Contrastive RL

By Kellen Kanarios

EMERGENCY INTERRUPTION: Deepseek R1 Released

	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (Pass@1)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	**92.3**
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	**96.6**	96.3
	Codeforces (Rating)	717	759	1134	1820	**2061**	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	**79.8**
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	**97.3**
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

Background: From Language to Numbers

Only important thing to understand here is **dimension**.

1. Given $\mathbf{x}$ = "the fish lived in the sea" split up into tokens with number

$\mathbf{x}$ = "the fish lived in the sea"

the	fish	lived	in	the sea
132	17	97	83	184

2. "Tokenized" $\mathbf{x} \in \mathbb{R}^n$ then embedded by learned word embedding $f_\theta : \mathbb{R}^n \to \mathbb{R}^{n \times d}$

$$\mathrm{Tok}(\mathbf{x}) = \begin{bmatrix} 132 \\ 17 \\ 87 \\ 83 \\ 184 \end{bmatrix} \xrightarrow{f_\theta(\mathbf{x})} \underbrace{\begin{bmatrix} \overbrace{\begin{bmatrix} 0.723 & \cdots & 4.291 \end{bmatrix}}^{\mathbf{x}^{(1)}} \\ \begin{bmatrix} 2.156 & \cdots & -1.843 \end{bmatrix} \\ \begin{bmatrix} -0.537 & \cdots & 3.102 \end{bmatrix} \\ \begin{bmatrix} 1.994 & \cdots & 0.618 \end{bmatrix} \\ \begin{bmatrix} 5.281 & \cdots & -2.475 \end{bmatrix} \end{bmatrix}}_{\in \mathbb{R}^d}$$

Background: From Language to Numbers

2. "Tokenized" $\mathbf{x} \in \mathbb{R}^n$ then embedded by learned word embedding $f_\theta : \mathbb{R}^n \to \mathbb{R}^{n \times d}$

$$\mathrm{Tok}(\mathbf{x}) = \begin{bmatrix} 132 \\ 17 \\ 87 \\ 83 \\ 184 \end{bmatrix} \xrightarrow{f_\theta(\mathbf{x})} \begin{bmatrix} \overbrace{\begin{bmatrix} 0.723 & \cdots & 4.291 \end{bmatrix}}^{\mathbf{x}^{(1)}} \\ \begin{bmatrix} 2.156 & \cdots & -1.843 \end{bmatrix} \\ \begin{bmatrix} -0.537 & \cdots & 3.102 \end{bmatrix} \\ \begin{bmatrix} 1.994 & \cdots & 0.618 \end{bmatrix} \\ \underbrace{\begin{bmatrix} 5.281 & \cdots & -2.475 \end{bmatrix}}_{\in \mathbb{R}^d} \end{bmatrix}$$

Embeddings have **semantic meaning** i.e. in **word2vec**:

$$\mathbf{v}_{\text{king}} - \mathbf{v}_{\text{man}} + \mathbf{v}_{\text{woman}} = \mathbf{v}_{\text{queen}}$$

Technicality: **positional encodings**

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean; *Efficient Estimation of Word Representations in Vector Space.*

Self Attention: What we want (from 3b1b)

American shrew $\underbrace{\text{mole}}_{\begin{bmatrix} 2.3 \\ 1.7 \\ \vdots \\ 7.9 \\ 3.6 \end{bmatrix}}$

One $\underbrace{\text{mole}}_{\begin{bmatrix} 2.3 \\ 1.7 \\ \vdots \\ 7.9 \\ 3.6 \end{bmatrix}}$ of carbon dioxide

Take a biopsy of the $\underbrace{\text{mole}}_{\begin{bmatrix} 2.3 \\ 1.7 \\ \vdots \\ 7.9 \\ 3.6 \end{bmatrix}}$

Self Attention: What we want (from 3b1b)

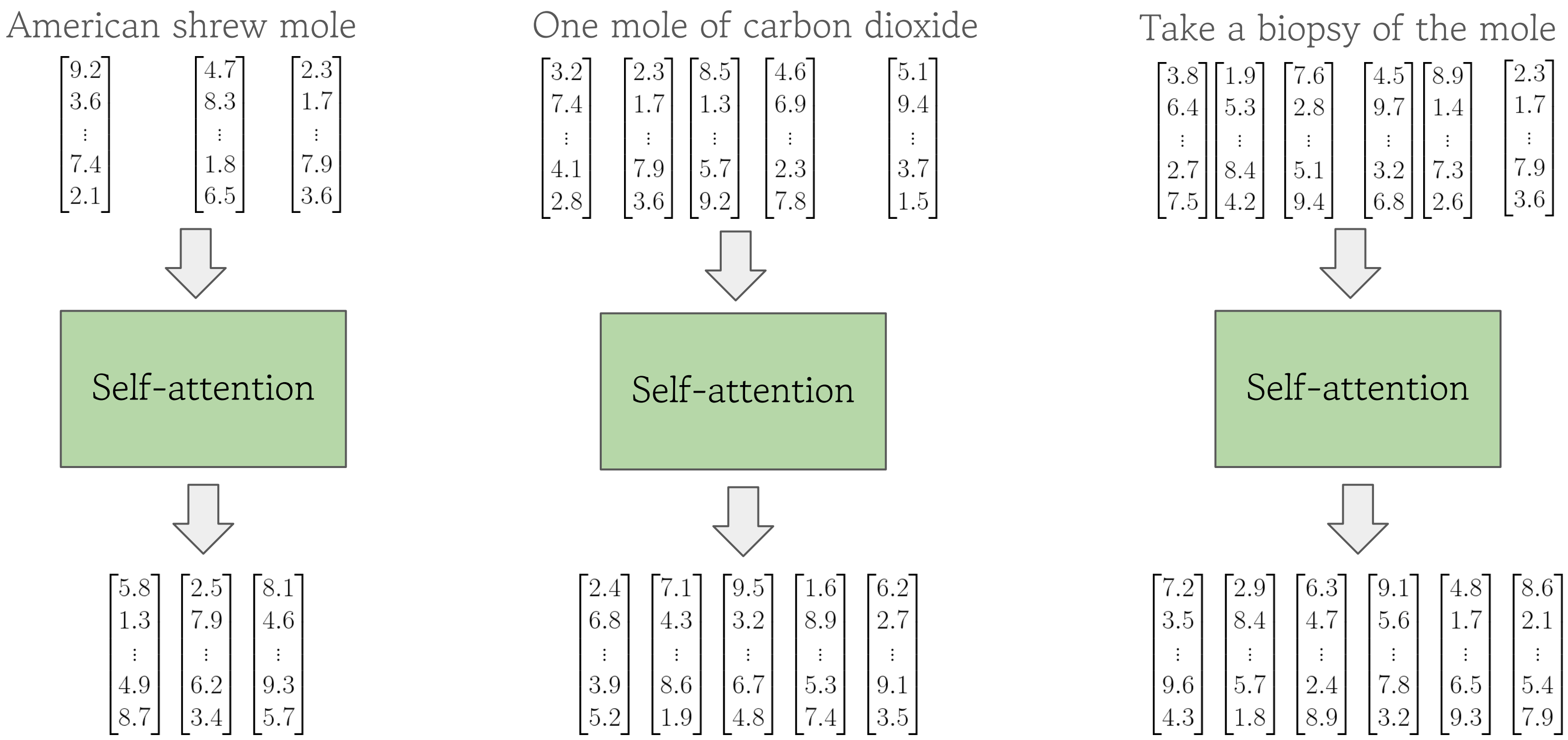
American shrew mole

One mole of carbon dioxide

6.02×10^{23}

Take a biopsy of the mole

Self Attention: What we want (from 3b1b)



Self Attention: What we want (from 3b1b)

American shrew mole

Self-attention

$$\begin{bmatrix}5.8\\1.3\\\vdots\\4.9\\8.7\end{bmatrix}\begin{bmatrix}2.5\\7.9\\\vdots\\6.2\\3.4\end{bmatrix}\boxed{\begin{bmatrix}8.1\\4.6\\\vdots\\9.3\\5.7\end{bmatrix}}$$

