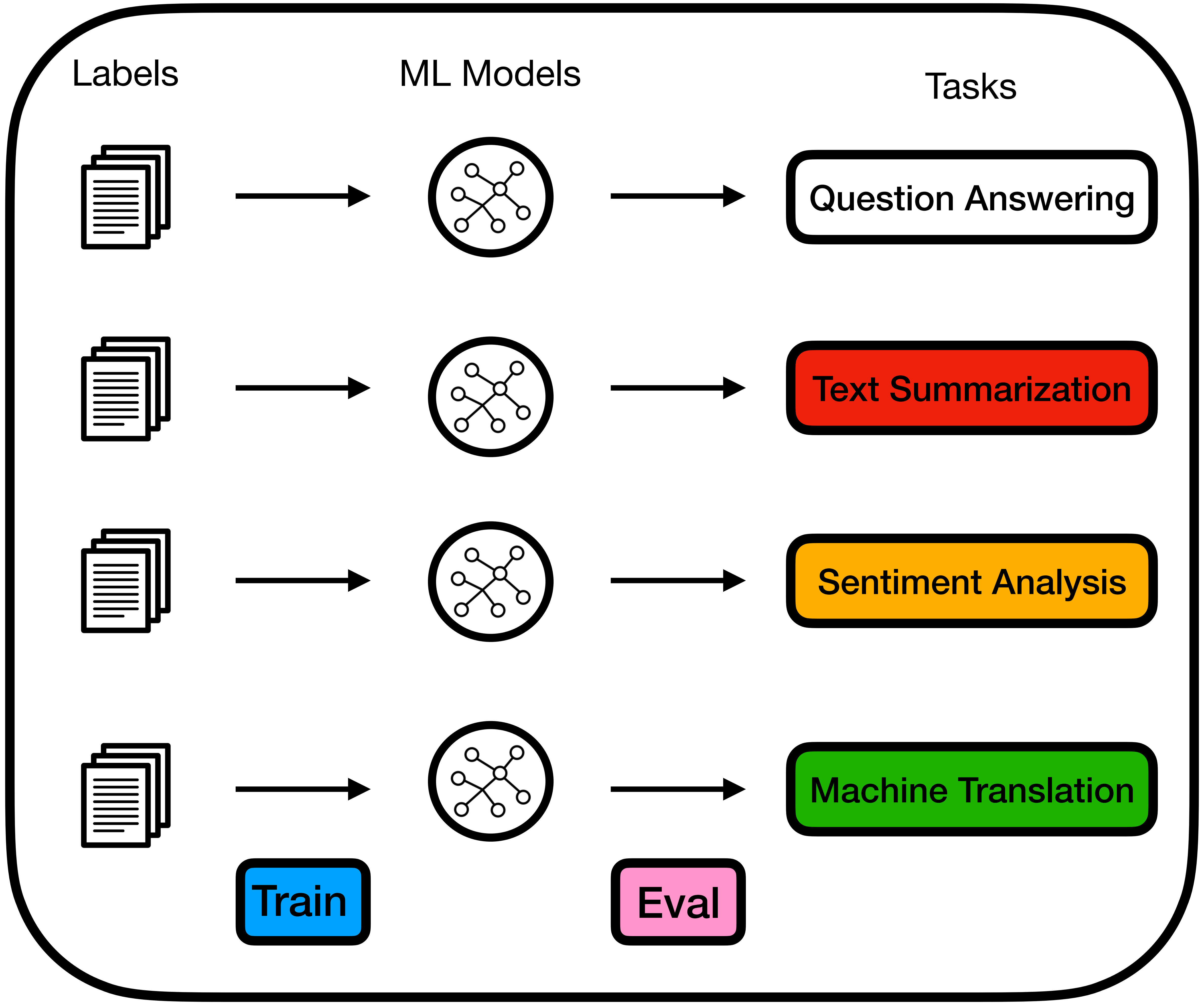
Brian Johnson, Tyler Bath, Xinyi Huang, Mark Lamm, Ashley Earles, Hyrum Eddington, Lily J. Jih, Samir Gupta, Shailja C. Shah, Kit Curtius

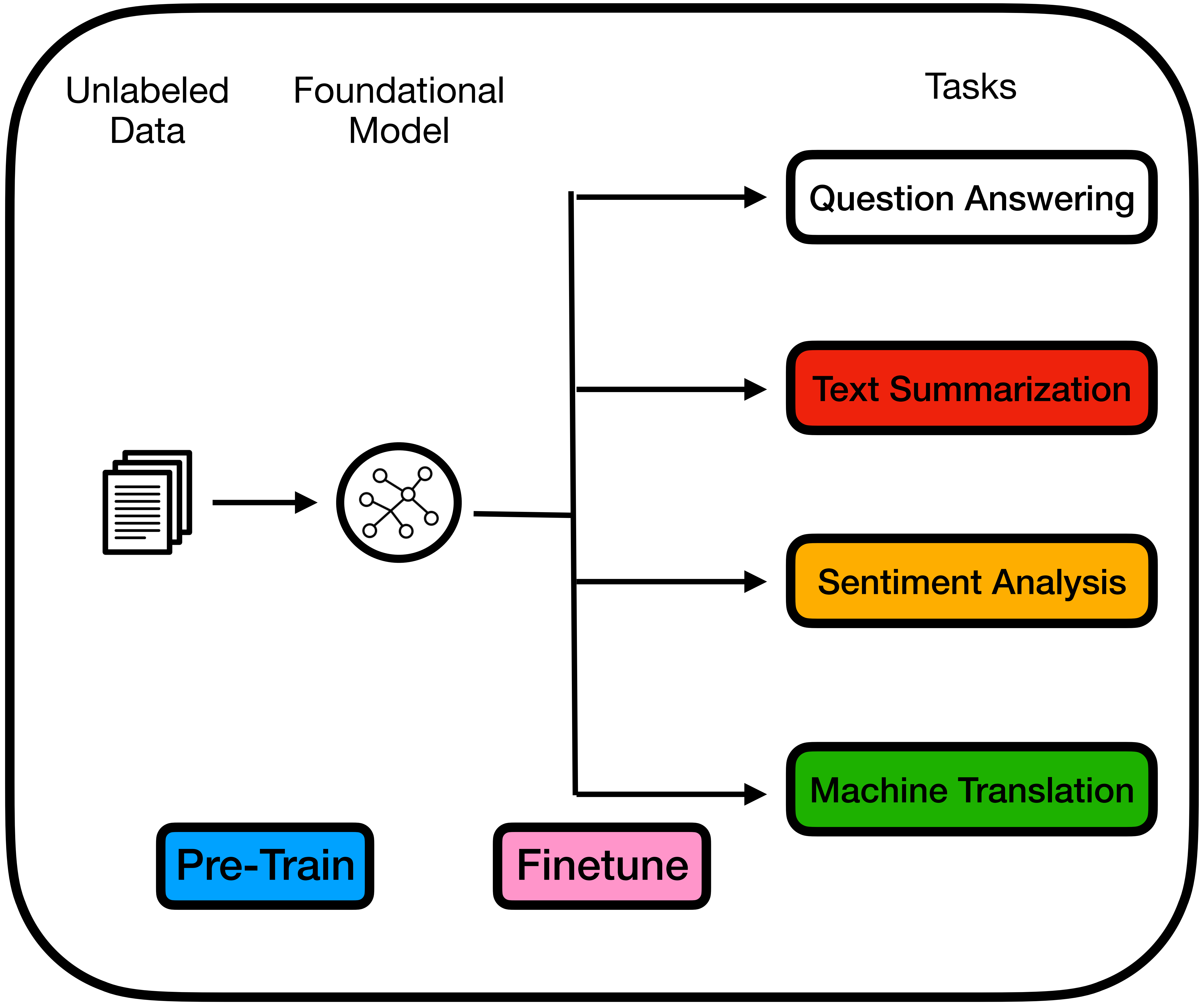
"A foundation model is **a large-scale machine learning model trained on a broad data set** that can be adapted and fine-tuned for a wide variety of applications"[1]

Foundational Models

[1] *On the Opportunities and Risks of Foundation Models by Bommasani et al. 2022*

Foundational Models





Traditional Machine Learning

Foundational Models

Diagram Credit: Kianté Brantley

Stages of LLM Training

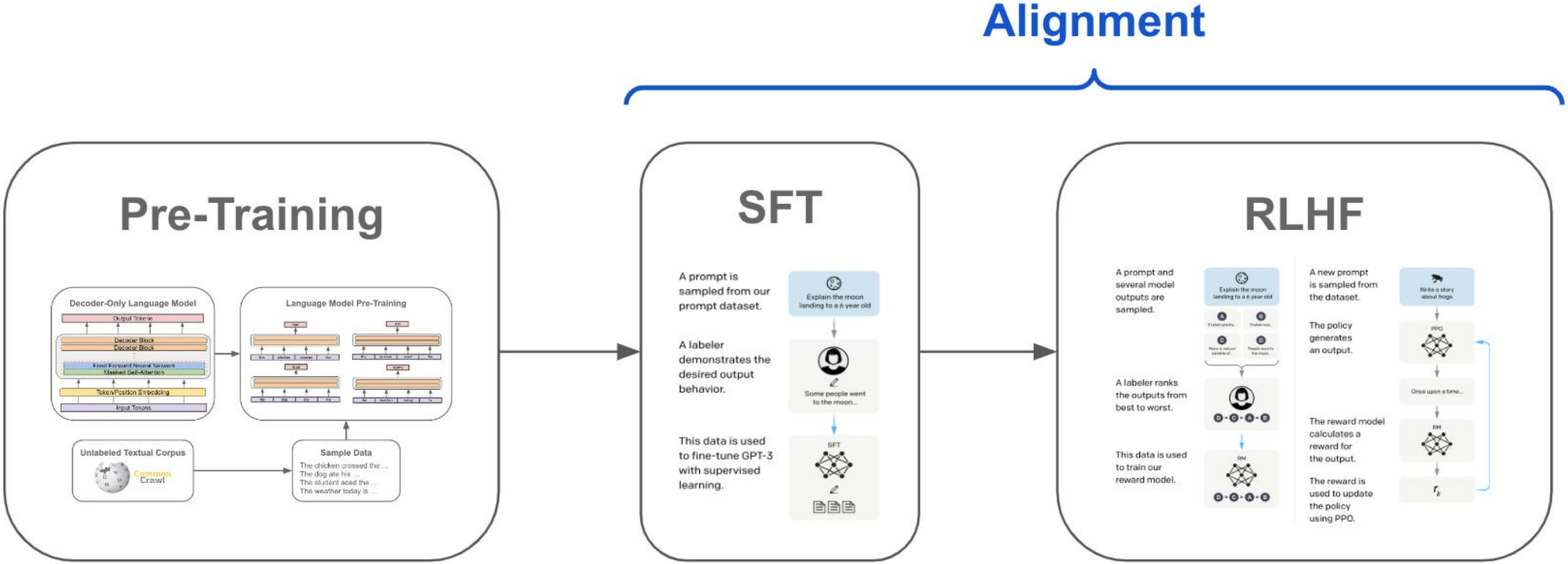


Diagram Credit: Training language models to follow instructions with human feedback by Ouyang, Long, et al.

Transformer Architecture

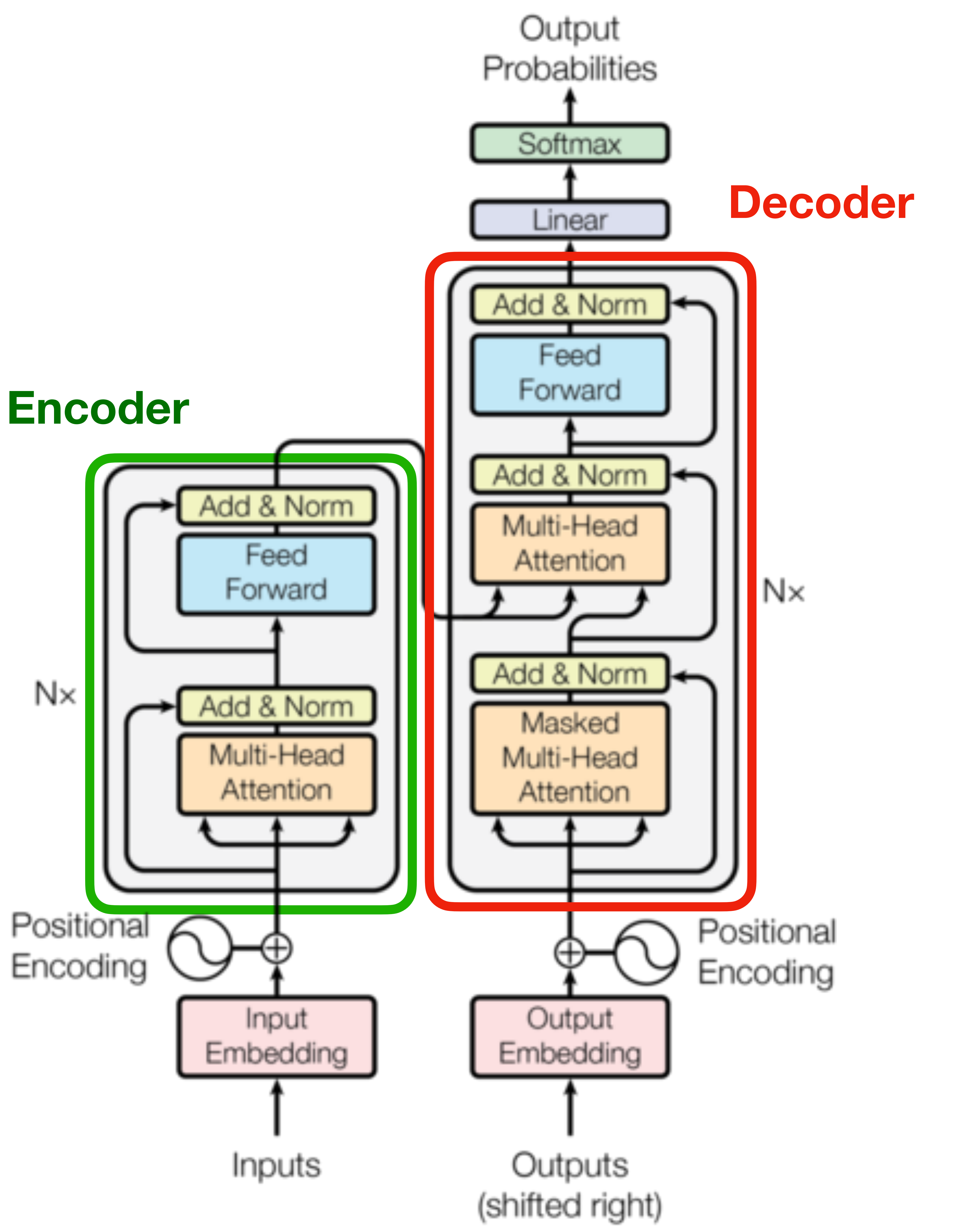
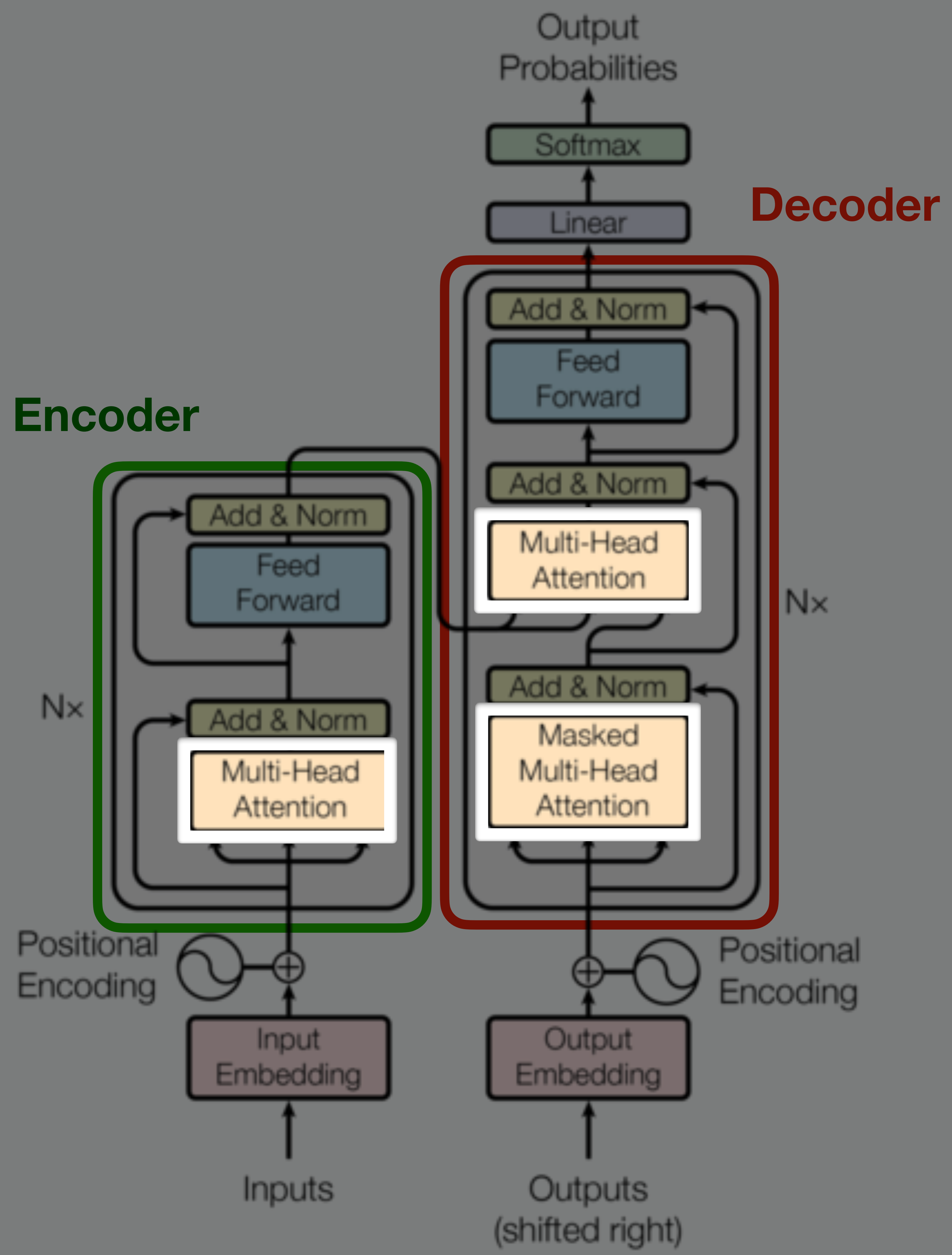