One mole of carbon dioxide

Self-attention

$$\begin{bmatrix}2.4\\6.8\\\vdots\\3.9\\5.2\end{bmatrix}\boxed{\begin{bmatrix}7.1\\4.3\\\vdots\\8.6\\1.9\end{bmatrix}}\begin{bmatrix}9.5\\3.2\\\vdots\\6.7\\4.8\end{bmatrix}\begin{bmatrix}1.6\\8.9\\\vdots\\5.3\\7.4\end{bmatrix}\begin{bmatrix}6.2\\2.7\\\vdots\\9.1\\3.5\end{bmatrix}$$

$$6.02 \times 10^{23}$$

Take a biopsy of the mole

Self-attention

$$\begin{bmatrix}7.2\\3.5\\\vdots\\9.6\\4.3\end{bmatrix}\begin{bmatrix}2.9\\8.4\\\vdots\\5.7\\1.8\end{bmatrix}\begin{bmatrix}6.3\\4.7\\\vdots\\2.4\\8.9\end{bmatrix}\begin{bmatrix}9.1\\5.6\\\vdots\\7.8\\3.2\end{bmatrix}\begin{bmatrix}4.8\\1.7\\\vdots\\6.5\\9.3\end{bmatrix}\boxed{\begin{bmatrix}8.6\\2.1\\\vdots\\5.4\\7.9\end{bmatrix}}$$

Background: Self-Attention

$$\mathbf{W}_q \in \mathbb{R}^{d_q \times d}, \quad \mathbf{W}_k \in \mathbb{R}^{d_k \times d}, \quad \mathbf{W}_v \in \mathbb{R}^{d_v \times d}$$

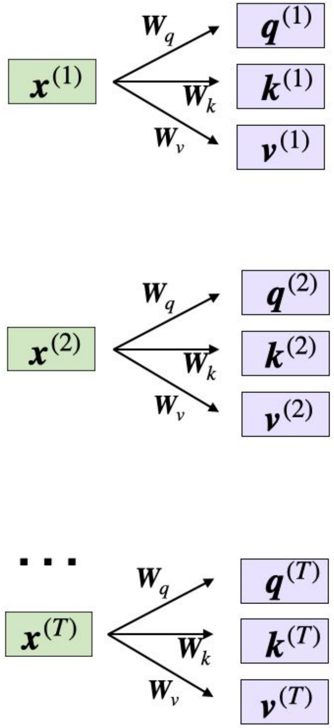
Queries: $\mathbf{q}^{(i)} = \mathbf{W}_q \mathbf{x}^{(i)}$ for $i \in [1, T]$

Keys: $\mathbf{k}^{(i)} = \mathbf{W}_k \mathbf{x}^{(i)}$ for $i \in [1, T]$

Values: $\mathbf{v}^{(i)} = \mathbf{W}_v \mathbf{x}^{(i)}$ for $i \in [1, T]$

$$\mathbf{Q} = \begin{bmatrix} [- & \mathbf{q}^{(1)} & -] \\ & \vdots & \\ [- & \mathbf{q}^{(n)} & -] \end{bmatrix} \in \mathbb{R}^{n \times d_q} \quad \mathbf{V} = \begin{bmatrix} [- & \mathbf{v}^{(1)} & -] \\ & \vdots & \\ [- & \mathbf{v}^{(n)} & -] \end{bmatrix} \in \mathbb{R}^{n \times d_v}$$

$$\mathbf{K} = \begin{bmatrix} [- & \mathbf{k}^{(1)} & -] \\ & \vdots & \\ [- & \mathbf{k}^{(n)} & -] \end{bmatrix} \in \mathbb{R}^{n \times d_k}$$

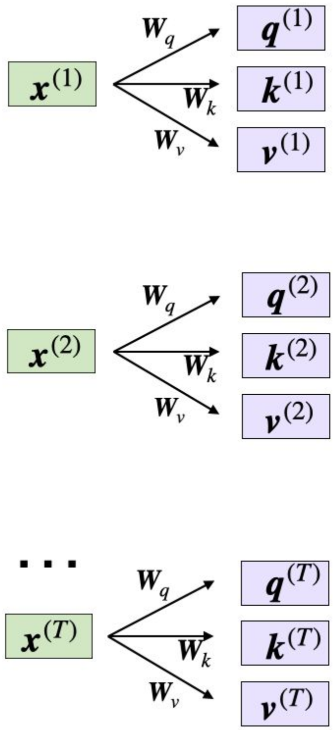


Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

Background: Self-Attention

$$\mathbf{Q} = \begin{bmatrix} [\text{—} \ \ \mathbf{q}^{(1)} \ \ \text{—}] \\ \vdots \\ [\text{—} \ \ \mathbf{q}^{(n)} \ \ \text{—}] \end{bmatrix} \in \mathbb{R}^{n \times d_q} \quad \mathbf{V} = \begin{bmatrix} [\text{—} \ \ \mathbf{v}^{(1)} \ \ \text{—}] \\ \vdots \\ [\text{—} \ \ \mathbf{v}^{(n)} \ \ \text{—}] \end{bmatrix} \in \mathbb{R}^{n \times d_v}$$

$$\mathbf{K} = \begin{bmatrix} [\text{—} \ \ \mathbf{k}^{(1)} \ \ \text{—}] \\ \vdots \\ [\text{—} \ \ \mathbf{k}^{(n)} \ \ \text{—}] \end{bmatrix} \in \mathbb{R}^{n \times d_k}$$



Self-Attention: $\mathbb{S}(\mathbf{QK}^T)\mathbf{V}$, where $\mathbb{S}$ is the **row-wise** softmax.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

Background: Self-Attention

Self-Attention: $\mathbb{S}(\mathbf{QK}^T)\mathbf{V}$, where $\mathbb{S}$ is the **row-wise** softmax.

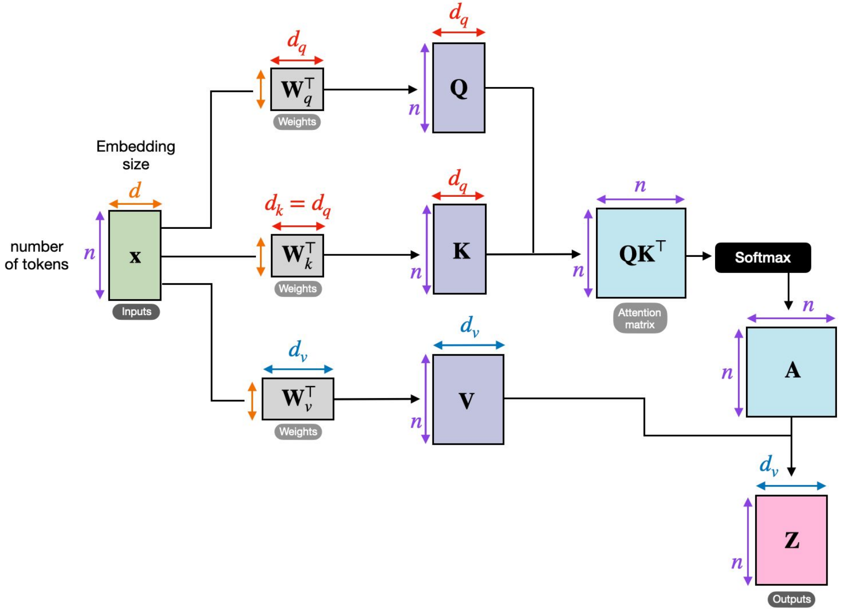
Observation 1: Convex combination of input.

$$[p_1 \;\; p_2 \;\; p_3] \begin{bmatrix} [- & \mathbf{v}^{(1)} & -] \\ [- & \mathbf{v}^{(2)} & -] \\ [- & \mathbf{v}^{(3)} & -] \end{bmatrix} = p_1\mathbf{v}^{(1)} + p_2\mathbf{v}^{(2)} + p_3\mathbf{v}^{(3)}$$

Observation 2: Context-dependent weighting

For, $\mathbf{p} = \mathbb{S}(\mathbf{QK}^T)$

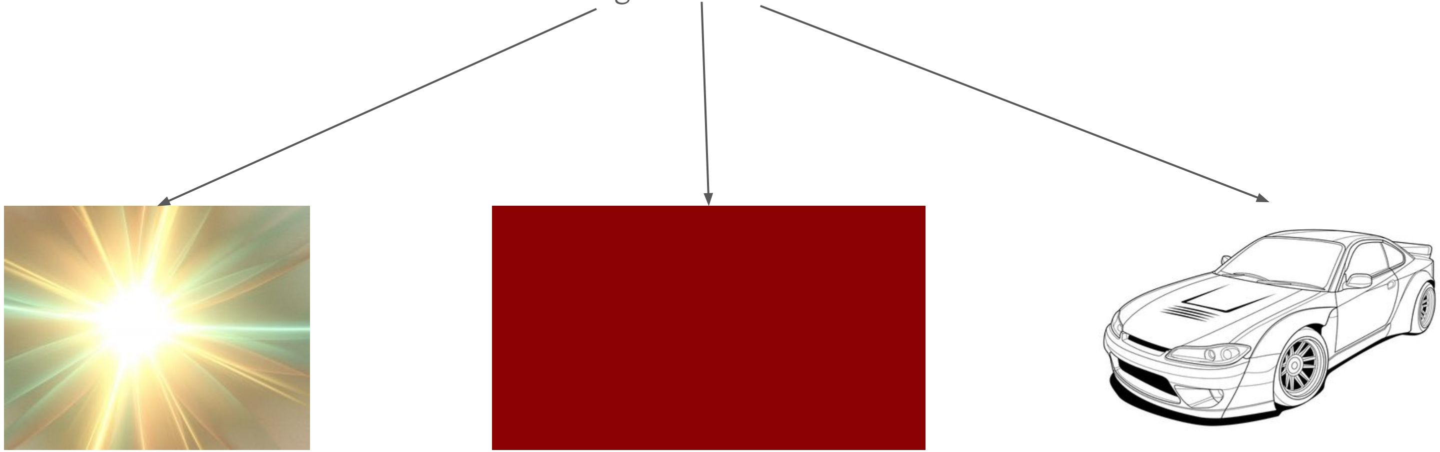
$$p_i = \frac{\mathbf{q}^{(i)} \cdot \mathbf{k}^{(i)}}{\sum_j \mathbf{q}^{(i)} \cdot \mathbf{k}^{(j)}}$$



Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

Self-Attention: Intuition?

$\mathbf{x}$ = “The bright red car was cool”

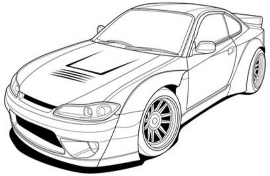


Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

Self-Attention: Intuition?

$\mathbf{x}$ = "The $\boxed{\text{bright}}$ red car was cool"

$\color{red}{K_2}$ $=$ Adjective in position 2



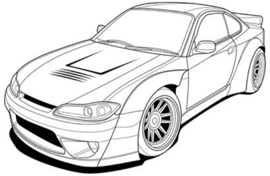
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

Self-Attention: Intuition?

$\mathbf{x}$ = “The bright red car was cool”

$K_2 =$ Adjective in position 2

$K_3 =$ Adjective in position 3



Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

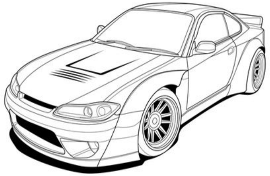
Self-Attention: Intuition?

$\mathbf{x}$ = “The bright red car was cool”

K_2 = Adjective in position 2

K_3 = Adjective in position 3

Q_4 = Noun in position 4



Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

Self-Attention: Intuition?

$\mathbf{x}$ = “The bright red car was cool”

K_2 = Adjective in position 2

K_3 = Adjective in position 3

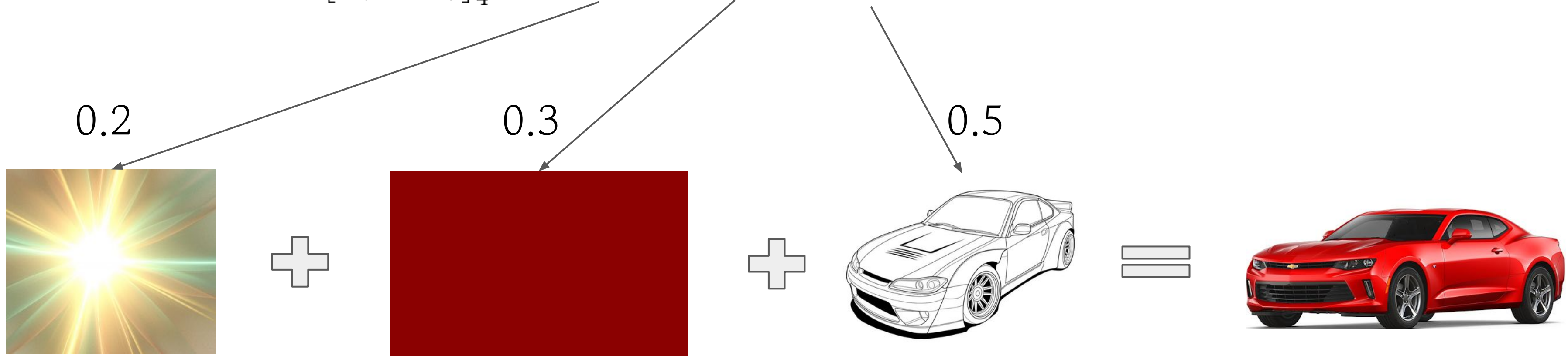
Q_4 = Noun in position 4

Intuition: Learn W_Q and W_K so that nouns “pay attention” to nearby adjectives.

Extreme setting:

$$\left[\mathbb{S}(\mathbf{QK}^T)\right]_4 = \mathbb{S}\left(\left[\mathbf{q}^{(4)}\cdot\mathbf{k}^{(1)}\ \ \mathbf{q}^{(4)}\cdot\mathbf{k}^{(2)}\ \ \mathbf{q}^{(4)}\cdot\mathbf{k}^{(3)}\ \ \mathbf{q}^{(4)}\cdot\mathbf{k}^{(4)}\ \ \mathbf{q}^{(4)}\cdot\mathbf{k}^{(5)}\ \ \mathbf{q}^{(4)}\cdot\mathbf{k}^{(6)}\right]\right)$$
$$= \left[0\ \ 0.2\ \ 0.3\ \ 0.5\ \ 0\ \ 0\right]$$

$$\left[\mathbb{S}(\mathbf{QK}^T)\right]_4 \mathbf{V} = 0.2\mathbf{v}^{(2)} + 0.3\mathbf{v}^{(3)} + 0.5\mathbf{v}^{(5)}$$



Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

Background: Masked Self-Attention

Observation 1: Convex combination of input.

$$\begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} [\text{—} & \mathbf{v}^{(1)} & \text{—}] \\ [\text{—} & \mathbf{v}^{(2)} & \text{—}] \\ [\text{—} & \mathbf{v}^{(3)} & \text{—}] \end{bmatrix} = p_1\mathbf{v}^{(1)} + p_2\mathbf{v}^{(2)} + p_3\mathbf{v}^{(3)}$$

Observation 2: Context-dependent weighting

For, $\mathbf{p} = \mathbb{S}(\mathbf{QK}^T)$

$$p_{ij} = \frac{\mathbf{q}^{(i)} \cdot \mathbf{k}^{(j)}}{\sum_j \mathbf{q}^{(i)} \cdot \mathbf{k}^{(j)}}$$

$$\mathcal{M}(\mathbf{QK}^T) = \begin{bmatrix} \cdot & -\infty & -\infty & -\infty & -\infty \\ \cdot & \cdot & -\infty & -\infty & -\infty \\ \cdot & \cdot & \cdot & -\infty & -\infty \\ \cdot & \cdot & \cdot & \cdot & -\infty \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

- Use $\mathbb{S}(\mathcal{M}(\mathbf{QK}^T))$
- Only attend to **previous** words in sentence.

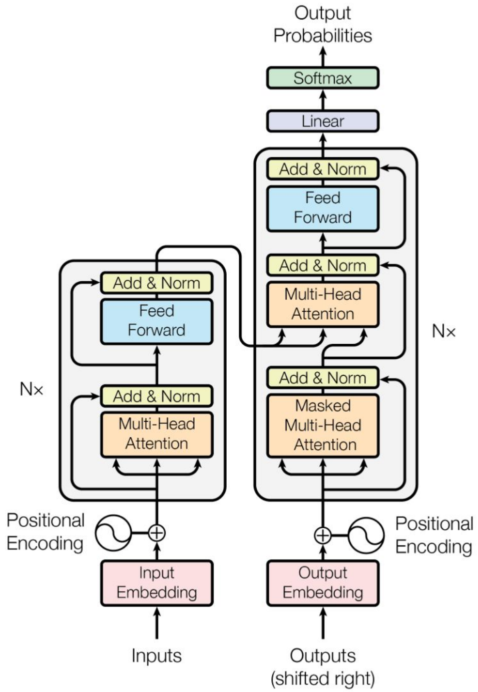
$$\begin{bmatrix} p_{11} & 0 & 0 \\ p_{21} & p_{22} & 0 \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} [\text{—} & \mathbf{v}^{(1)} & \text{—}] \\ [\text{—} & \mathbf{v}^{(2)} & \text{—}] \\ [\text{—} & \mathbf{v}^{(3)} & \text{—}] \end{bmatrix} = \begin{bmatrix} p_{11}\mathbf{v}^{(1)} \\ p_{21}\mathbf{v}^{(1)} + p_{22}\mathbf{v}^{(2)} \\ p_{31}\mathbf{v}^{(1)} + p_{32}\mathbf{v}^{(2)} + p_{33}\mathbf{v}^{(3)} \end{bmatrix}$$

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

Background: Transformer Architecture

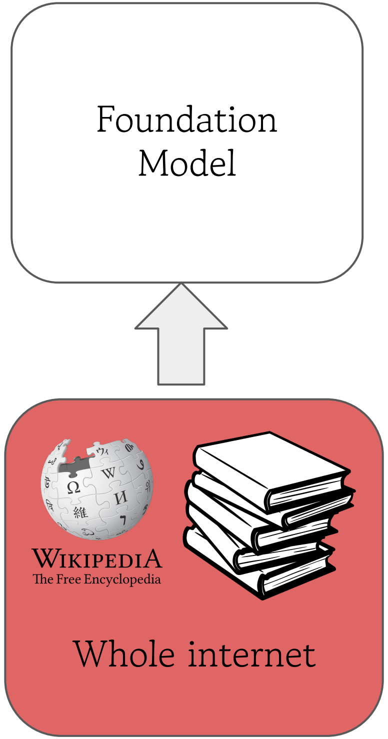
- Transformer is just stacking boxes of attention, FFN, skip connections, etc.