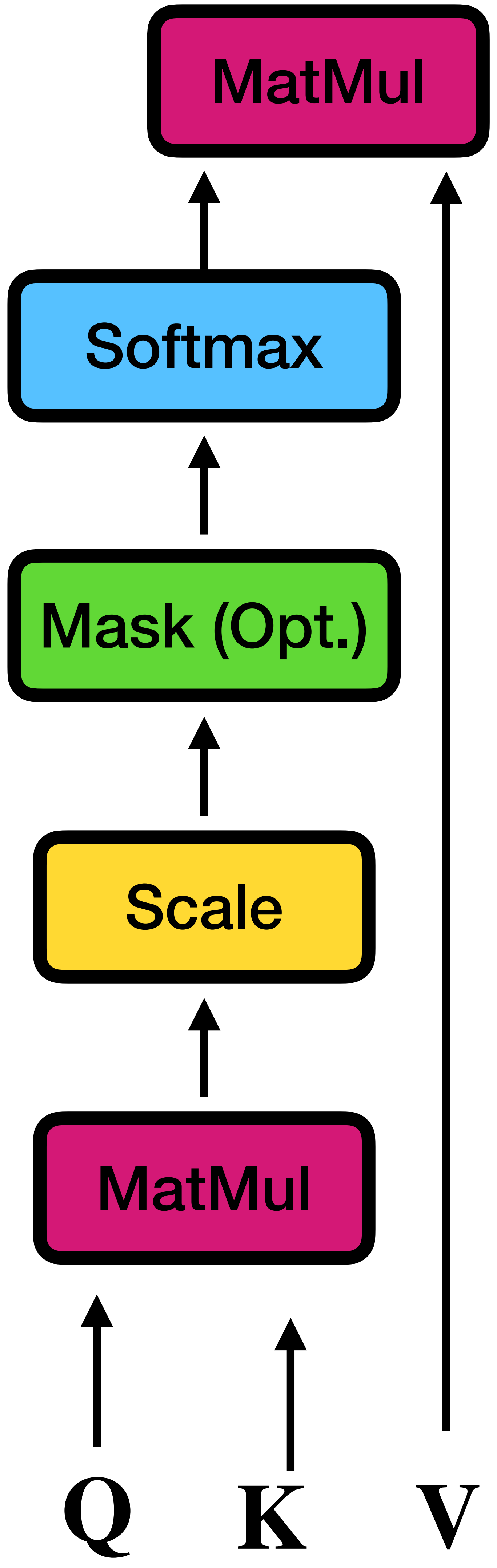
Diagram Credit: Attention Is All You Need by Vaswani et al.

Transformer Architecture



Scaled Dot-Product Attention

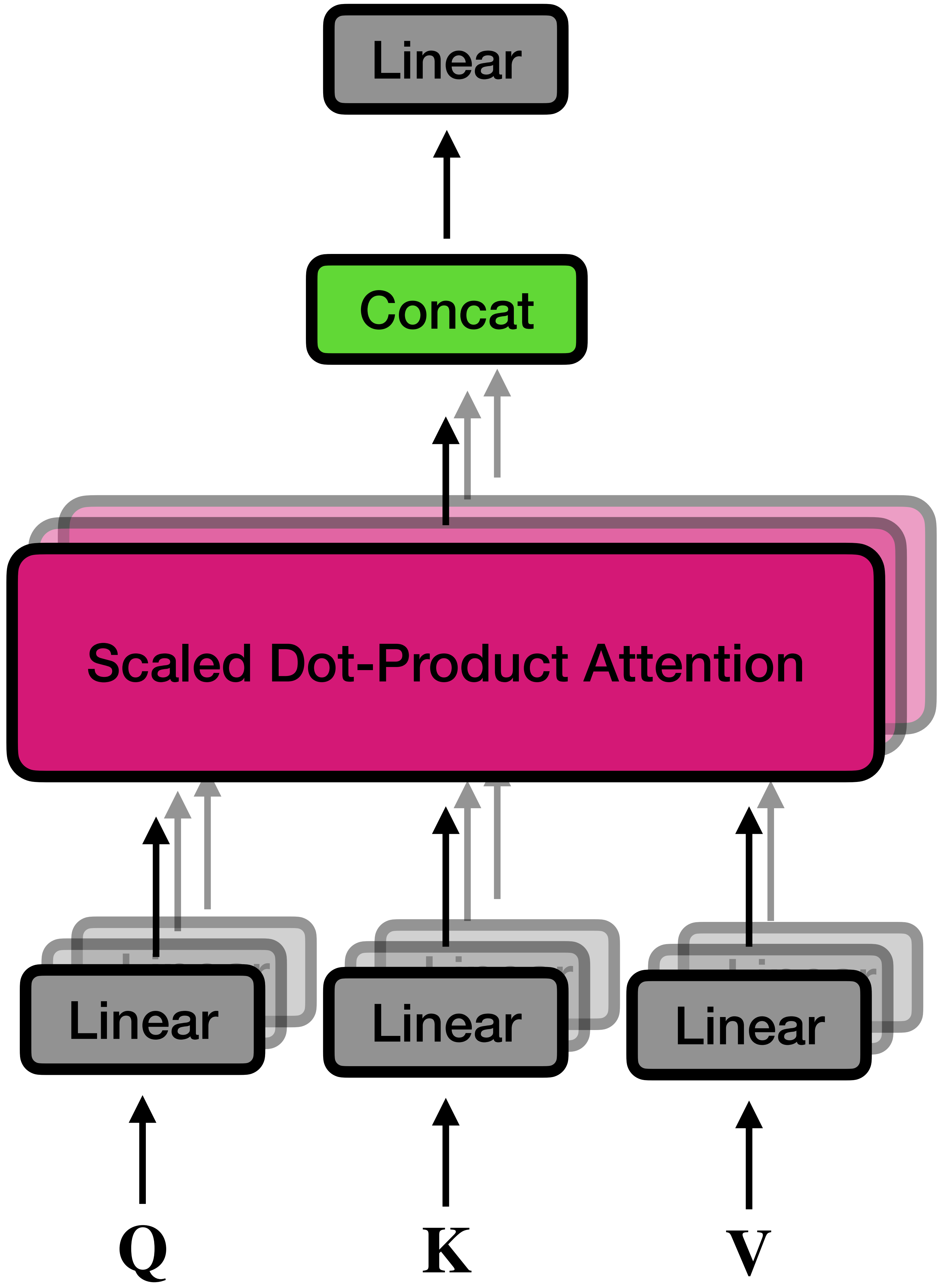
$$\mathrm{Attention}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}$$



Multi-Head Attention

$$\mathrm{MultiHead}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\mathrm{Concat}\left(\mathrm{head}_1,\ldots,\mathrm{head}_h\right)\mathbf{W}^{O}$$

$$\mathrm{head}_i=\mathrm{Attention}\left(\mathbf{Q}\mathbf{W}_i^{Q},\mathbf{K}\mathbf{W}_i^{K},\mathbf{V}\mathbf{W}_i^{V}\right)$$



First Stage of LLMs: Pre-Training (SSL)

Language Models

$$p(y_1, \ldots, y_n) = p(y_1)p(y_2 \mid y_1)\cdots p(y_n \mid y_1, \ldots, y_{n-1}) = \prod_{k=1}^{n} p(y_k \mid y_1, \ldots, y_{k-1})$$

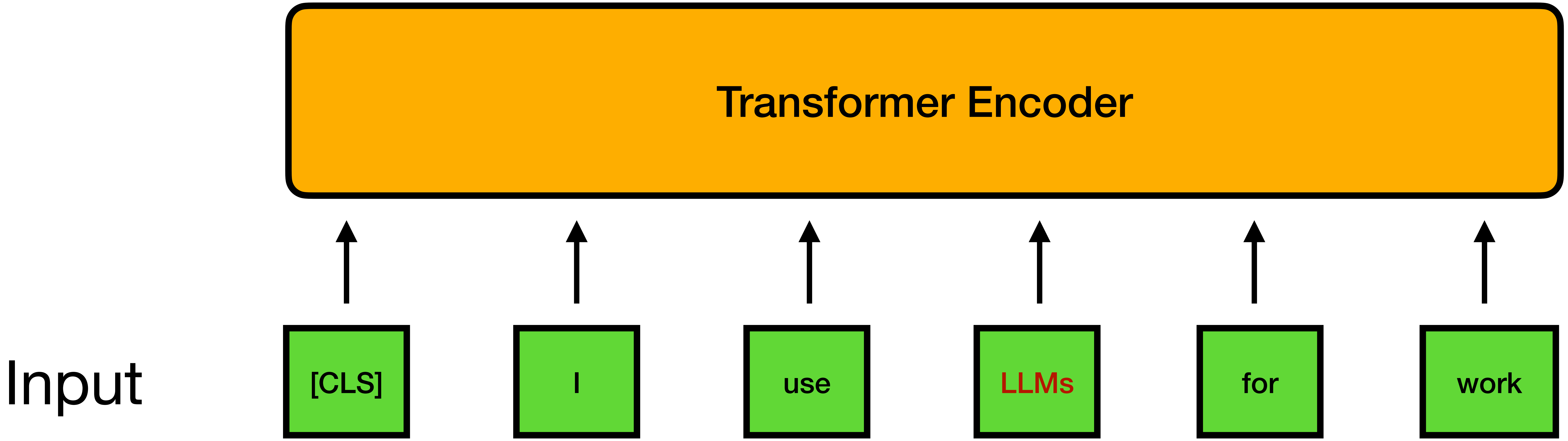
Pre-Train (SSL)

Large Unlabeled Text Data

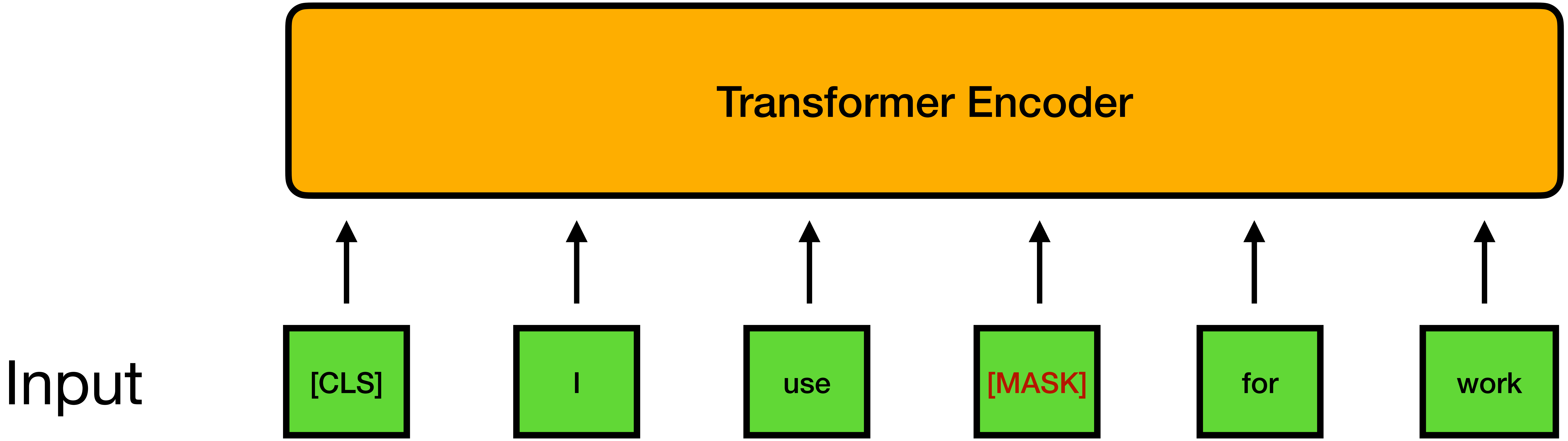
Diagram Credit: Kianté Brantley

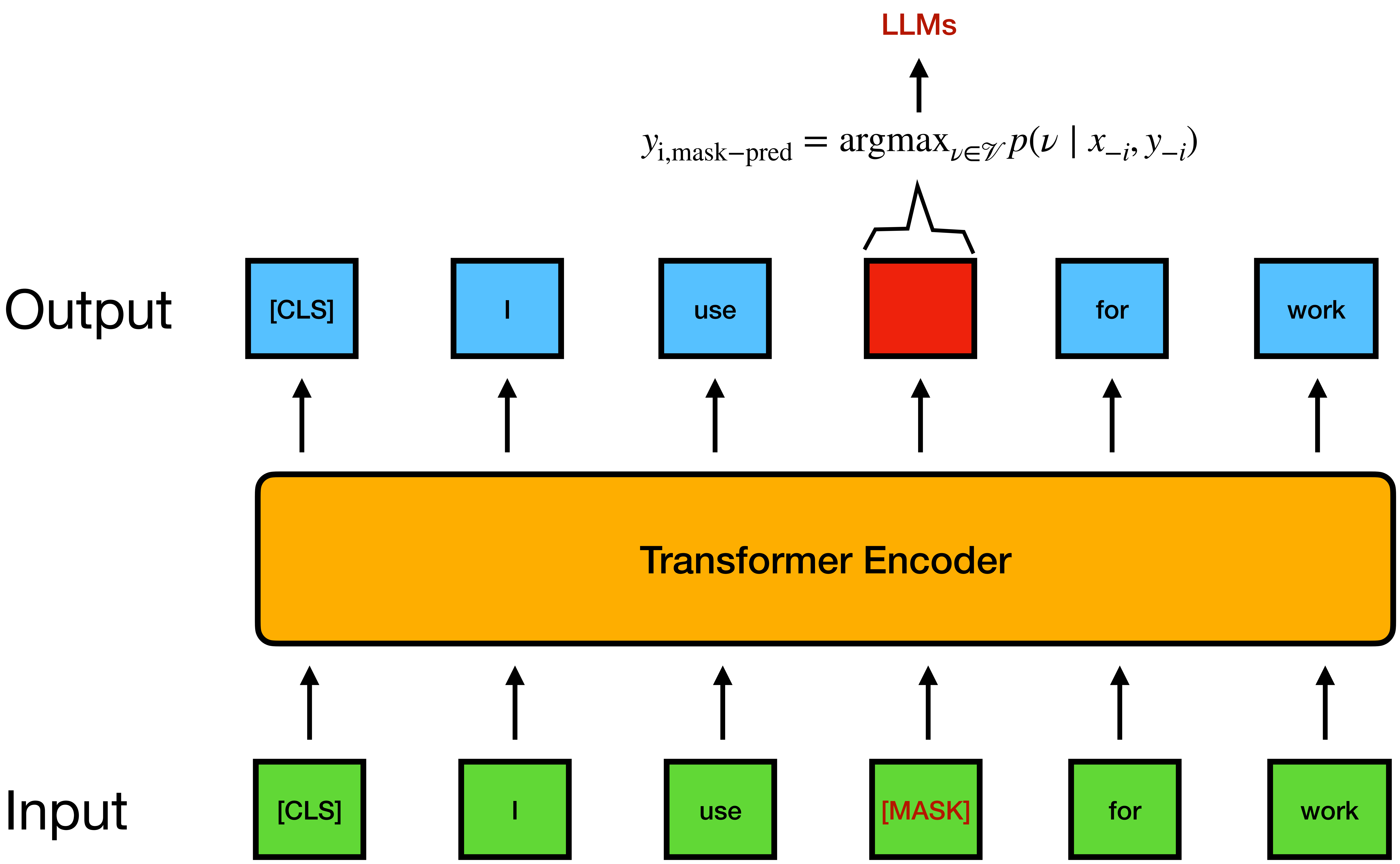
Pre-Training in Encoder Models

Masked Language Modeling (MLM)



Masked Language Modeling (MLM)




LLMs
$y_{\mathrm{i,mask-pred}} = \mathrm{argmax}_{\nu \in \mathcal{V}}\, p(\nu \mid x_{-i}, y_{-i})$
Output
[CLS]
I
use
for
work
Transformer Encoder
Input
[CLS]
I
use
[MASK]
for
work

LLMs

$y_{\text{i,mask-pred}} = \text{argmax}_{\nu \in \mathcal{V}}\, p(\nu \mid x_{-i}, y_{-i})$

Output

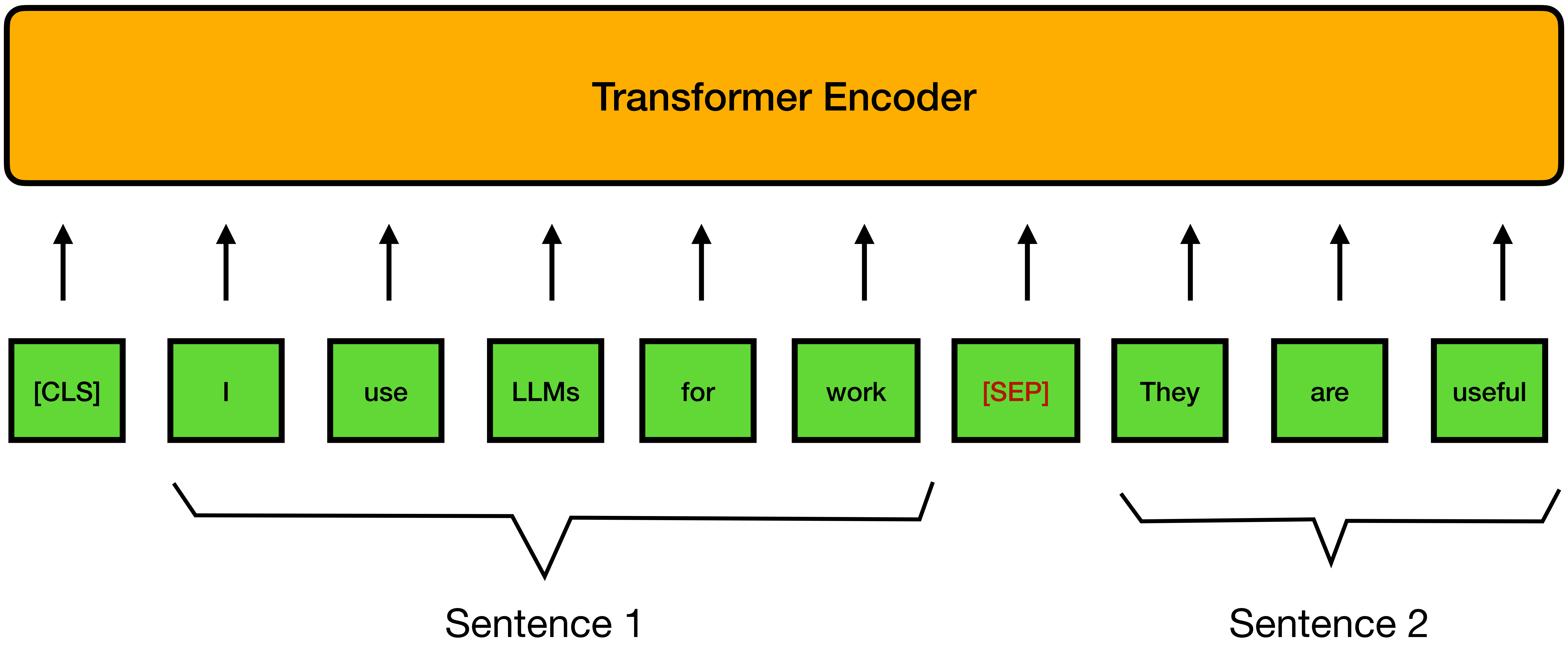
Let $m_i \sim_{\text{i.i.d}} \text{Unif}\,(1,n)$ for $i = 1,\ldots,k$ be the randomly chosen tokens to mask, $M = \{m_i\}_{i=1}^{k}$, and

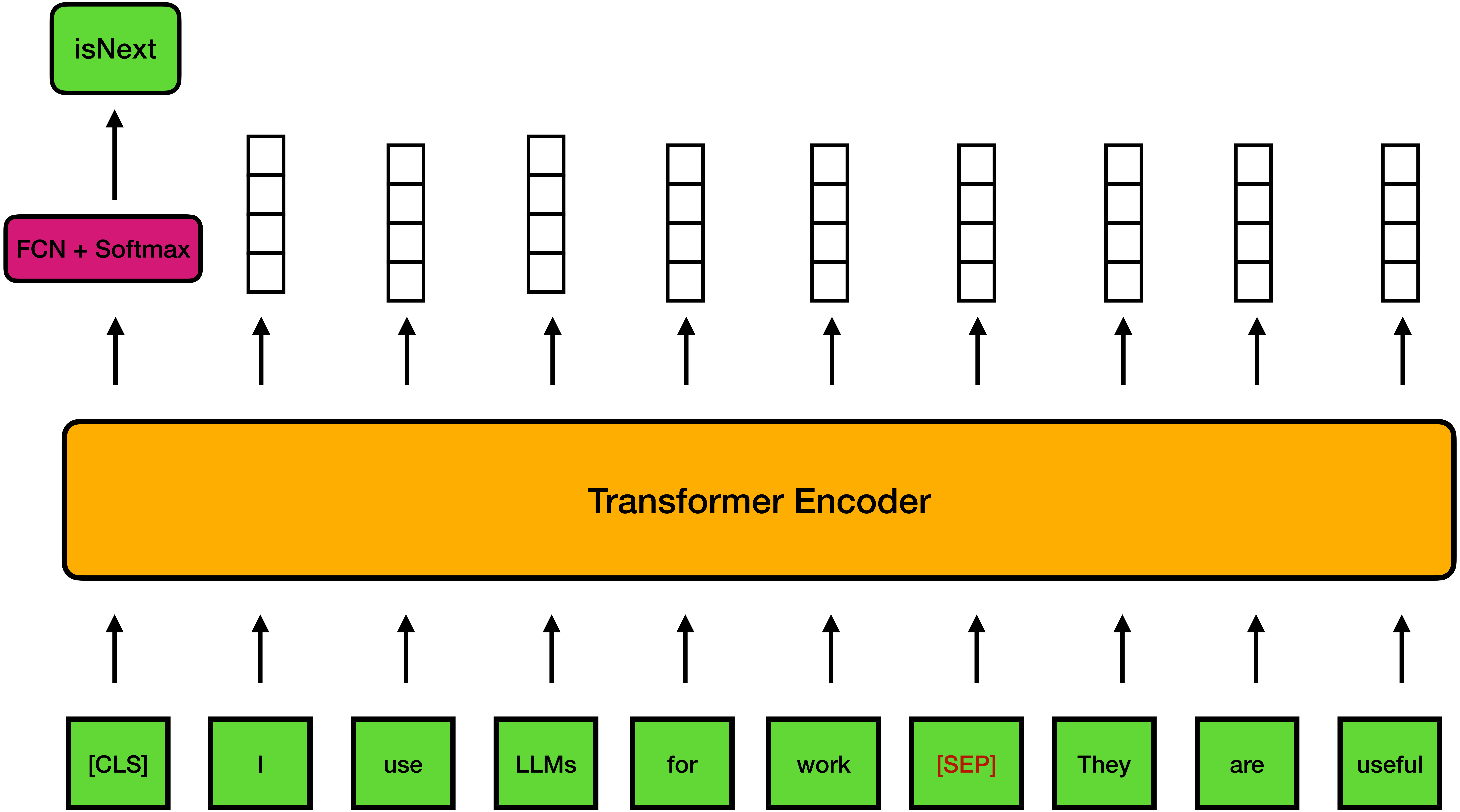
$$x^{\text{masked}} = \text{REPLACE}\,(x, m, [\text{MASK}])$$

$$\min_{\theta \in \Theta} \mathcal{L}_{\text{MLM}}\,(x, \theta) = \mathbb{E}\left[-\sum_{i \in M} \log\, \mathbb{P}_\theta \left(x_i \mid x^{\text{masked}}\right)\right]$$

Input

Next Sentence Prediction




isNext
FCN + Softmax
Transformer Encoder
[CLS]
I
use
LLMs
for
work
[SEP]
They
are
useful

Pre-Training in Encoder-Decoder Models

Consecutive Span Prediction

Original Text

I use LLMs for work! They are useful!

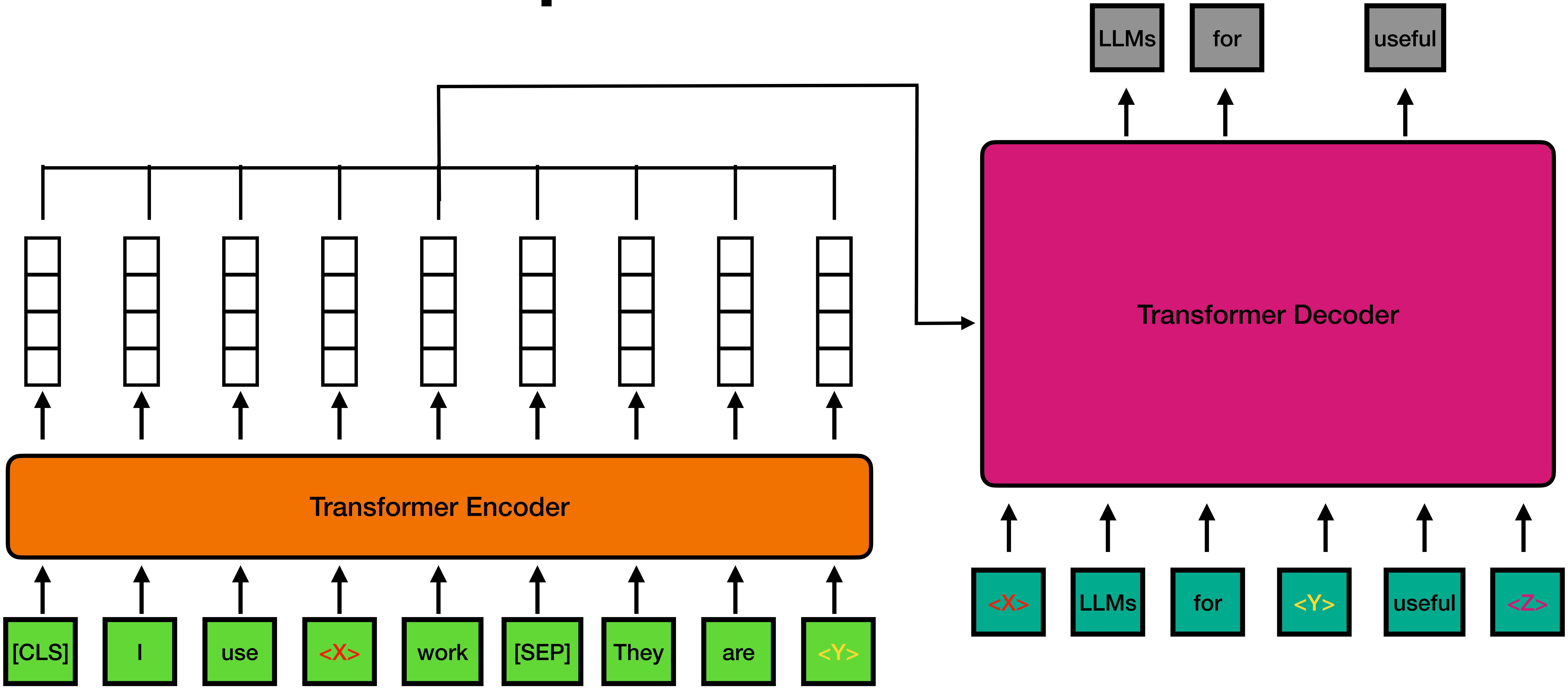
Input Text

I use <X> work! They are <Y>!