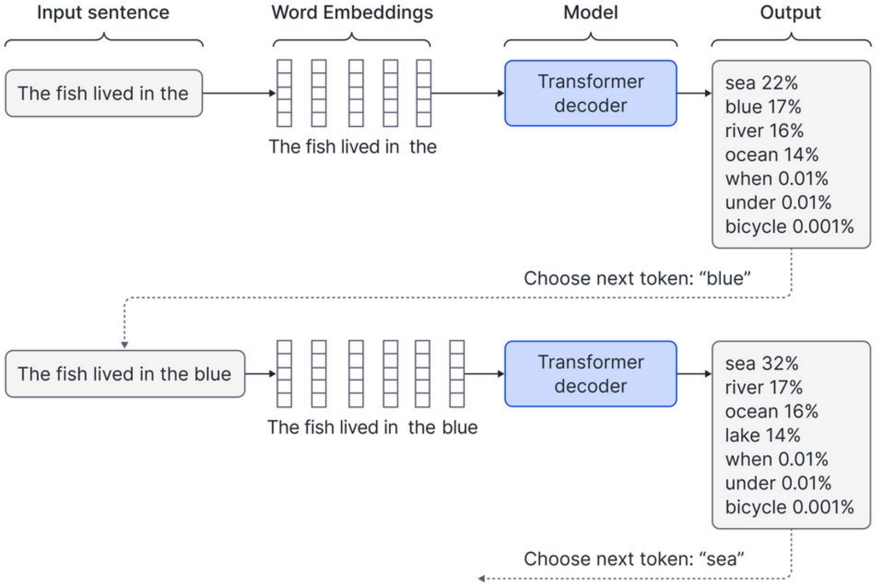
- Modern architectures tend to just be more computationally efficient variants i.e.
    - Multi-Query Attention
    - Grouped-Query Attention
    - Multi-Latent Attention
    - …



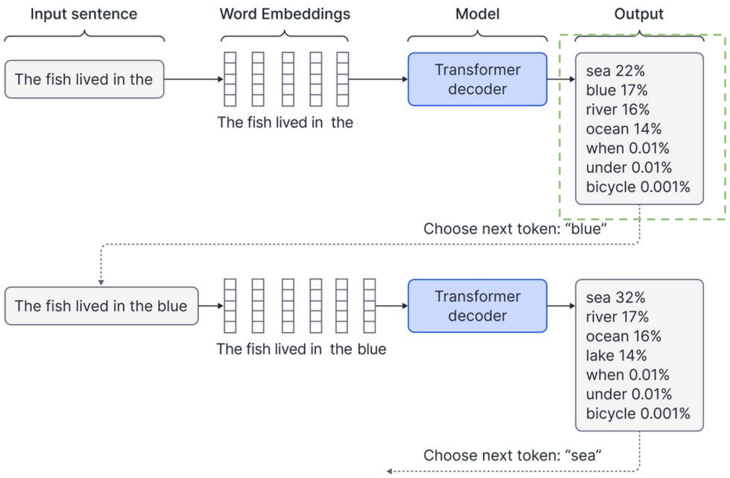
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need.*

Pre-R1: Training Pipeline





Example Pre-training Step



$$p_\theta(\cdot \mid x_i) = \begin{bmatrix} .22 \\ .17 \\ .16 \\ \vdots \end{bmatrix}$$

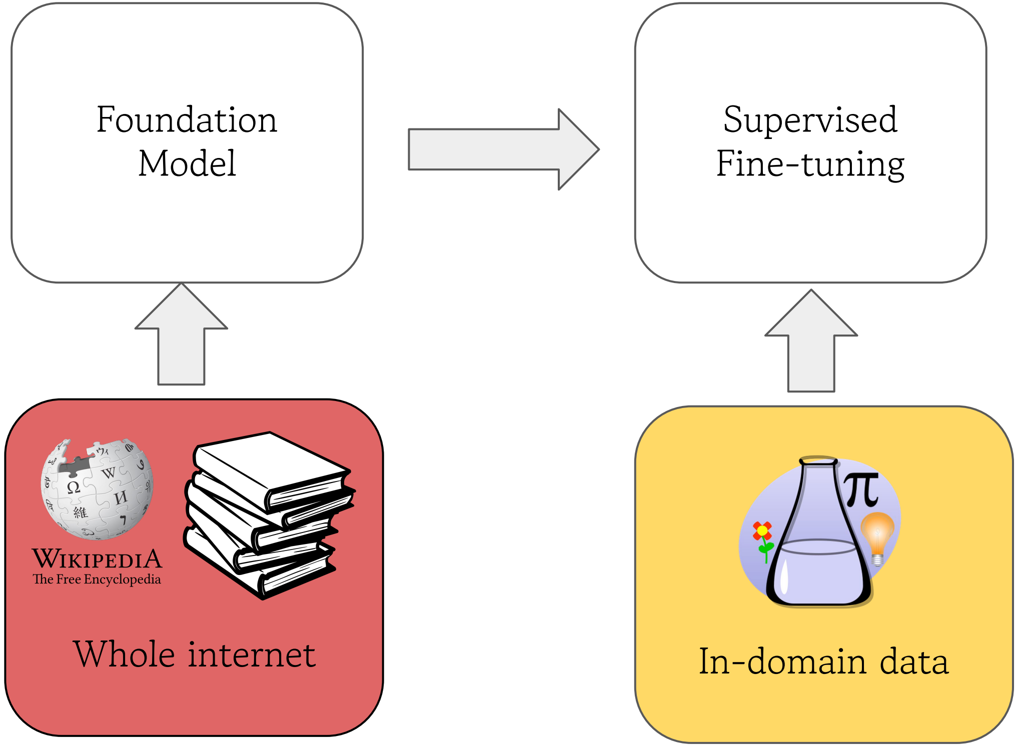
$$p_Y(\cdot | x_i) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

$x_i =$ "The fish lived in the"

$y_i =$ "The fish lived in the blue"

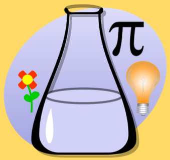
$$\mathcal{L}(\theta) = H(p_\theta,\ p_Y \mid \mathbf{x})$$