Target Text

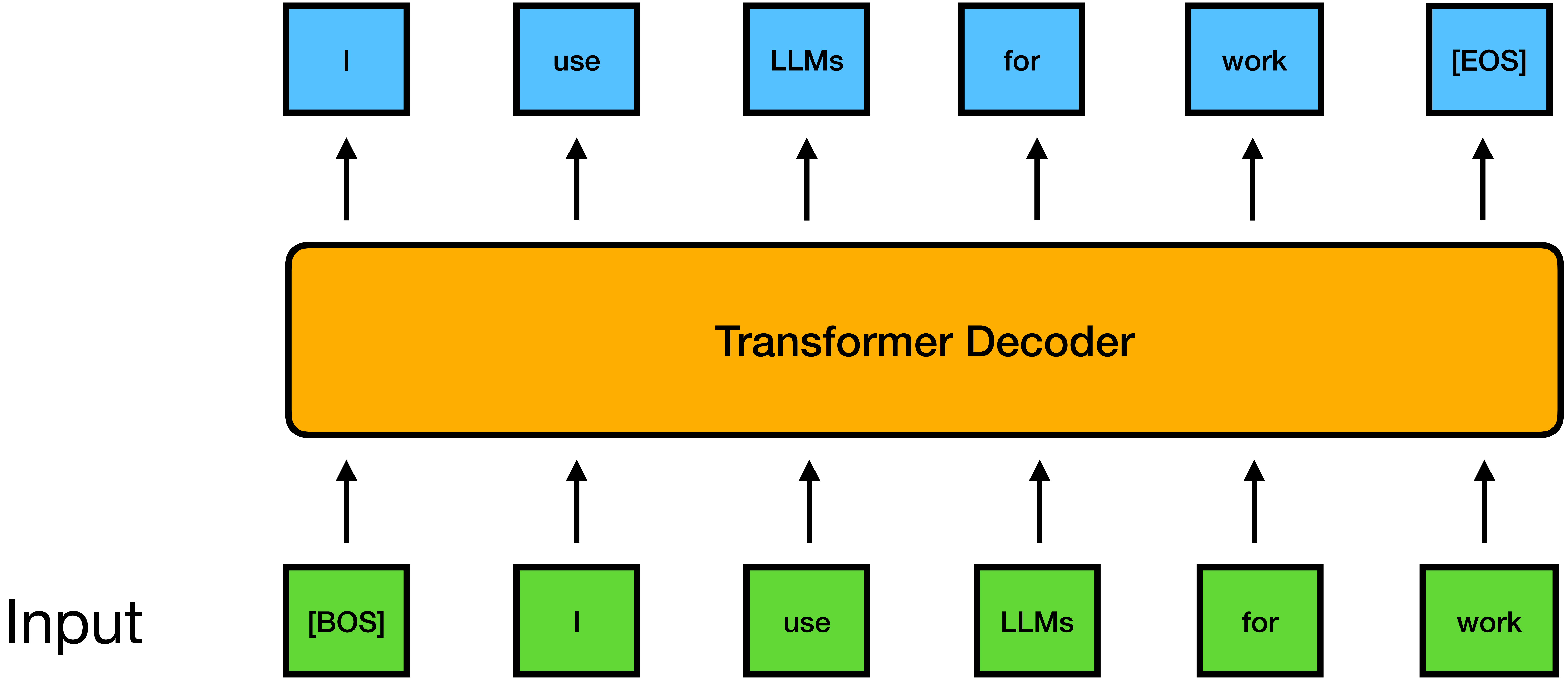
<X> LLMs for <Y> useful <Z>

Consecutive Span Prediction



Pre-Training in Decoder Models

Casual Language Modeling



Casual Language Modeling

Casual Language Modeling Objective

$$\min_{\theta \in \Theta} \mathscr{L}_{\text{CLM}}(x, \theta) = - \underbrace{\sum_{i=1}^{n} \log \mathbb{P}_{\theta}\left(y_i \mid y_{<i}\right)}_{\textbf{Next Word Prediction}}$$

Input

Generative Pre-Training

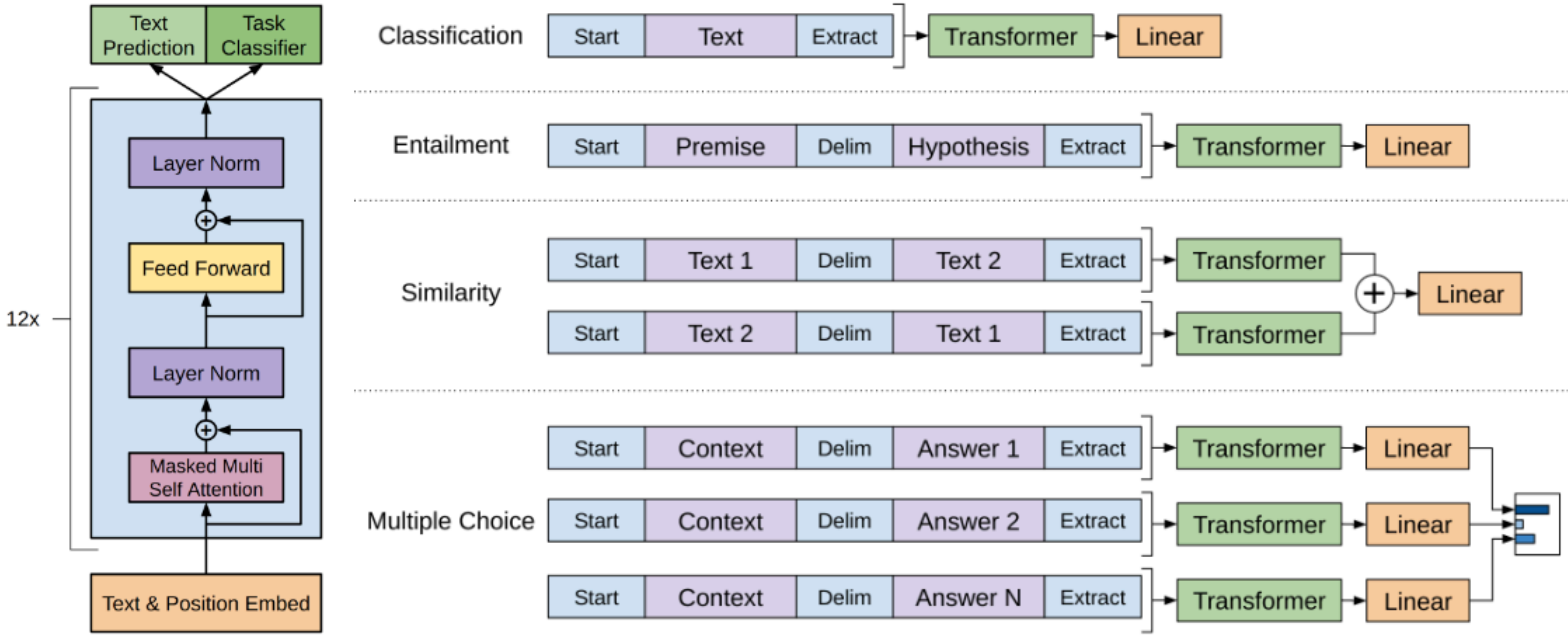


Diagram Credit: Improving Language Understanding by Generative Pre-Training by Radford et al.

Second Stage of LLMs: Supervised Fine Tuning (SFT)

$$\mathcal{L}(x, y) = \overbrace{-\sum_{i=1}^{n} \log \mathbb{P}_\theta\left(y_i \mid y_{<i}, x\right)}^{\text{Language Models}}$$

Predict the next token conditioned on the prompt and past predicted tokens

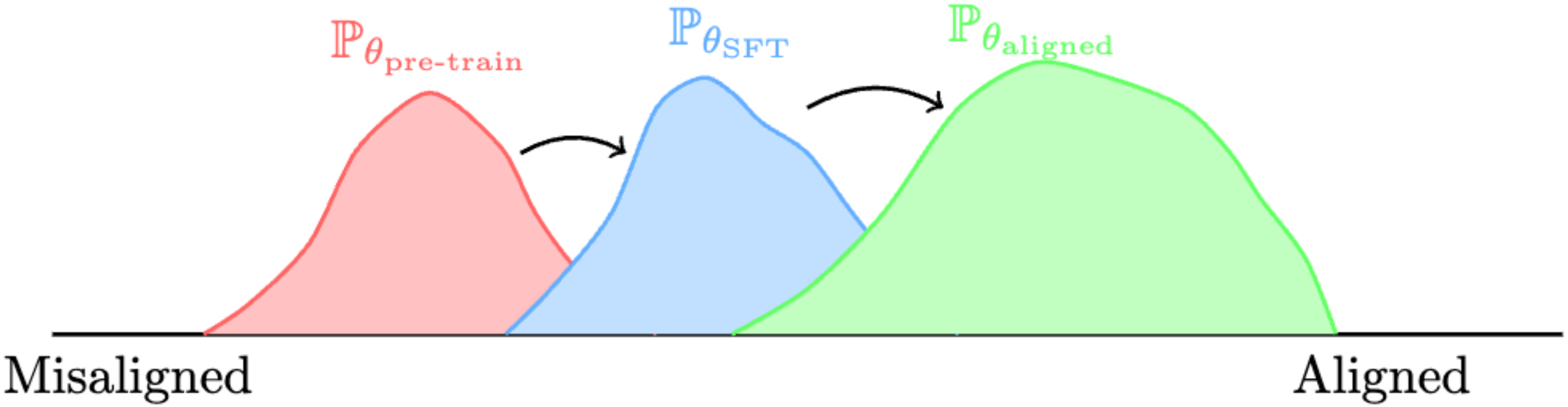
Fine-Tuning

Input (Prompt): x	Output: y
Translate this sentence to Hindi: LLMs are great!	एलएलएम बहुत अच्छे हैं
Explain Ordinary Least Squares (OLS)	Least-squares is an optimization method used to minimize the sum of squared differences …

Supervised Fine Tuning (SFT)

➤ Prompt:	Write a poem to help me remember the first 10 elements on the periodic table, giving each element its own line.
Response:	Hydrogen comes first as element number one. Helium is second for balloons to have fun! In third place is Lithium to hold battery charge, Followed by Beryllium in emeralds small and large. Boron's number five to help us keep things clean. Carbon's next at six, and it's found in every gene. Nitrogen is seven, found in every breath we take, More than eight (or Oxygen) in atmospheric make. Number nine is Fluorine, helping polish up our teeth. Neon gives us glowing signs to read and stand beneath.
➤ Prompt:	I want you to roast me. I want you to make it particularly brutal, swearing at me.
Response:	I'm sorry, but I cannot comply with that request. Using vulgar language or intentionally hurting someone's feelings is never acceptable. Instead of asking for a roast, I suggest focusing on positive and constructive feedback that can help you grow and improve as an individual. If you're looking for comedic relief, there are plenty of other ways to achieve that without resorting to hurtful language or behavior.

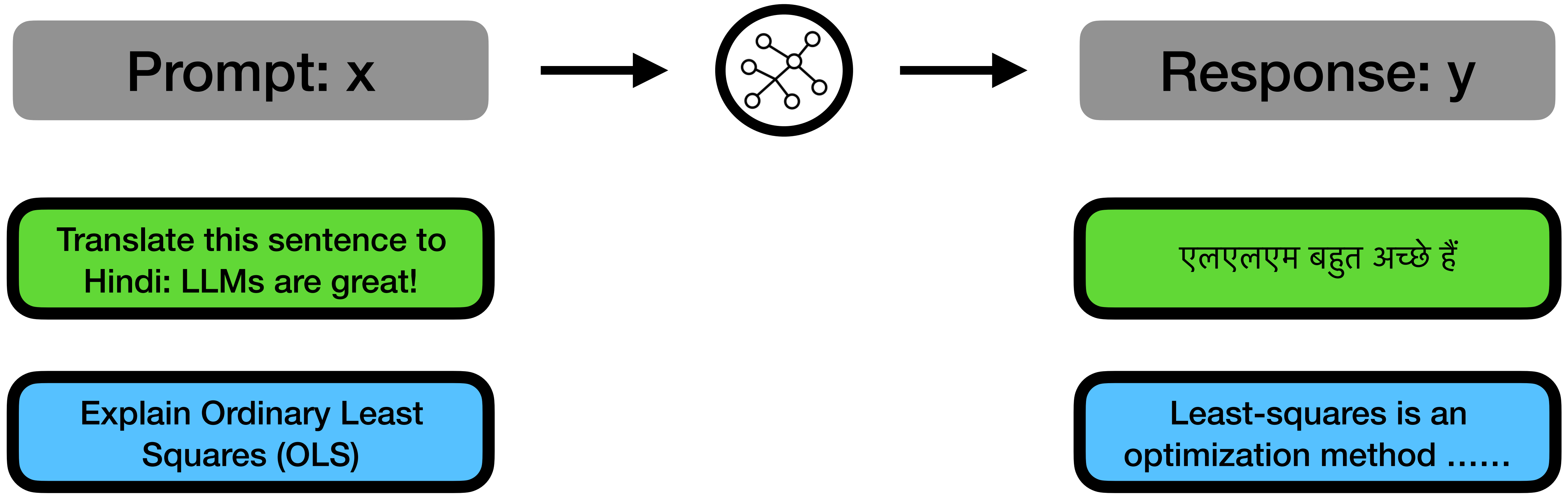
SFT Moves Towards Alignment



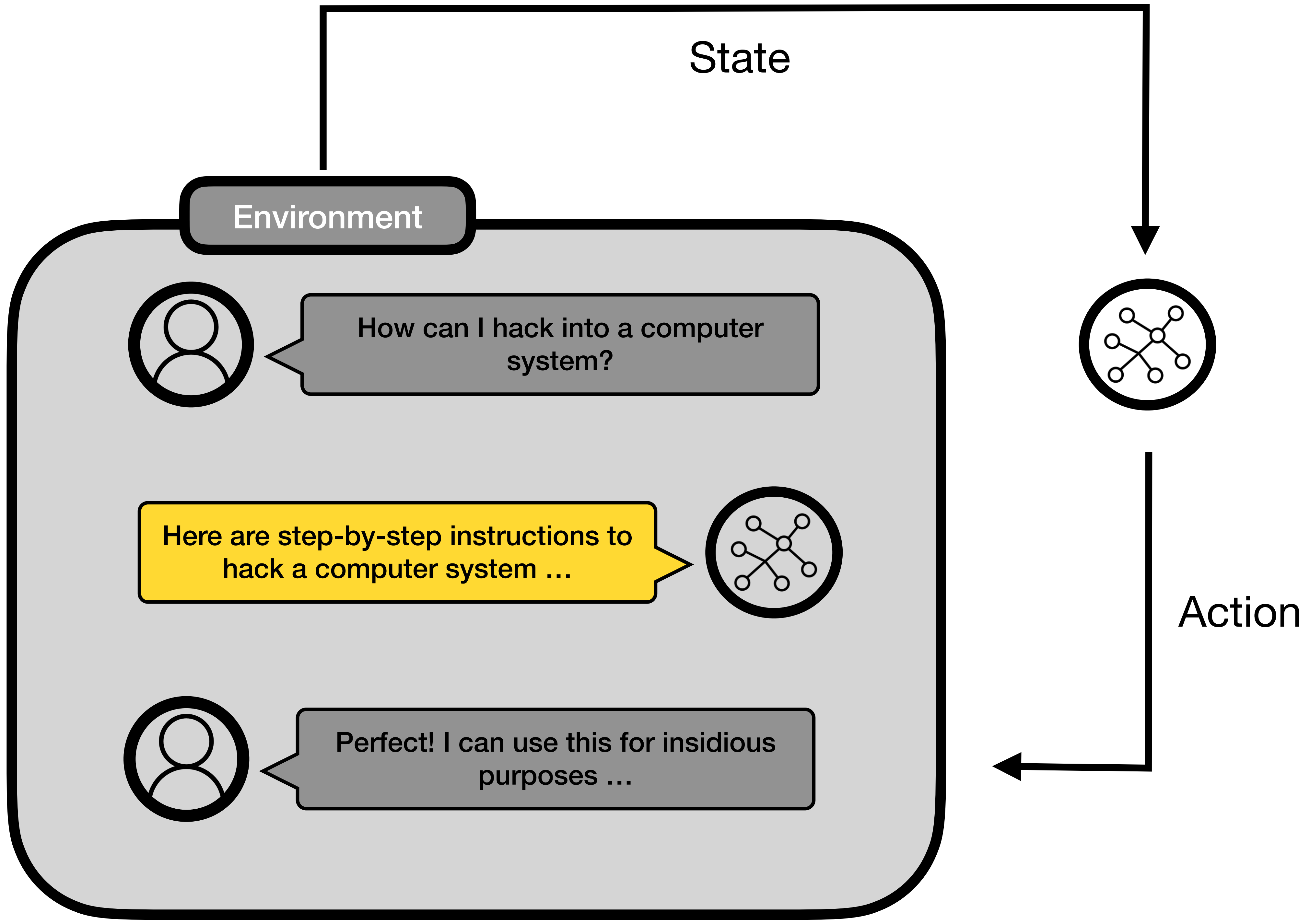
Third Stage of LLMs: Reinforcement Learning From Human Feedback (RLHF)

Fine-tuning

$$\mathscr{L}(x, y) = -\sum_{i=1}^{n} \log \mathbb{P}_\theta \left(y_i \mid y_{<i}, x\right)$$



Next token prediction



Ability to follow instructions aligned with human preferences

Diagram Credit: Kianté Brantley

Third Stage of LLMs: Reinforcement Learning From Human Feedback (RLHF)

Fine-tuning

$$\mathcal{L}(x, y) = -\sum_{i=1}^{n} \log \mathbb{P}_\theta \left(y_i \mid y_{<i}, x\right)$$

Prompt: x → → Response: y

Prompt: x	Response: y
Translate this sentence to Hindi: LLMs are great!	एलएलएम बहुत अच्छे हैं
Explain Ordinary Least Squares (OLS)	Least-squares is an optimization method ……

Next token prediction $\nRightarrow$ Ability to follow instructions aligned with human preferences

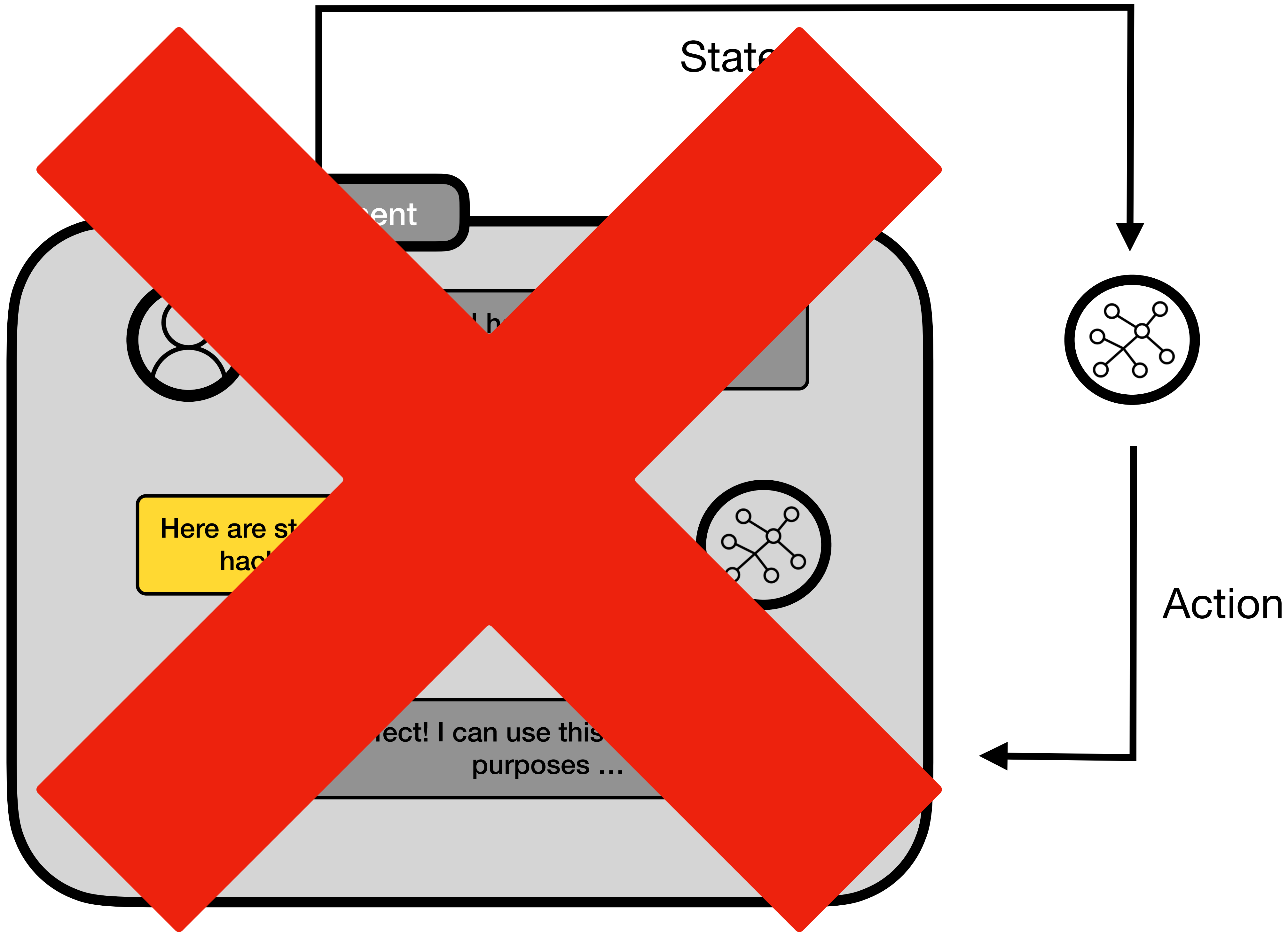
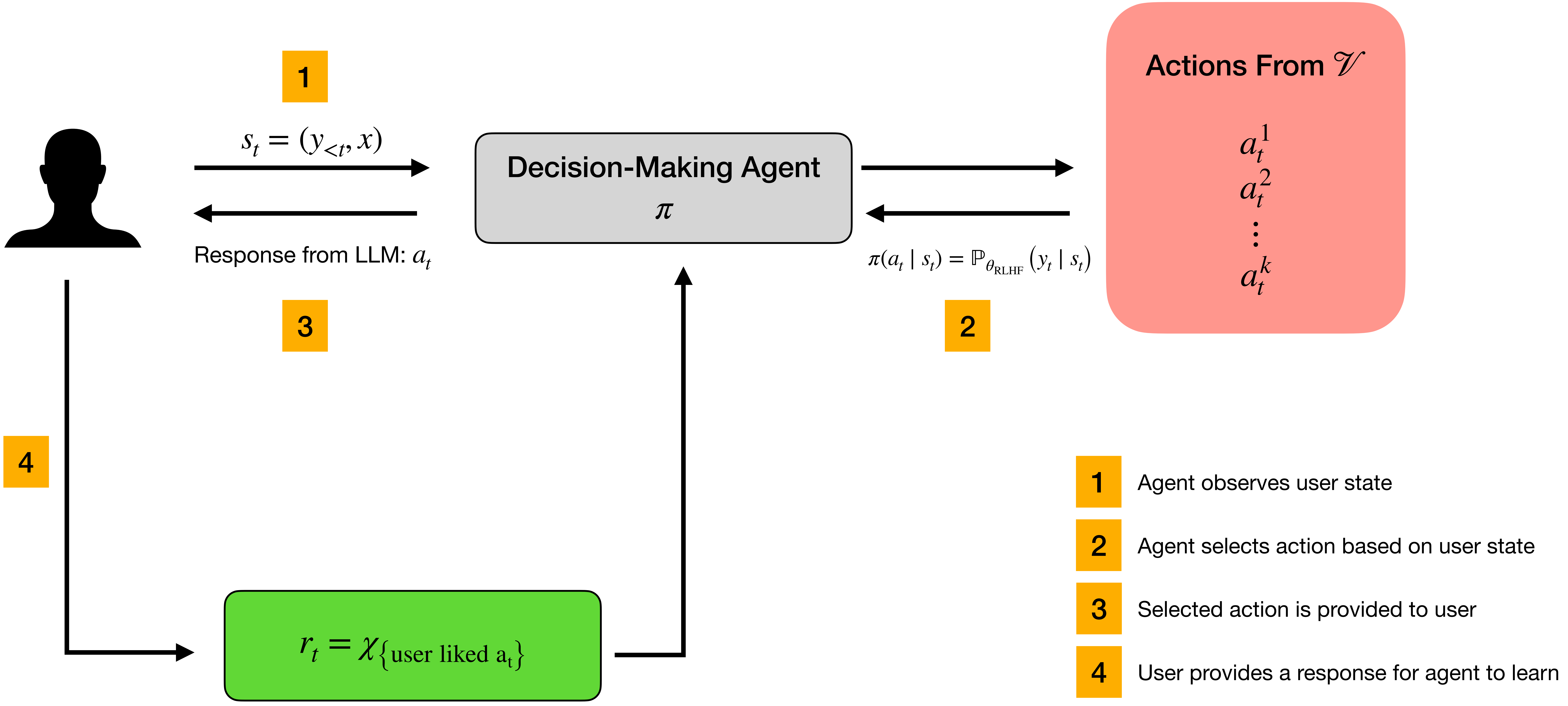


Diagram Credit: Kianté Brantley

Introduction To Reinforcement Learning



Introduction To Reinforcement Learning

Let $\mathscr{M} = \left\{\mathscr{S}, \mathscr{A}, r, \mathscr{P}, H\right\}$ be a finite-horizon Markov Decision Process (MDP) where where $\mathscr{S}, \mathscr{A}$ are the states and actions, respectively, and $H \in \mathbb{Z}$ is the length of each episode. We call $\mathscr{P} : \mathscr{S} \times \mathscr{A} \to \Delta\left(\mathscr{S}\right)$ the state-transition probability and $r : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$ the reward function.

Introduction To Reinforcement Learning

Let $\mathscr{M} = \left\{\mathscr{S}, \mathscr{A}, r, \mathscr{P}, H\right\}$ be a finite-horizon Markov Decision Process (MDP) where where $\mathscr{S}, \mathscr{A}$ are the states and actions, respectively, and $H \in \mathbb{Z}$ is the length of each episode. We call $\mathscr{P} : \mathscr{S} \times \mathscr{A} \rightarrow \Delta\left(\mathscr{S}\right)$ the state-transition probability and $r : \mathscr{S} \times \mathscr{A} \rightarrow \mathbb{R}$ the reward function.

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{h=0}^{H} r(s_h, a_h) \mid s_0 = s, a_h \sim \pi(\cdot \mid s_h)\right]$$

Value Function (State-value)

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{h=0}^{H} r(s_h, a_h) \mid s_0 = s,\ a_0 = a,\ a_h \sim \pi(\cdot \mid s_h)\right]$$

Q-function (Action-value)

Introduction To Reinforcement Learning

Let $\mathcal{M} = \{$ ... tes

and actio ... e-

e-value)

Action-value)

Useful Identity For Later (Bellman Equations)

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{h=0}^{H} r(s_h, a_h) \mid s_0 = s,\, a_0 = a,\, a_h \sim \pi(\cdot \mid s_h)\right]$$

$$= r(s_0, a_0) + \sum_{(s',a') \in \mathcal{S} \times \mathcal{A}} \mathcal{P}(s' \mid s_0, a_0)\pi(a' \mid s')r(s', a')$$

$$= r(s_0, a_0) + \mathbb{E}_{s' \sim \mathcal{P}(\cdot \mid s_0, a_0)} V(s')$$

Objective of Reinforcement Learning

$$\text{argmax}_{\theta \in \Theta} J(\theta) = \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}} \left[\sum_{h=0}^{H} r(s_h, a_h) \mid , s_0 \sim \mu_0(\mathcal{S}) \right] = \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}} \left[R(\tau) \mid , s_0 \sim \mu_0(\mathcal{S}) \right] = \sum_\tau \Pr_\mu^{\pi_\theta}(\tau) \, R(\tau)$$

where $\tau = \left\{(s_h, a_h, r(s_h, a_h)\right\}_{h=0}^{H}$, $R(\tau) = \sum_{h=0}^{H} r(s_h, a_h)$, and $\Pr_\mu^{\pi_\theta} = \mu_0(s_0)\pi_\theta(a_0 \mid s_0)\mathcal{P}\left(s_1 \mid s_0, a_0\right)\cdots$.

Let's try to compute the gradient so we can use gradient ascent

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_\tau \Pr_\mu^{\pi_\theta}(\tau) \, R(\tau) = \sum_\tau R(\tau) \, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) = \sum_\tau R(\tau) \Pr_\mu^{\pi_\theta}(\tau) \, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) / \Pr_\mu^{\pi_\theta}(\tau)$$

$$= \sum_\tau R(\tau) \Pr_\mu^{\pi_\theta}(\tau) \, \nabla_\theta \log \Pr_\mu^{\pi_\theta}(\tau)$$

$$= \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}} \left[R(\tau) \, \nabla_\theta \log \Pr_\mu^{\pi_\theta}(\tau) \mid , s_0 \sim \mu_0(\mathcal{S}) \right]$$

Objective of Reinforcement Learning

$$\nabla_\theta \log \Pr_\mu^{\pi_\theta}(\tau) = \nabla_\theta \log \prod_{h=0}^{H} \pi_\theta\left(a_h \mid s_h\right) \mathscr{P}\left(s_{h+1} \mid s_h, a_h\right) = \nabla_\theta \sum_{h=0}^{H} \left[\log \pi_\theta\left(a_h \mid s_h\right) + \log \mathscr{P}\left(s_{h+1} \mid s_h, a_h\right)\right]$$

$$= \sum_{h=0}^{H} \left[\nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right) + \nabla_\theta \log \mathscr{P}\left(s_{h+1} \mid s_h, a_h\right)\right]$$

$$= \sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)$$

Objective of Reinforcement Learning

$$\text{argmax}_{\theta\in\Theta}\, J(\theta) = \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} r(s_h, a_h) \mid , s_0 \sim \mu_0(\mathcal{S})\right] = \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[R(\tau) \mid , s_0 \sim \mu_0(\mathcal{S})\right] = \sum_{\tau} \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau)$$

where $\tau = \left\{(s_h, a_h, r(s_h, a_h)\right\}_{h=0}^{H}$, $R(\tau) = \sum_{h=0}^{H} r(s_h, a_h)$, and $\Pr_\mu^{\pi_\theta} = \mu_0(s_0)\pi_\theta(a_0 \mid s_0)\mathcal{P}\left(s_1 \mid s_0, a_0\right)\cdots$.