Pre-R1: Training Pipeline



Pre-R1: Training Pipeline

Supervised Fine-tuning



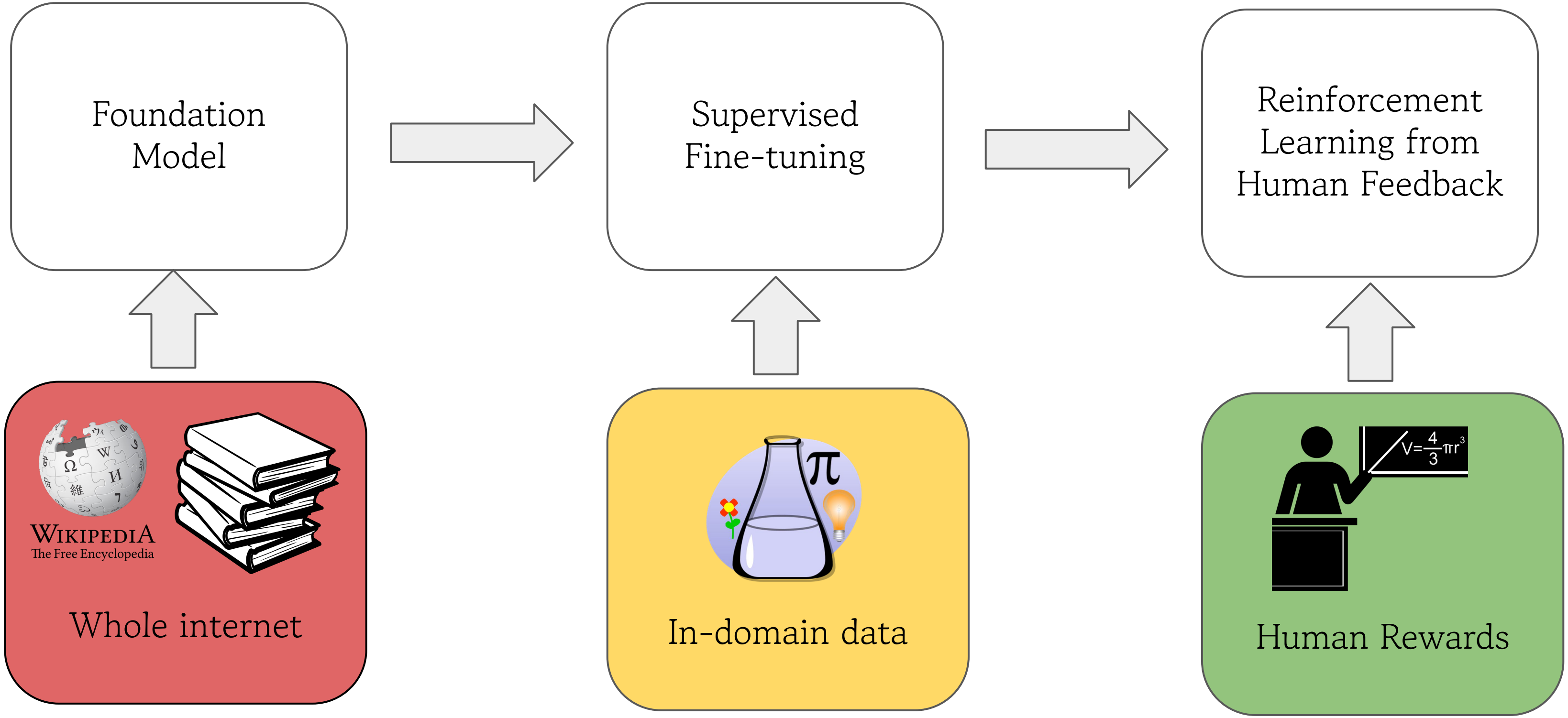
In-domain data

$\mathbf{x}$ = Consider a regular octagon. How many different triangles can be formed if the octagon is placed inside a circle and we can also use the center of the circle as a vertex for the triangles? Let's think step by step.

$\mathbf{y}$ = *Consider a regular octagon. How many different triangles can be formed if the octagon is placed inside a circle and we can also use the center of the circle as a vertex for the triangles?* Let's think step by step. <Thought> Alright, I need to figure out how many different triangles can be formed in a regular octagon that's placed inside a circle, with the added twist that the center of the circle can also be used as a vertex for the triangles. Let's break this down step by step. First, let's recall that a regular octagon has 8 equal sides and 8 equal angles. Since it's placed inside a circle, each vertex of the octagon lies on the circumference of the circle. Additionally, we have the center of the circle as an extra point that can be used as a vertex for forming triangles. So, in total, we have 9 points to consider: the 8 vertices of the octagon and the center of the circle. Now, to form a triangle, we need to select any 3 of these 9 points. The number of ways to choose 3 points out of 9 is given by the combination formula C(n, k) = n! / (k!(n-k)!), where n is the total number of points and k is the number of points to choose. Calculating C(9, 3): C(9, 3) = 9! / (3! * (9-3)!) = (9 * 8 * 7) / (3 * 2 * 1) = 84. So, there are 84 possible triangles if we consider all combinations of the 9 points. But wait, let's make sure we're not missing anything. Are there any restrictions or special cases? For instance, could any three points be collinear, making them unable to form a triangle? In a regular octagon inscribed in a circle, no three vertices are collinear because the angles between consecutive vertices are 45 degrees, and no three points on the circumference of a circle are collinear unless they lie on a straight line, which doesn't happen here. Additionally, including the center doesn't introduce any collinearity issues because the center is equidistant from all vertices, and connecting it with any two vertices forms a triangle. Therefore, all 84 combinations of 3 points will indeed form valid triangles. But let's double-check by considering different scenarios: 1. Triangles formed by three vertices of the octagon. 2. Triangles formed by two vertices and the center. 3. Triangles formed by one vertex, the center, and another point (but since we only have the center and the vertices, this is covered in the previous points). Calculating each scenario separately: 1. Number of triangles with three vertices: C(8, 3) = 56. 2. Number of triangles with two vertices and the center: C(8, 2) = 28. 3. Since there's only one center, and we've already accounted for triangles with two vertices and the center, there are no additional triangles to consider. Adding these up: 56 + 28 = 84, which matches our initial calculation. Is there any other consideration? Perhaps symmetry or identical triangles? In a regular octagon, due to its symmetry, some triangles might be congruent, but the question asks for different triangles, not necessarily unique in shape or size. So, even if some triangles are congruent, they are still counted as different if they consist of different sets of points. Therefore, the total number of different triangles that can be formed is 84. </Thought> <Output> 84 different triangles can be formed. </Output>

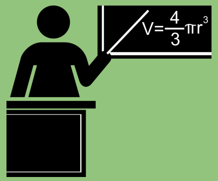
Now just do the same thing as pre-training!

Pre-R1: Training Pipeline

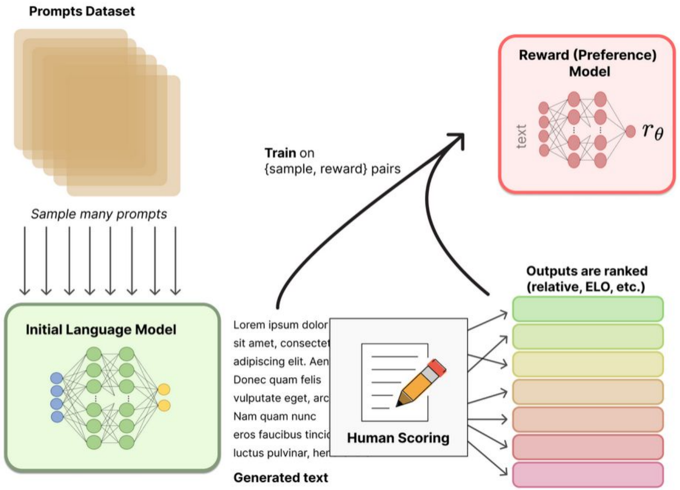


Pre-R1: Training Pipeline

Reinforcement Learning from Human Feedback



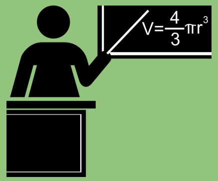
Human Rewards



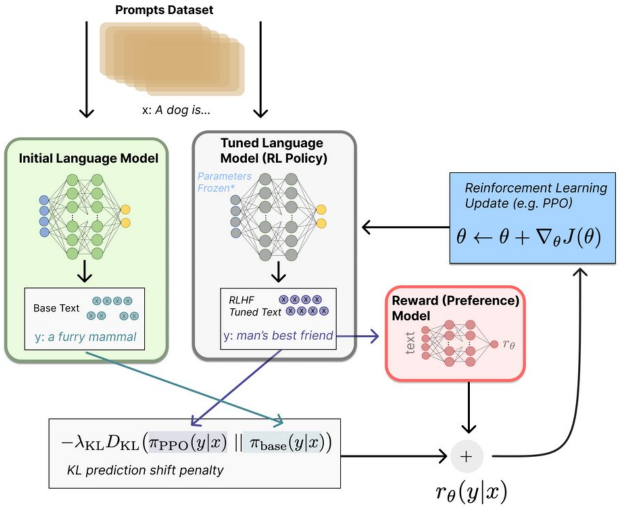
$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\theta(x,y_w) - r_\theta(x,y_l)\right)\right)\right]$$