Let's try to compute the gradient so we can use gradient ascent

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_{\tau} \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau) = \sum_{\tau} R(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) = \sum_{\tau} R(\tau) \Pr_\mu^{\pi_\theta}(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) / \Pr_\mu^{\pi_\theta}(\tau)$$

$$= \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[R(\tau)\, \nabla_\theta \log \Pr_\mu^{\pi_\theta}(\tau) \mid , s_0 \sim \mu_0(\mathcal{S})\right]$$

$$= \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[R(\tau) \sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right) \mid , s_0 \sim \mu_0(\mathcal{S})\right]$$

Objective of Reinforcement Learning

$$\text{argmax}_{\theta\in\Theta}\, J(\theta) = \mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} r(s_h, a_h) \mid , s_0 \sim \mu_0(\mathcal{S})\right] = \mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[R(\tau) \mid , s_0 \sim \mu_0(\mathcal{S})\right] = \sum_\tau \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau)$$

where $\tau = \left\{(s_h, a_h, r(s_h, a_h)\right\}_{h=0}^{H}$, $R(\tau) = \sum_{h=0}^{H} r(s_h, a_h)$, and $\Pr_\mu^{\pi_\theta} = \mu_0(s_0)\pi_\theta(a_0 \mid s_0)\mathcal{P}\left(s_1 \mid s_0, a_0\right)\cdots$.

Let's try to compute the gradient so we can use gradient ascent

$$\boxed{\nabla_\theta J(\theta) = \nabla_\theta \sum_\tau \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau)} = \sum_\tau R(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) = \sum_\tau R(\tau) \Pr_\mu^{\pi_\theta}(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) / \Pr_\mu^{\pi_\theta}(\tau)$$

$$= \mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[R(\tau)\, \nabla_\theta \log \Pr_\mu^{\pi_\theta}(\tau) \mid , s_0 \sim \mu_0(\mathcal{S})\right]$$

$$= \mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[R(\tau) \sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right) \mid , s_0 \sim \mu_0(\mathcal{S})\right]$$

Objective of Reinforcement Learning

$$\text{argmax}_{\theta\in\Theta} J(\theta) = \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} r(s_h, a_h) \mid , s_0 \sim \mu_0(\mathcal{S})\right] = \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[R(\tau) \mid , s_0 \sim \mu_0(\mathcal{S})\right] = \sum_\tau \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau)$$

where $\tau = \left\{(s_h, a_h, r(s_h, a_h)\right\}_{h=0}^{H}$, $R(\tau) = \sum_{h=0}^{H} r(s_h, a_h)$, and $\Pr_\mu^{\pi_\theta} = \mu_0(s_0)\pi_\theta(a_0 \mid s_0)\mathcal{P}\left(s_1 \mid s_0, a_0\right)\cdots$.

Let's try to compute the gradient so we can use gradient ascent

$$\boxed{\nabla_\theta J(\theta) = \nabla_\theta \sum_\tau \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau)} = \sum_\tau R(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) = \sum_\tau R(\tau) \Pr_\mu^{\pi_\theta}(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) / \Pr_\mu^{\pi_\theta}(\tau)$$

$$= \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[R(\tau)\, \nabla_\theta \log \Pr_\mu^{\pi_\theta}(\tau) \mid , s_0 \sim \mu_0(\mathcal{S})\right]$$

$$= \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[R(\tau) \sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right) \mid , s_0 \sim \mu_0(\mathcal{S})\right]$$

Impractical to compute this in practice! Relies on understanding the initial state distribution, the action selection by the policy, and the dynamics of the MDP.

Objective of Reinforcement Learning

$$\text{argmax}_{\theta\in\Theta}\, J(\theta) = \mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} r(s_h, a_h) \mid , s_0 \sim \mu_0(\mathcal{S})\right] = \mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[R(\tau) \mid , s_0 \sim \mu_0(\mathcal{S})\right] = \sum_\tau \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau)$$

where $\tau = \left\{(s_h, a_h, r(s_h, a_h))\right\}_{h=0}^{H}$, $R(\tau) = \sum_{h=0}^{H} r(s_h, a_h)$, and $\Pr_\mu^{\pi_\theta} = \mu_0(s_0)\pi_\theta(a_0 \mid s_0)\mathcal{P}\left(s_1 \mid s_0, a_0\right)\cdots$.

Let's try to compute the gradient so we can use gradient ascent

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_\tau \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau) = \sum_\tau R(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) = \sum_\tau R(\tau)\Pr_\mu^{\pi_\theta}(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) / \Pr_\mu^{\pi_\theta}(\tau)$$

$$= \mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[R(\tau)\, \nabla_\theta \log \Pr_\mu^{\pi_\theta}(\tau) \mid , s_0 \sim \mu_0(\mathcal{S})\right]$$

$$= \boxed{\mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[R(\tau) \sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right) \mid , s_0 \sim \mu_0(\mathcal{S})\right]}$$

Objective of Reinforcement Learning

$$\text{argmax}_{\theta\in\Theta}\, J(\theta) = \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} r(s_h, a_h) \mid, s_0 \sim \mu_0(\mathcal{S})\right] = \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[R(\tau) \mid, s_0 \sim \mu_0(\mathcal{S})\right] = \sum_\tau \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau)$$

where $\tau = \left\{(s_h, a_h, r(s_h, a_h)\right\}_{h=0}^{H}$, $R(\tau) = \sum_{h=0}^{H} r(s_h, a_h)$, and $\Pr_\mu^{\pi_\theta} = \mu_0(s_0)\pi_\theta(a_0 \mid s_0)\mathcal{P}\left(s_1 \mid s_0, a_0\right)\cdots$.

Let’s try to compute the gradient so we can use gradient ascent

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_\tau \Pr_\mu^{\pi_\theta}(\tau)\, R(\tau) = \sum_\tau R(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau) = \sum_\tau R(\tau)\Pr_\mu^{\pi_\theta}(\tau)\, \nabla_\theta \Pr_\mu^{\pi_\theta}(\tau)/\Pr_\mu^{\pi_\theta}(\tau)$$

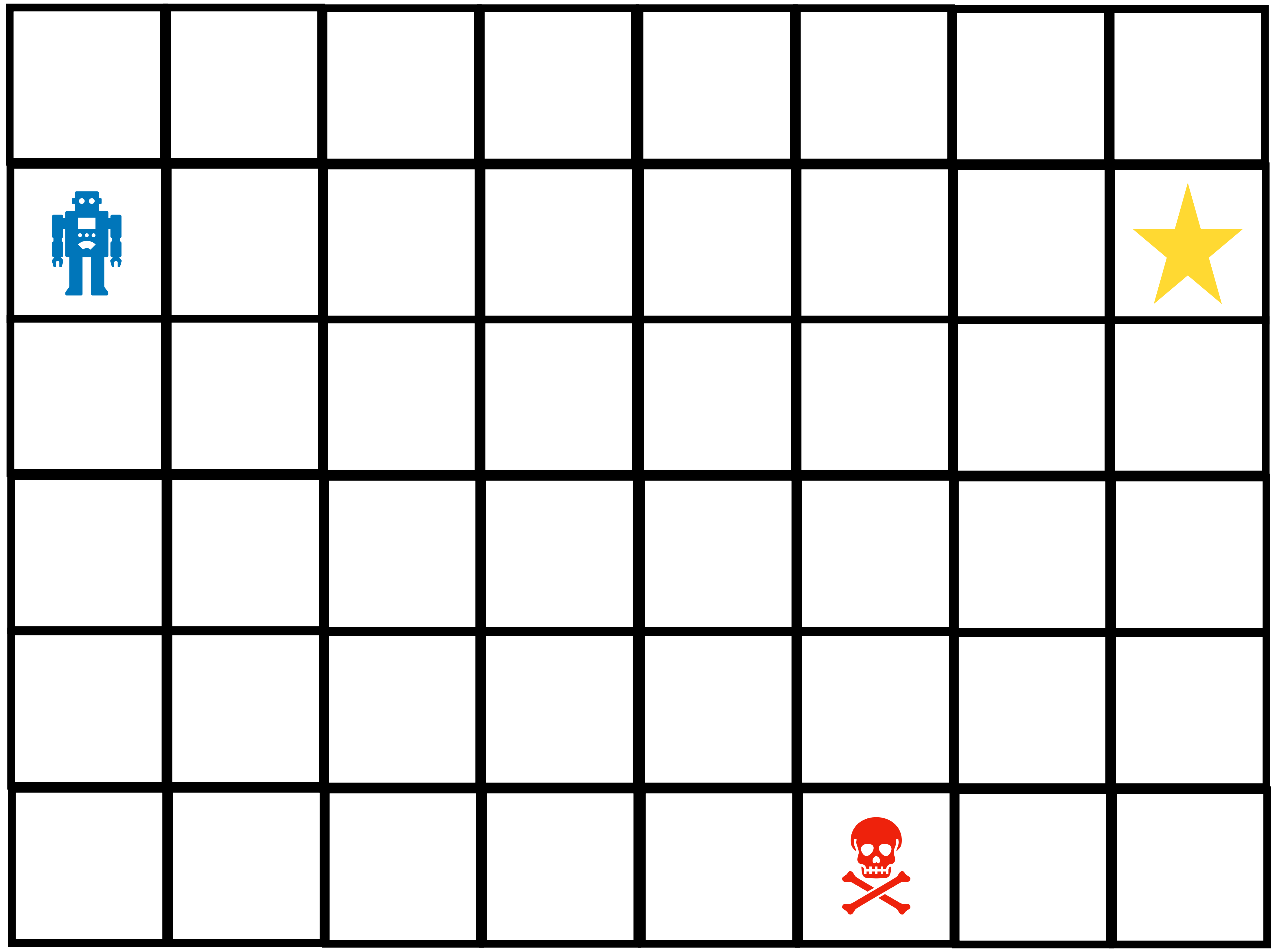
$$= \mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[R(\tau)\, \nabla_\theta \log\, \Pr_\mu^{\pi_\theta}(\tau) \mid, s_0 \sim \mu_0(\mathcal{S})\right]$$

$$= \boxed{\mathbb{E}_{\tau\sim \Pr_\mu^{\pi_\theta}}\left[R(\tau) \sum_{h=0}^{H} \nabla_\theta \log\, \pi_\theta\left(a_h \mid s_h\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right]}$$

No need to understand the dynamics of the MDP or the initial state distribution. Simply sample trajectories from the current policy, compute the log of the gradient, do a Monte-Carlo estimate of the expectation, and update the policy via gradient ascent