Pre-R1: Training Pipeline

Reinforcement Learning from Human Feedback

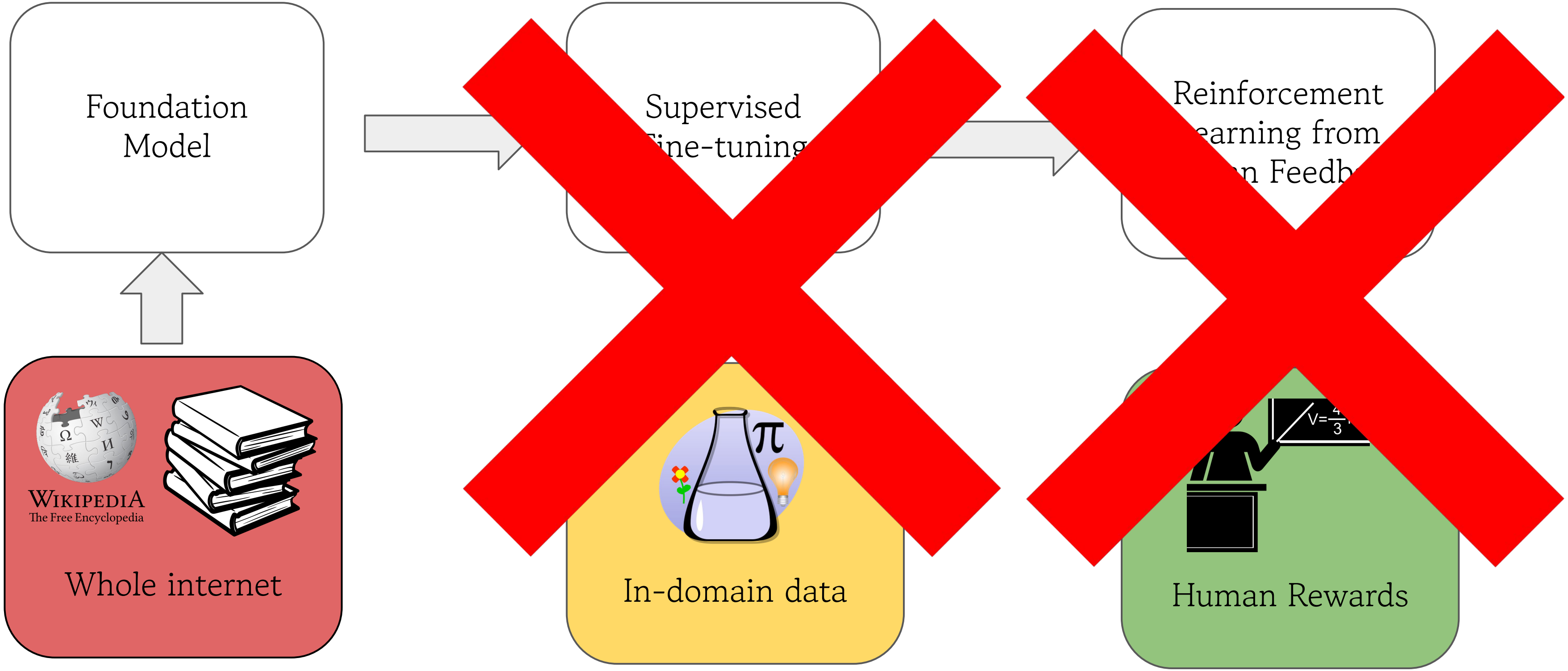


Human Rewards



$$\operatorname{loss}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_\theta(x,y) - \beta \log\left(\pi_\phi^{\mathrm{RL}}(y \mid x)/\pi^{\mathrm{SFT}}(y \mid x)\right)\right] + \gamma E_{x\sim D_{\mathrm{pretrain}}}\left[\log(\pi_\phi^{\mathrm{RL}}(x))\right]$$

Deepseek R1-Zero: RL Just Works



DeepSeek-AI; *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.*

Deepseek R1-Zero: RL Just Works

Foundation Model

Reinforcement Learning

WIKIPEDIA
The Free Encyclopedia

Whole internet

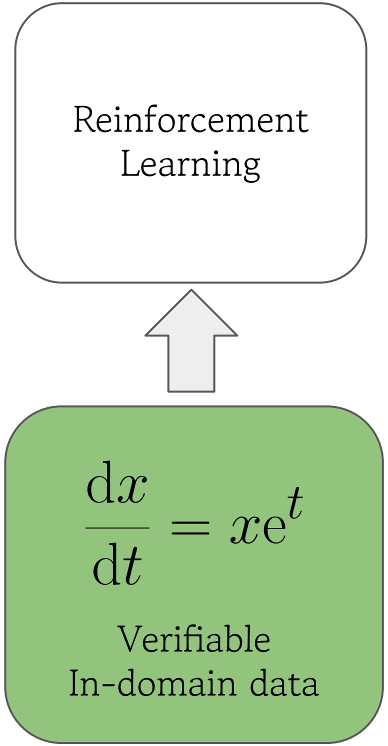
$$\frac{\mathrm{d}x}{\mathrm{d}t} = xe^t$$

Verifiable In-domain data

$$r = \begin{cases} 1, & \text{if correct} \\ -0.5, & \text{if incorrect but answered} \\ -1, & \text{if could not answer} \end{cases}$$

DeepSeek-AI; *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.*

Deepseek R1-Zero: aha!



Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>
To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\cdots$
$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$
Rearrange to isolate the inner square root term:
$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$
...
Wait, wait. Wait. That's an aha moment I can flag here.
Let's reevaluate this step-by-step to identify if the correct sum can be $\cdots$
We started with the equation:
$\sqrt{a - \sqrt{a + x}} = x$
First, let's square both sides:
$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$
Next, I could square both sides again, treating the equation: $\cdots$
...

DeepSeek-AI; *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.*

Caveat: Group Relative Policy Optimization

Question q

Sample G outputs from $\pi_{\text{old}}: \ \{o_i\}_{i=1}^{G}$

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$
$$= \frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{od}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{odd}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right)$$

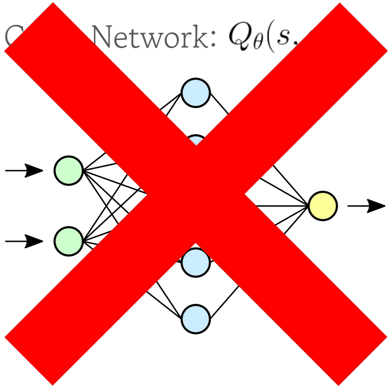
Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo; *Deepseekmath: Pushing the limits of mathematical reasoning in open language models.*

Caveat: Group Relative Policy Optimization

$$
\begin{aligned}
\mathcal{J}_{\mathrm{GRPO}}(\theta) &= \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)] \\
&= \frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{od}}(o_i|q)}A_i, \mathrm{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{odd}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta || \pi_{ref}\right)\right)
\end{aligned}
$$

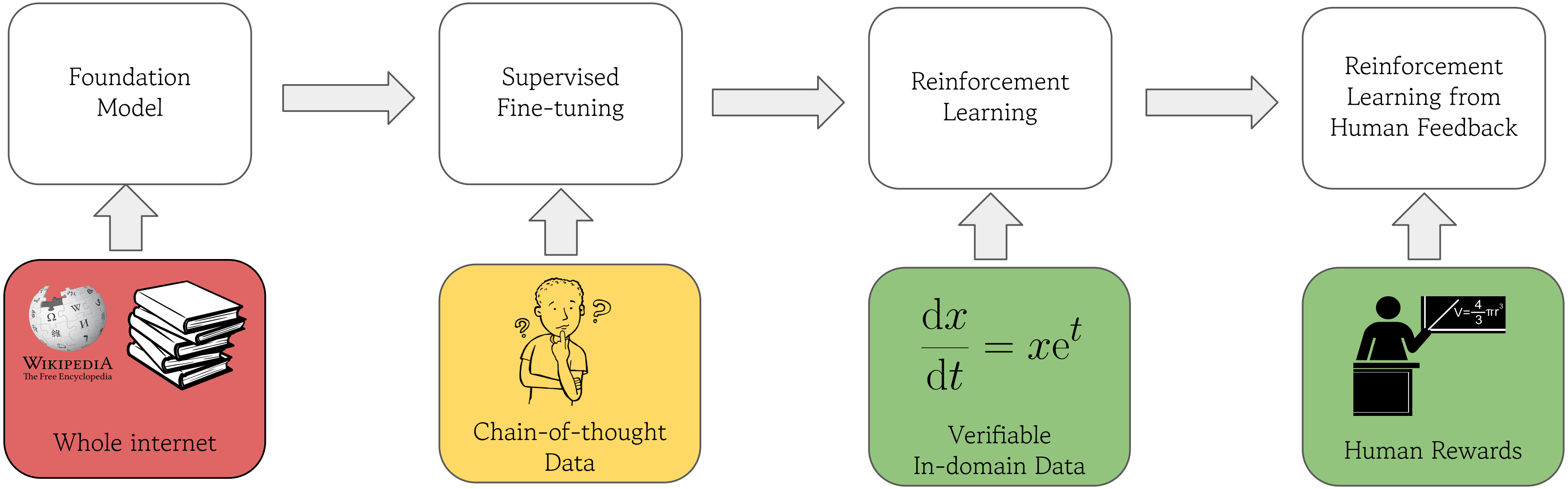
Difference with PPO

$$
A_i = \frac{r_i - \mathrm{mean}(\{r_1, r_2, \cdots, r_G\})}{\mathrm{std}(\{r_1, r_2, \cdots, r_G\})}
$$



Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo; *Deepseekmath: Pushing the limits of mathematical reasoning in open language models.*

Caveat: Deepseek R1



DeepSeek-AI; *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.*

Deepseek R1: More Results

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820
OpenAI-o1-0912	74.4	83.3	94.8	77.3	63.4	1843
DeepSeek-R1-Zero	$\boxed{71.0}$	86.7	$\boxed{95.9}$	73.3	$\boxed{50.0}$	$\boxed{1444}$

R1 Zero results i.e. no SFT / RLHF.