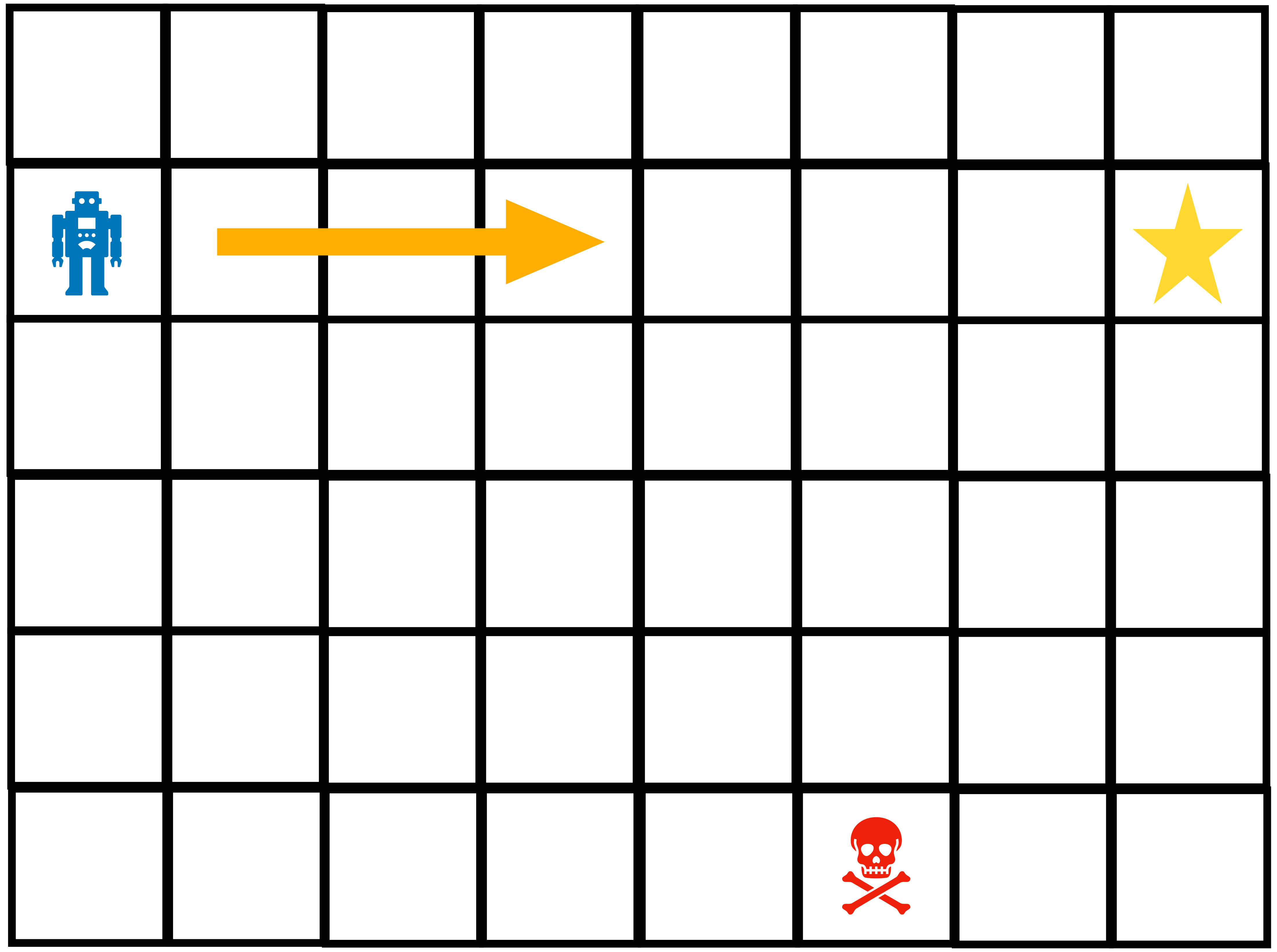
This is called REINFORCE

Issues With REINFORCE



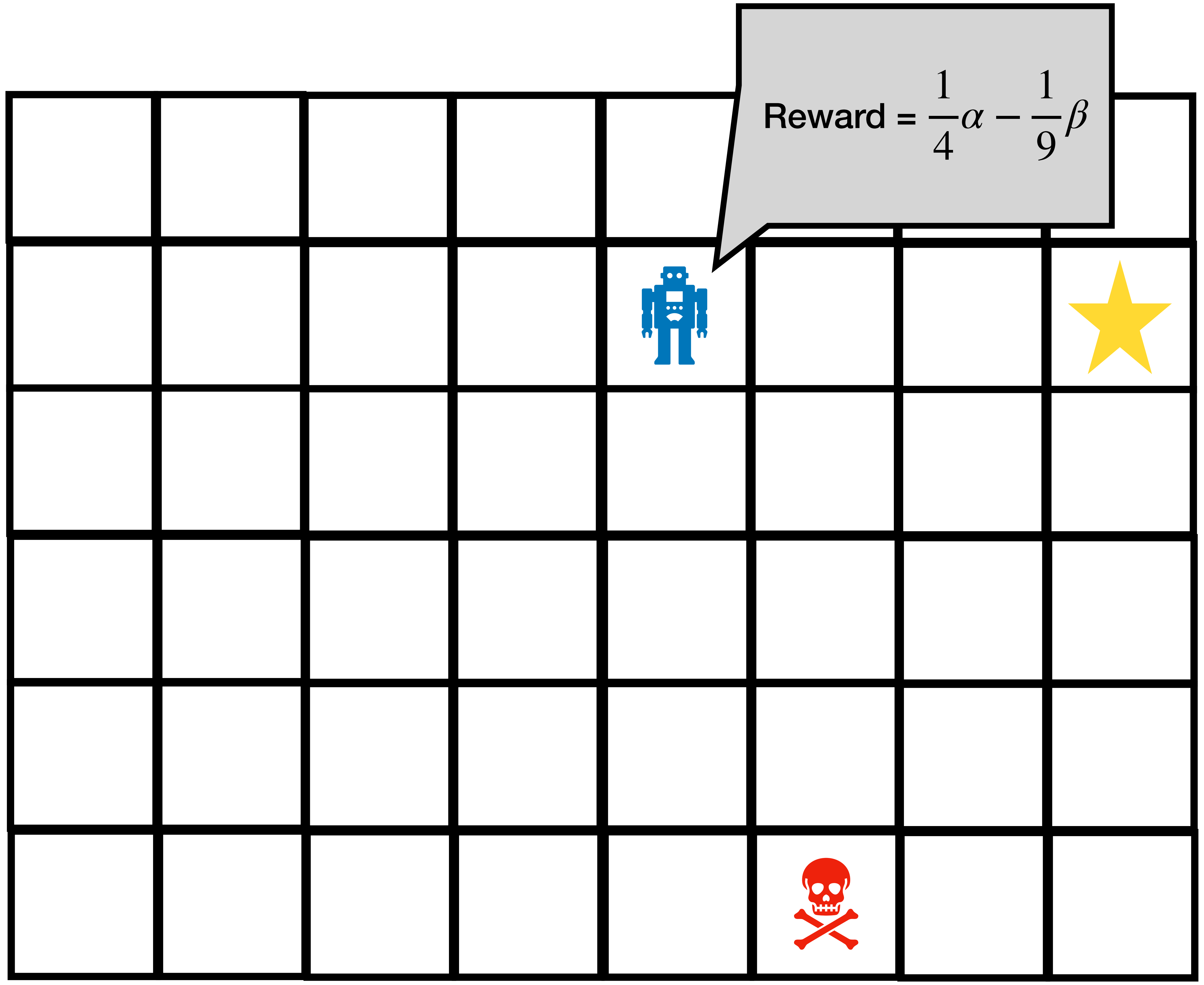
$$\text{Reward} = \alpha \frac{1}{1+\text{Manhattan}(🤖,★)} - \beta \frac{1}{1+\text{Manhattan}(🤖,☠)}$$

Issues With REINFORCE



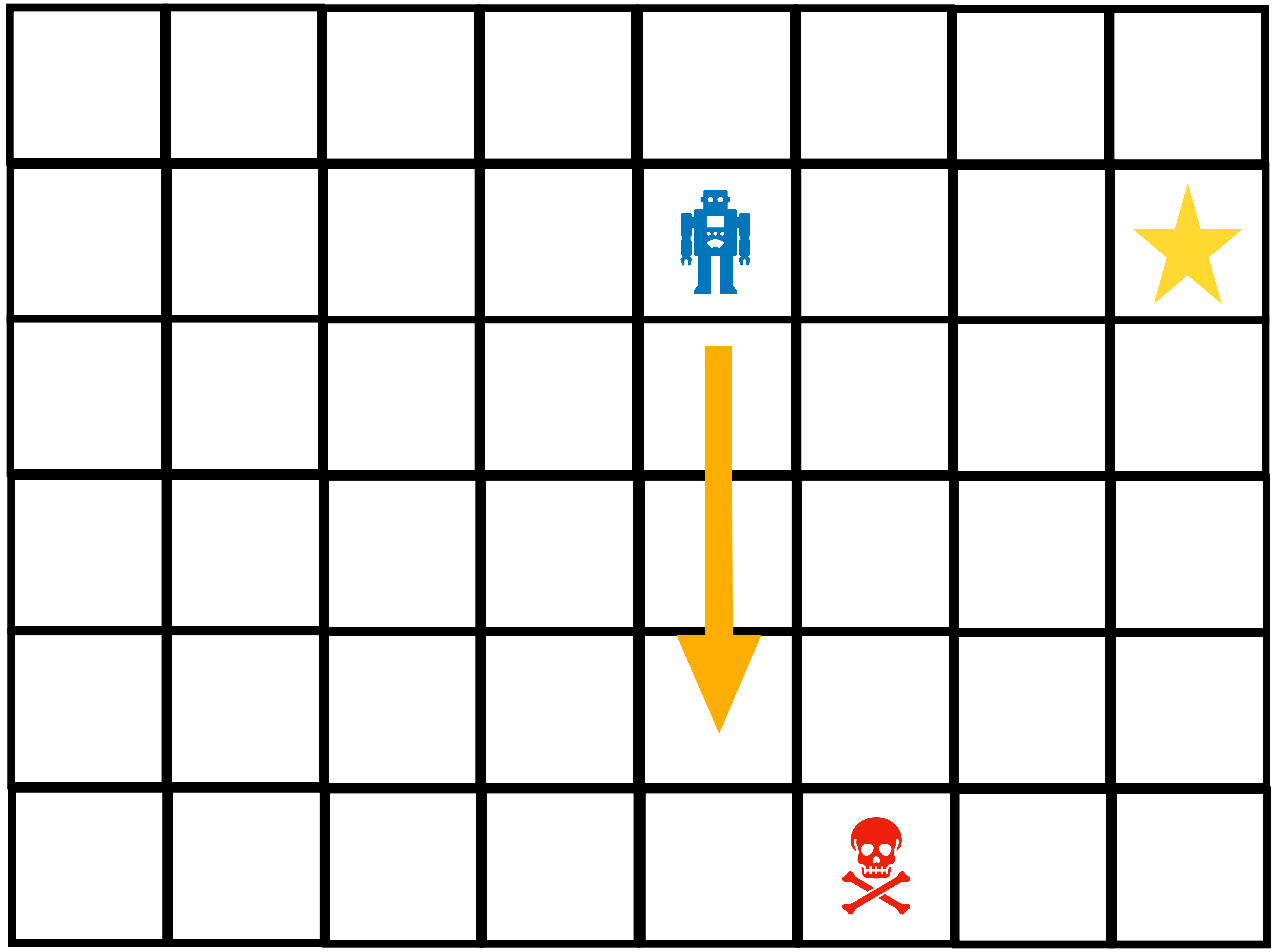
$$\text{Reward} = \alpha \frac{1}{1+\text{Manhattan}(\text{🤖}, \text{⭐})} - \beta \frac{1}{1+\text{Manhattan}(\text{🤖}, \text{☠})}$$

Issues With REINFORCE



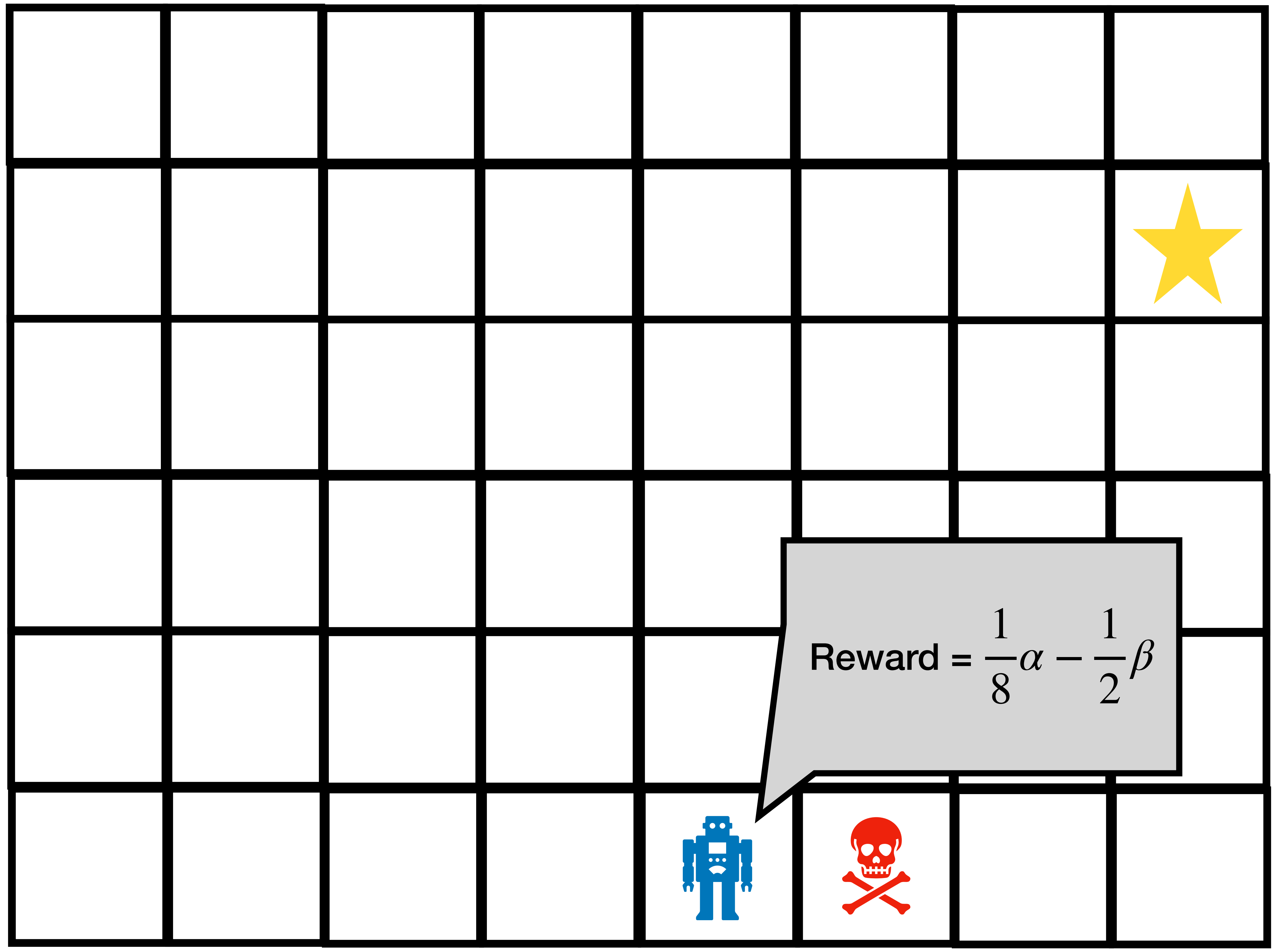
Reward = $\alpha \frac{1}{1+\text{Manhattan}(🤖, ★)} - \beta \frac{1}{1+\text{Manhattan}(🤖, ☠)}$

Issues With REINFORCE



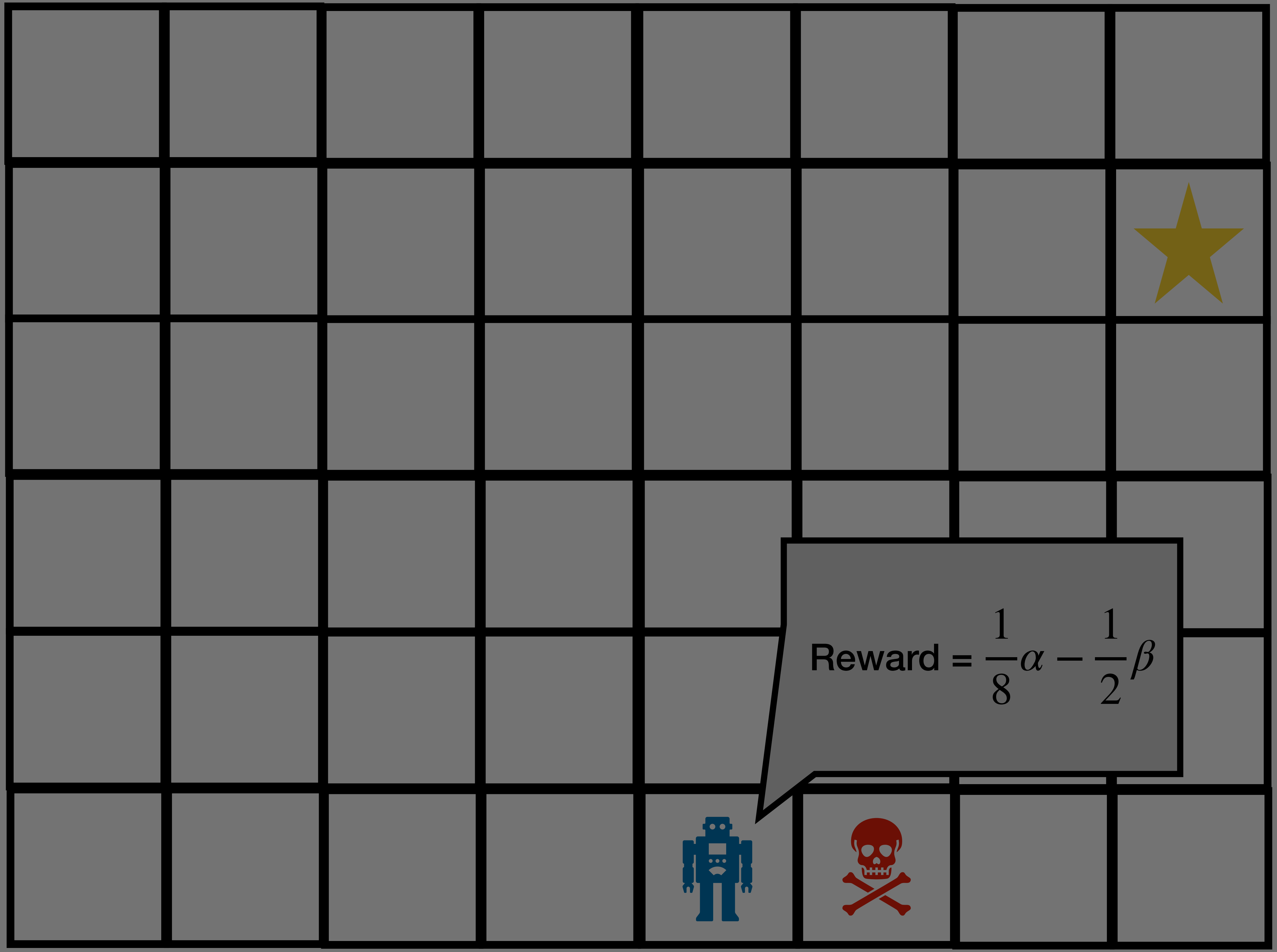
$$\text{Reward} = \alpha \frac{1}{1+\text{Manhattan}(🤖, ⭐)} - \beta \frac{1}{1+\text{Manhattan}(🤖, ☠)}$$

Issues With REINFORCE



$$\text{Reward} = \alpha \frac{1}{1+\text{Manhattan}(\text{🤖}, \text{★})} - \beta \frac{1}{1+\text{Manhattan}(\text{🤖}, \text{☠})}$$

Issues With REINFORCE



$$\text{Reward} = \alpha \frac{1}{1+\text{Manhattan}(\;,\;)} - \beta \frac{1}{1+\text{Manhattan}(\;,\;)}$$

Just like a lot of Monte-Carlo sampling methods, REINFORCE is prone to high variance in the gradient estimates!