- R1 - Improved **language coherence**
- Curated SFT dataset improves **performance**

	Benchmark (Metric)	**Claude-3.5-Sonnet-1022**	**GPT-4o 0513**	**DeepSeek V3**	**OpenAI o1-mini**	**OpenAI o1-1217**	**DeepSeek R1**
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (Pass@1)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	**92.3**
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	$\boxed{\mathbf{65.9}}$
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	**96.6**	96.3
	Codeforces (Rating)	717	759	1134	1820	**2061**	$\boxed{2029}$
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	$\boxed{\mathbf{79.8}}$
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	$\boxed{\mathbf{97.3}}$
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

DeepSeek-AI; *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.*

Deepseek R1: More Results

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820
OpenAI-o1-0912	74.4	83.3	94.8	77.3	63.4	1843
DeepSeek-R1-Zero	71.0	86.7	95.9	$\boxed{73.3}$	50.0	1444

R1 Zero results i.e. no SFT / RLHF.

- R1 - Improved **language coherence**
- Curated SFT dataset improves **performance**
- **Not** a total ordering

	Benchmark (Metric)	**Claude-3.5-Sonnet-1022**	**GPT-4o 0513**	**DeepSeek V3**	**OpenAI o1-mini**	**OpenAI o1-1217**	**DeepSeek R1**
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (Pass@1)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	$\boxed{71.5}$
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	**92.3**
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	**96.6**	96.3
	Codeforces (Rating)	717	759	1134	1820	**2061**	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	**79.8**
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	**97.3**
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

DeepSeek-AI; *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.*

Deepseek R1: More Results

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	**1820**
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9	1316
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	**72.6**	83.3	94.3	62.1	57.2	1691
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205
DeepSeek-R1-Distill-Llama-70B	70.0	**86.7**	**94.5**	**65.2**	**57.5**	1633

Distilling R1 results.

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCodeBench
	pass@1	cons@64	pass@1	pass@1	pass@1
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9
DeepSeek-R1-Zero-Qwen-32B	47.0	60.0	91.6	55.0	40.2
DeepSeek-R1-Distill-Qwen-32B	**72.6**	**83.3**	**94.3**	**62.1**	**57.2**

Training Qwen 32B with RL vs. Distilling with R1

	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (Pass@1)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	**92.3**
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	**96.6**	96.3
	Codeforces (Rating)	717	759	1134	1820	**2061**	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	**79.8**
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	**97.3**
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

DeepSeek-AI; *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.*

Now back to the regularly scheduled program!

Contrastive Reinforcement Learning

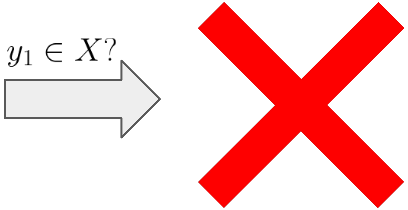
Preliminaries: Noise Contrastive Estimation

- Observed dataset $(x_1, \ldots, x_T)$: T observations from the dataset $\mathbf{x}$
  - Assume $x \sim p_X$
- Generated dataset $(y_1, \ldots, y_T)$: T observations drawn from some p_Y

Idea: Learn p_X by learning to distinguish between $\mathbf{x}$ and $\mathbf{y}$

x_1

y_1



Michael Gutmann, Aapo Hyvärinen; *Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.*

Preliminaries: Noise Contrastive Estimation

Idea: Learn p_X by learning to distinguish between $\mathbf{x}$ and $\mathbf{y}$

How? Binary classification loss

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim p_X, y \sim p_Y}\Big[\log \sigma(f_\theta(x)) + \log(1 - \sigma(f_\theta(y)))\Big]$$

$$f^*(z) = \log\left(\frac{p_X(z)}{p_Y(z)}\right)$$

Michael Gutmann, Aapo Hyvärinen; *Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.*

Preliminaries: Noise Contrastive Estimation

Idea: Learn p_X by learning to distinguish between $\mathbf{x}$ and $\mathbf{y}$

How?

$$J_T(\theta) = \frac{1}{2T} \sum_t \underbrace{\ln\left[h\left(\mathbf{x}_t;\theta\right)\right]}_{\text{Increase } p_X(\mathbf{x}_t;\theta)} + \underbrace{\ln\left[1 - h\left(\mathbf{y}_t;\theta\right)\right]}_{\text{Decrease } p_X(\mathbf{y}_t;\theta)}$$

Michael Gutmann, Aapo Hyvärinen; *Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.*

Preliminaries: Noise Contrastive Estimation

How?

$$J_T(\theta) = \frac{1}{2T}\sum_t \ln\left[h\left(\mathbf{x}_t;\theta\right)\right] + \ln\left[1 - h\left(\mathbf{y}_t;\theta\right)\right]$$

Unpacking

$$h(\mathbf{u};\theta) = \frac{1}{\underbrace{1+\exp[-G(\mathbf{u};\theta)]}_{\text{Standard policy parametrization}}},$$

$$G(\mathbf{u};\theta) = \underbrace{\ln p_X(\mathbf{u};\theta) - \ln p_Y(\mathbf{u})}_{\text{Use knowledge of } p_Y \text{ to only update on disagreement.}}$$

Michael Gutmann, Aapo Hyvärinen; *Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.*

Preliminaries: Noise Contrastive Estimation

How?

$$J_T(\theta) = \frac{1}{2T} \sum_t \ln\left[h\left(\mathbf{x}_t;\theta\right)\right] + \ln\left[1 - h\left(\mathbf{y}_t;\theta\right)\right]$$

Rewrite

$$\tilde{J}(f) = \frac{1}{2} \mathrm{E} \ln\left[r\left(f(\mathbf{x}) - \ln p_n(\mathbf{x})\right)\right] + \ln\left[1 - r\left(f(\mathbf{y}) - \ln p_n(\mathbf{y})\right)\right].$$

Then

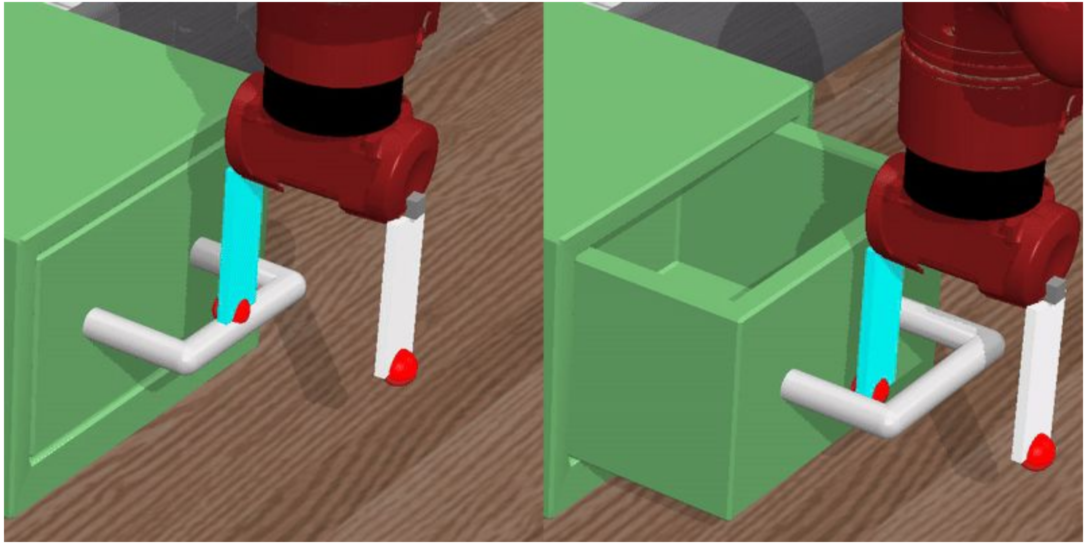
Theorem. $\tilde{J}$ attains a maximum at $f(\cdot) = \ln p_X(\cdot)$. There are no other extrema if p_Y is chosen such that it is non-zero whenever p_X is non zero.

Michael Gutmann, Aapo Hyvärinen; *Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.*

Contrastive Reinforcement Learning

Goal conditioned RL: *How do we learn* $Q(s, a, g)$ *for*

$$r_g(s_t, a_t) \triangleq (1 - \gamma)p(s_{t+1} = s_g \mid s_t, a_t).$$



Benjamin Eysenbach, Tianjun Zhang, Ruslan Salakhutdinov, Sergey Levine; *Contrastive Learning as Goal-Conditioned Reinforcement Learning*

Contrastive Reinforcement Learning: Big Picture

Key observation:

$$Q^{\pi}(s,a,s_g) = p^{\pi(\cdot|\cdot)}(s_{t+} = s_g|s,a) \triangleq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t^{\pi(\cdot|\cdot)}(s_t = s_g|s,a)$$

Distribution -> **contrastive learning!**

Where,

$p_t^{\pi}(s)$ is the probability density over states that policy π visits after t steps.

$$p^{\pi(\cdot\,|\cdot,s_g)}(s_{t+} = s) \triangleq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t^{\pi(\cdot\,|\cdot,s_g)}(s_t = s),$$

$$\pi(a \mid s) \triangleq \int \pi(a \mid s, s_g) p^{\pi}(s_g \mid s) ds_g$$

Contrastive Reinforcement Learning: Big Picture

Key observation:

$$Q^{\pi}(s,a,s_g) = p^{\pi(\cdot|\cdot)}(s_{t+} = s_g|s,a) \triangleq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t^{\pi(\cdot|\cdot)}(s_t = s_g|s,a)$$

Distribution -> **contrastive learning!**

Use NCE:

$$\max_f \mathbb{E}_{\substack{(s,a)\sim p(s,a), s_f^- \sim p(s_f) \\ s_f^+ \sim p^{\pi(\cdot|\cdot)}(s_{t+}|s_t,a_t)}} \left[\mathcal{L}(s,a,s_f^+,s_f^-)\right],$$

$$\text{where} \quad \mathcal{L}(s,a,s_f^+,s_f^-) \triangleq \log\sigma(\underbrace{f(s,a,s_f^+)}_{\phi(s,a)^T\psi(s_f^+)}) + \log(1-\sigma(\underbrace{f(s,a,s_f^-)}_{\phi(s,a)^T\psi(s_f^-)})).$$

How do we pick positive / negative samples?

Benjamin Eysenbach, Tianjun Zhang, Ruslan Salakhutdinov, Sergey Levine; *Contrastive Learning as Goal-Conditioned Reinforcement Learning*

Contrastive Reinforcement Learning: Big Picture

$t \sim \mathrm{Geom}(1-\gamma)$ $\quad$ $t \sim \mathrm{Unif}(n)$

$$(s_1, a_1), (s_2, a_2), (s_3, a_3), (s_4, a_4), (s_5, a_5), (s_6, a_6), (s_7, a_7), (s_8, a_8), (s_9, a_9), (s_{10}, a_{10}), \ldots, (s_n, a_n)$$

Contrastive Reinforcement Learning: Big Picture

$t \sim \text{Geom}(1-\gamma) \qquad t \sim \text{Unif}(n)$

$$(s_1, a_1), (s_2, a_2), (s_3, a_3), (s_4, a_4), (s_5, a_5), (s_6, a_6), (s_7, a_7), (s_8, a_8), (s_9, a_9), (s_{10}, a_{10}), \ldots, (s_n, a_n)$$

$$\mathcal{L}_1(\theta) = \log \sigma(f_\theta(s_1, a_1, s_8)) + \log(1 - \sigma(f_\theta(s_1, a_1, s_3)))$$

$$\begin{aligned}
\widehat{\mathcal{L}}(\theta) &= \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i \\
&= \frac{1}{n} \sum_{i=1}^{n} \left[\log \sigma(f_\theta(s_i, a_i, s_f^+)) + \log(1 - \sigma(f_\theta(s_i, a_i, s_f^-))) \right]
\end{aligned}$$

Benjamin Eysenbach, Tianjun Zhang, Ruslan Salakhutdinov, Sergey Levine; *Contrastive Learning as Goal-Conditioned Reinforcement Learning*

Contrastive Reinforcement Learning: Big Payoff

Recall:

Theorem. $\tilde{J}$ attains a maximum at $f(\cdot) = \ln p_X(\cdot)$. There are no other extrema if p_Y is chosen such that it is non-zero whenever p_X is non zero.

Use NCE:

$$\max_f \mathbb{E}_{\substack{(s,a)\sim p(s,a), s_f^- \sim p(s_f) \\ s_f^+ \sim p^{\pi(\cdot|\cdot)}(s_{t+}|s_t,a_t)}} \left[\mathcal{L}(s, a, s_f^+, s_f^-)\right],$$

$$\text{where} \quad \mathcal{L}(s, a, s_f^+, s_f^-) \triangleq \log \sigma(\underbrace{f(s, a, s_f^+)}_{\phi(s,a)^T\psi(s_f^+)}) + \log(1 - \sigma(\underbrace{f(s, a, s_f^-)}_{\phi(s,a)^T\psi(s_f^-)})).$$

Contrastive RL Version: $\exp(f^*(s, a, s_f)) = \dfrac{1}{p(s_f)} \cdot Q_{s_f}^{\pi(\cdot|\cdot)}(s, a).$

Benjamin Eysenbach, Tianjun Zhang, Ruslan Salakhutdinov, Sergey Levine; *Contrastive Learning as Goal-Conditioned Reinforcement Learning*

Contrastive Reinforcement Learning: Big Payoff

Recall:

$$\max_f \mathbb{E}_{\substack{(s,a)\sim p(s,a), s_f^- \sim p(s_f) \\ s_f^+ \sim p^{\pi(\cdot|\cdot)}(s_{t+}|s_t,a_t)}} \left[\mathcal{L}(s, a, s_f^+, s_f^-)\right],$$

$$\text{where} \quad \mathcal{L}(s, a, s_f^+, s_f^-) \triangleq \log \sigma(\underbrace{f(s, a, s_f^+)}_{\phi(s,a)^T \psi(s_f^+)}) + \log(1 - \sigma(\underbrace{f(s, a, s_f^-)}_{\phi(s,a)^T \psi(s_f^-)})).$$

$$\Rightarrow \quad f^*(s, a, s_g) = \log\left(\frac{p^{\pi(\cdot|\cdot)}(s_g \mid s, a)}{p(s_g)}\right)$$

Contrastive RL Version: $\exp(f^*(s, a, s_f)) = \dfrac{1}{p(s_f)} \cdot Q_{s_f}^{\pi(\cdot|\cdot)}(s, a).$

Benjamin Eysenbach, Tianjun Zhang, Ruslan Salakhutdinov, Sergey Levine; *Contrastive Learning as Goal-Conditioned Reinforcement Learning*

Research Direction: Safe Reinforcement Learning

Obstacles can be seen as **goals**

Instead of solving

$$\pi(a|s,a,g) = \arg\max_a Q(s,a,g)$$

Solve **constrained** optimization where

$$\pi(a|s,a,\mathcal{O}) = \arg\max_a Q(s,a,g) - \sum_{o\in\mathcal{O}} \lambda_o Q(s,a,o)$$

Or

$$\pi(a|s,a,\mathcal{O}) = \arg\max_a \log Q(s,a,g) - \sum_{o\in\mathcal{O}} \log(1 - Q(s,a,o))$$

Policy can adapt zero-shot to new obstacles and goals!

Research Direction: Hierarchical Reinforcement Learning

Sparse reward for **long-term** behavior.
Natural **decomposition** into goals.

i.e. to build a pickaxe intermediate goal to collect wood, iron, etc.

Sub-policies can correspond to different **latent observations**

$$
\begin{array}{lll}
\pi_1^{\text{sub}} & \text{sees} & (\phi_1(s_1), a_1), (\phi_1(s_2), a_2), (\phi_1(s_3), a_3), \ldots \\
\pi_2^{\text{sub}} & \text{sees} & (\phi_2(s_1), a_1), (\phi_2(s_2), a_2), (\phi_2(s_3), a_3), \ldots \\
\pi_3^{\text{sub}} & \text{sees} & (\phi_3(s_1), a_1), (\phi_3(s_2), a_2), (\phi_3(s_3), a_3), \ldots \\
\vdots & \vdots & \vdots
\end{array}
$$

PLEASE REACH OUT IF INTERESTED / OR HAVE RELATED IDEAS

THANK YOU