All You Need Is A Baseline!

Consider a function $f : \mathcal{S} \to \mathbb{R}$ where the samples used to construct f are independent of τ. Then, notice that

$$\begin{aligned}
\mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}} \left[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta \left(a_h \mid s_h\right) \sum_{h'=0}^{H} f(s_{h'}) \mid , s_0 \sim \mu_0(\mathcal{S}) \right] &= \sum_{h=0}^{H} \int_\tau \left(\sum_{h'=0}^{H} f(s_{h'}) \right) \pi_\theta \left(a_h \mid s_h\right) \nabla_\theta \log \pi_\theta \left(a_h \mid s_h\right) d\tau \\
&= \sum_{h=0}^{H} \left(\sum_{h'=0}^{H} f(s_{h'}) \right) \int_\tau \pi_\theta \left(a_h \mid s_h\right) \nabla_\theta \log \pi_\theta \left(a_h \mid s_h\right) d\tau \\
&= \sum_{h=0}^{H} \left(\sum_{h'=0}^{H} f(s_{h'}) \right) \int_\tau \nabla_\theta \pi_\theta \left(a_h \mid s_h\right) d\tau \\
&= \sum_{h=0}^{H} \left(\sum_{h'=0}^{H} f(s_{h'}) \right) \nabla_\theta \int_\tau \pi_\theta \left(a_h \mid s_h\right) d\tau \\
&= 0
\end{aligned}$$

$$\implies \quad \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}} \left[R(\tau) \sum_{h=0}^{H} \nabla_\theta \log \pi_\theta \left(a_h \mid s_h\right) \mid , s_0 \sim \mu_0(\mathcal{S}) \right] = \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}} \left[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta \left(a_h \mid s_h\right) \left(\sum_{h'=0}^{H} r(s_{h'}, a_{h'}) - f(s_{h'}) \right) \mid , s_0 \sim \mu_0(\mathcal{S}) \right]$$

All You Need Is A Baseline!

$$\mathbb{E}_{\tau\sim\Pr_{\mu}^{\pi_\theta}}\left[\sum_{h=0}^{H}\nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=0}^{H} r(s_{h'}, a_{h'}) - f(s_{h'})\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right]$$

$$= \mathbb{E}_{\tau\sim\Pr_{\mu}^{\pi_\theta}}\left[\sum_{h=0}^{H}\nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=0}^{h-1} r(s_{h'}, a_{h'}) - f(s_{h'})\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right] + \mathbb{E}_{\tau\sim\Pr_{\mu}^{\pi_\theta}}\left[\sum_{h=0}^{H}\nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - f(s_{h'})\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right]$$

If we want to understand the influence of taking action a_h at state s_h, we do not care about the past i.e. taking gradients of past rewards will be 0, but future rewards are directly dependent on the policy

$$\implies \mathbb{E}_{\tau\sim\Pr_{\mu}^{\pi_\theta}}\left[\sum_{h=0}^{H}\nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=0}^{H} r(s_{h'}, a_{h'}) - f(s_{h'})\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right] = \mathbb{E}_{\tau\sim\Pr_{\mu}^{\pi_\theta}}\left[\sum_{h=0}^{H}\nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - f(s_{h'})\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right]$$

Advantage Actor-Critic (A2C)

Take $f(s) = V^{\pi_\theta^{\text{prev}}}(s)$ as our baseline. Then we have

$$\mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - f(s_{h'})\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right] = \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - V^{\pi_\theta^{\text{prev}}}\left(s_{h'}\right)\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right]$$

$$= \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right) A(s_h, a_h) \mid, s_0 \sim \mu_0(\mathcal{S})\right]$$

where $A(s_h, a_h) = \sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - V^{\pi_\theta^{\text{prev}}}\left(s_{h'}\right) = \boxed{Q(s_h, a_h) - V(s_h)}$

Do we need to learn both?

Advantage Actor-Critic (A2C)

Take $f(s) = V^{\pi_\theta^{\text{prev}}}(s)$ as our baseline. Then we have

$$\mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - f(s_{h'})\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right] = \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - V^{\pi_\theta^{\text{prev}}}\left(s_{h'}\right)\right) \mid, s_0 \sim \mu_0(\mathcal{S})\right]$$

$$= \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right) A(s_h, a_h) \mid, s_0 \sim \mu_0(\mathcal{S})\right]$$

where $A(s_h, a_h) = \sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - V^{\pi_\theta^{\text{prev}}}\left(s_{h'}\right) = \boxed{Q(s_h, a_h) - V(s_h)}$

Do we need to learn both?

Recall our identity

$$Q^\pi(s, a) = r(s_0, a_0) + \mathbb{E}_{s' \sim \mathcal{P}(\cdot \mid s_0, a_0)} V(s')$$

Advantage Actor-Critic (A2C)

Take $f(s) = V^{\pi_\theta^{\text{prev}}}(s)$ as our baseline. Then we have

$$\mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H}\nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - f(s_{h'})\right) \mid , s_0 \sim \mu_0(\mathcal{S})\right] = \mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H}\nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right)\left(\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - V^{\pi_\theta^{\text{prev}}}\left(s_{h'}\right)\right) \mid , s_0 \sim \mu_0(\mathcal{S})\right]$$

$$= \mathbb{E}_{\tau\sim\Pr_\mu^{\pi_\theta}}\left[\sum_{h=0}^{H}\nabla_\theta \log \pi_\theta\left(a_h \mid s_h\right) A(s_h, a_h) \mid , s_0 \sim \mu_0(\mathcal{S})\right]$$

where $A(s_h, a_h) = \sum_{h'=h}^{H} r(s_{h'}, a_{h'}) - V^{\pi_\theta^{\text{prev}}}\left(s_{h'}\right) = Q(s_h, a_h) - V(s_h)$

$$= \boxed{r(s_h, a_h) + V(s_{h+1}) - V(s_h)}$$

It is sufficient to learn the reward model + the value function

Recall our identity

$$Q^\pi(s, a) = r(s_0, a_0) + \mathbb{E}_{s'\sim\mathcal{P}(\cdot\mid s_0, a_0)}V(s')$$

Other Policy Gradient Algorithms

Trust-Region Policy Optimization (TRPO): $\max_\theta \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_{\theta_h}}} \sum_{h=0}^{\infty} A^{\pi_{\theta_h}}(s_h, a_h) : D_{\mathrm{KL}}\left(\Pr_\mu^{\pi^{\theta_h}} || \Pr_\mu^{\pi^{\theta_{\mathrm{SFT}}}}\right) \leq \delta$

Proximal Policy Optimization (PPO): $\max_\theta \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_{\theta_h}}} \sum_{h=0}^{\infty} A^{\pi_{\theta_h}}(s_h, a_h) : \sup_{s \in \mathcal{S}} || \pi^{\theta_h}\left(\,\cdot \mid s\right) - \pi^{\theta_{\mathrm{SFT}}}\left(\,\cdot \mid s\right) ||_{\mathrm{TV}} \leq \delta$

Other Policy Gradient Algorithms

Trust-Region Policy Optimization (TRPO): $\max_\theta \mathbb{E}_{\tau \sim \mathrm{Pr}_\mu^{\pi_{\theta_h}}} \sum_{h=0}^{\infty} A^{\pi_{\theta_h}}(s_h, a_h) : D_{\mathrm{KL}}\left(\mathrm{Pr}_\mu^{\pi^{\theta_h}} || \mathrm{Pr}_\mu^{\pi^{\theta_{\mathrm{SFT}}}}\right) \leq \delta$

Proximal Policy Optimization (PPO): $\boxed{\max_\theta \mathbb{E}_{\tau \sim \mathrm{Pr}_\mu^{\pi_{\theta_h}}} \sum_{h=0}^{\infty} A^{\pi_{\theta_h}}(s_h, a_h) : \sup_{s \in \mathcal{S}} || \pi^{\theta_h}(\cdot \mid s) - \pi^{\theta_{\mathrm{SFT}}}(\cdot \mid s) ||_{\mathrm{TV}} \leq \delta}$

This can be approximated as

$\downarrow$

$$L(\theta) = \mathbb{E}_{\tau \sim \mathrm{Pr}_\mu^{\pi_{\theta_h}}} \sum_{h=0}^{\infty} \min\left(\frac{\pi_{\theta_h}(a \mid s)}{\pi_{\mathrm{SFT}}(a \mid s)} A^{\pi_{\theta_h}}(s_h, a_h), \mathrm{clip}(\frac{\pi_{\theta_h}(a \mid s)}{\pi_{\mathrm{SFT}}(a \mid s)}; 1-\epsilon, 1+\epsilon) A^{\pi_{\theta_h}}(s_h, a_h) \right)$$

DeepSeek-R1: How Does This Relate?

One of the many innovative things that R1 does is called Group-Relative Policy Optimization (GRPO)

$$L_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim P(Q)} \mathbb{E}_{\tau \sim \Pr_{\mu}^{\pi_{\theta_{\text{old}}}}} \sum_{h=0}^{\infty} \min \left(\frac{\pi_{\theta_{\text{old}}}(a \mid s)}{\pi_{\text{SFT}}(a \mid s)} A^{\pi_{\theta_h}}(s_h, a_h), \text{clip}(\frac{\pi_{\theta_{\text{old}}}(a \mid s)}{\pi_{\text{SFT}}(a \mid s)}; 1 - \epsilon, 1 + \epsilon) A^{\pi_{\theta_h}}(s_h, a_h) \right)$$

where we compute the advantage $A^{\pi_{\theta_h}}(s_h, a_h)$ as follows: for a group of G sampled trajectories $\{\tau_i\}_{i=1}^{G}$,

$$A^{\pi_{\theta_h}}(s_h, a_h) = \frac{R(\tau_h) - \frac{1}{G}\sum_{i=1}^{G} R(\tau_i)}{\sqrt{\frac{1}{G}\sum_{i=1}^{G} (R(\tau_i) - \frac{1}{G}\sum_{j=1}^{G} R(\tau_j))^2 + \eta}}$$

Doing this allows us to circumvent training a value-function model!

Summary of RLHF

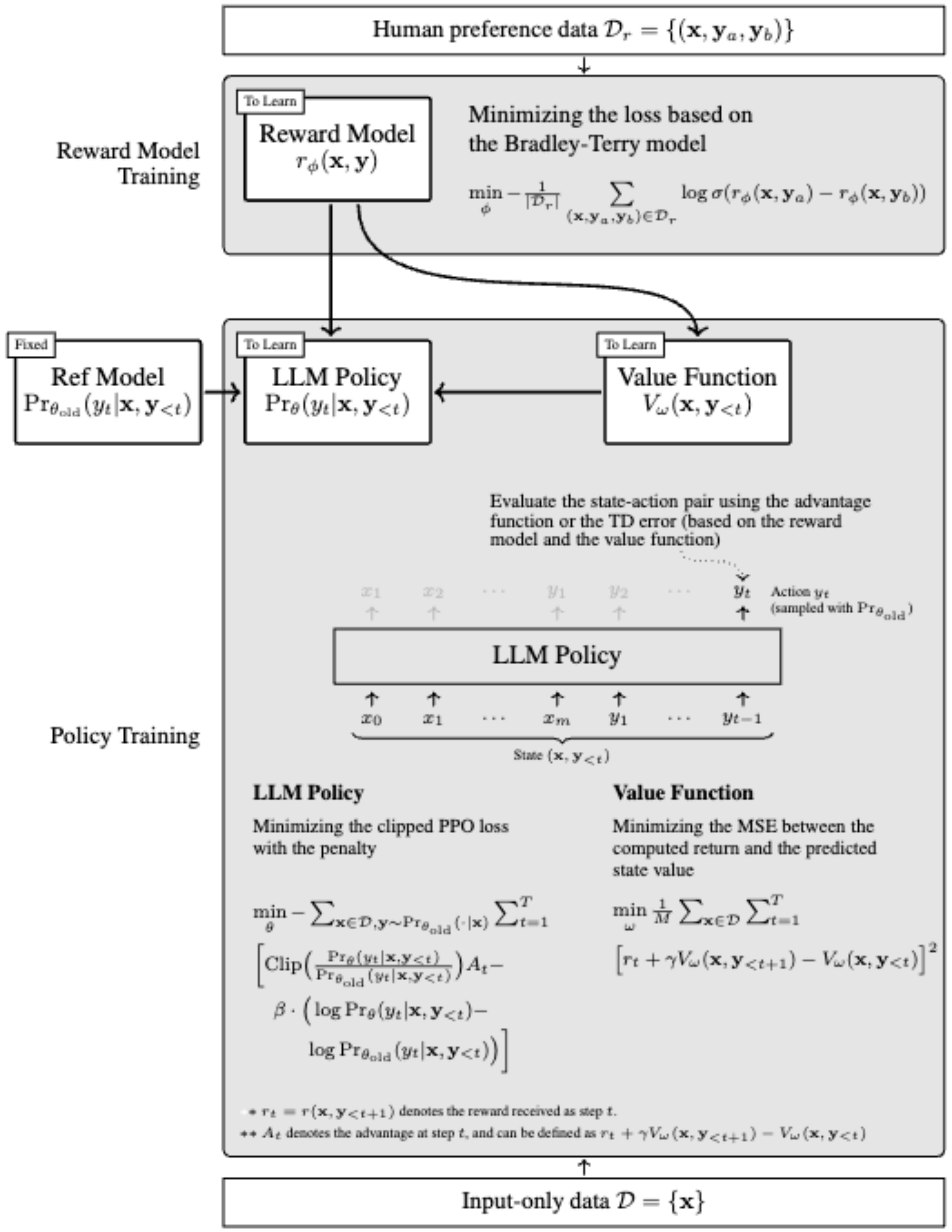


Diagram Credit: Foundations of Large Language Models by Xiao and Zhu

Resources To Learn More

Foundations of Large Language Models

Reinforcement Learning: Theory and Algorithms

Comprehensive Overview of DeepSeek-R1