13th November, 2025 Date:

Alignment Problem:

Assumption → Suppose we have a Pre-trained LLM "ω_p", we fine tune it for a downstream application like solving math problem.

Issues: ↳ math problem are not straight-forward to solve.

→ we want our LLMs to learn what actions not to take.

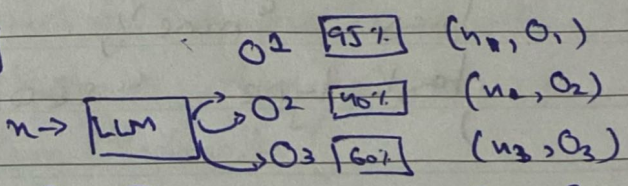
→ $x \rightarrow \hat{y}_+$

$\rightarrow \hat{y}_-$

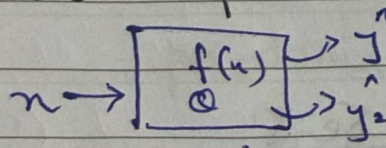
* in training LLM we try to get correct outp
* we don't teach which other solutions are wro
* to be able to follow our preferences.

1) Making Preference data

2) Ways of making preference data:
   1) Point-wise
   2) Pair-wise
   3) List-wise



⇒ Procedure of Generating Pair wise Data:



1) take LLM & generate responses
2) Rate Relative performance. (compare them)

$(x, \hat{y}_1)$ & $(x, \hat{y}_2)$

- User select his preference

x, y_{winner} & x, y_{loser}

Date: *coder *codeLLAMA

• π_θ is policy → LLM

⇒ RLHF:

- Reinforcement learning with Human feedback
- aligning LLM with human feedback

⇒ 2 Models:
Reward, Reinforcement.

⇒ Reward Model:
Quantifying quality of responses (reward by optimizing responses based on human feedback, human feedback is training label.)

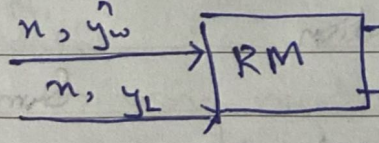
$x \rightarrow [f] \rightarrow \hat{y} \rightarrow [RM] \rightarrow r(x, \hat{y})$

↕ rhf → human feedback

→ Training a Reward Model on human Rated preference data:

① $(x, \hat{y}_w)$ $(x, \hat{y}_L)$

② Pairwise Loss:



Bradly Terr Model

③ Compute Loss: $\frac{1}{N}\sum_{n=1}^{N}$ sigmoid

$$L(\theta) = -E\left[\log\left(\sigma\left(r(x, \hat{y}_w) - r(x, \hat{y}_L)\right)\right)\right]$$

average; reward for winner; reward for losser

- Contrastive loss
- we want distinction b/w winning & losing so that it always choose winner one.

④ Reinforcement learning:
- Proximal Policy Optimizatio

$$r(o) = -\left[r(x, \hat{y}) - \lambda KL\left(\pi_\theta(\hat{y}/x) \,\|\, \pi_{old}(\hat{y}/x)\right)\right]$$

reward-; deviation reg-

- π_{KL} → Non-symmetric one distribution. b/w 2 distribution.
- $\pi_{old}(\hat{y}/x)$
- λ Controls change

③ BackProp to update Reward model

RL for deepSeek Math - Instruct 7B

Date

① Initialize policy $\pi_\theta \leftarrow \pi_{\theta old}$
② Update the weights of Policy
③ ↳ generate response $x \xrightarrow{\pi_{\theta old}} y$
↳ Assign Quality score.
$\hat{y} \rightarrow$ RM
↳ update Policy weights:
use generated response $\hat{y}$ along with its associated estimated $\gamma(x, \hat{y})$ to update π_θ using PPO loss.

$$L^{DPO} = -E\left[\log\left(\sigma\left(\beta \log\left(\frac{\pi_\theta(\hat{y}_w/x)}{\pi_{\theta old}(\hat{y}_w/x)}\right) - \beta\left(\frac{\pi_\theta(\hat{y}_l(x))}{\pi_{old}(\hat{y}_l/x)}\right)\right)\right)\right]$$

- here we are calculating ratio.
- β controls divergence
- less divergence

→ advantage *
→ value model.

→ Short Coming of Reward Model:
1) Too many steps
2) Imperfect proxies ← avg reward model
3) Many hyperparameters to update.

→ Reward • how correct
→ Value • go to
→ Reference, reward - baseline

- use Sampling model] Monte carlo
- another is DPO

Thinking model.
$x \rightarrow wp \rightarrow GRPO \rightarrow y$

Direct Preference Optimization
↳ Steps:
1) $\pi_\theta \leftarrow wp$ (LLM → policy)
2) for a given input prompt "x", Sample observation from π_{old}
3) Rate responses such that we know which one should be the winning & Losing Observation.
4) Minimize ↓ DPO loss.

Summary:
→ wp
→ pref data
→ RM {0, 1}
→ Reward model → PPO / DPO, GPO to backprop

17th November, 2025 Date: Pg 11

DeepSeek

Reasoning model:

Objective: → get good, clean, reliable, relevant task specific data.] crawle

2) Validate by using smaller (1.3 B) coder base LLM.

3) Once the hypothesis is validated → Train a LLM using SFT and GRPO.

→ How to get more math related data:

1) Go to commoncrawl

2) Seed Math Corpus (already established math datasets) such as Openweb Math. & others

- they created a model

Text → fasttext(n) → math / Not math.

50,000 samples from math, 50,000 random from crawler. → using negativehard, soft label mining.

- model was created using fastText to classify webpages as math or non-math.
- they created confusion matrix.

→ Math seed: math examples

→ it is easy to search domain wise in common crawl

→ Near Dedyplication, Simple duplication

→ on 4th epoch, 3rd iteration they found 98% unique math related pages. } scraping better approach

→ they got 40B Html pages

→ out of which 120B tokens

→ they used small model just to verify it.

⇒ Super-vised fine tuning & Chain of thought.

→ main reason for deepseek performance is deepseek's dataset & SFT + chain of thought.

20th November, 2025 Date:

SAT Benchmark
↳ Score = 84.4%

DeepSeek - Math 7-B Base

Contributions →
- New Math Benchmarkship dataset
- New state of the Art (SOTA) on MATH
- GRPO (Thinking model)

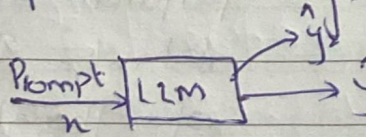
Achievement →
Better performance than 10 times Bigger Models on English & Chinese benchmarks.

→ Math } logically multistep
→ Code } solutions.

- code is verified using Test case verification.
- for math we can verify using solution as label.

=> Thinking Models: (Chatgpt o1)
- Preference Tunning: PPO, BON, DPO, GRPO
  ↳ with chain of thought

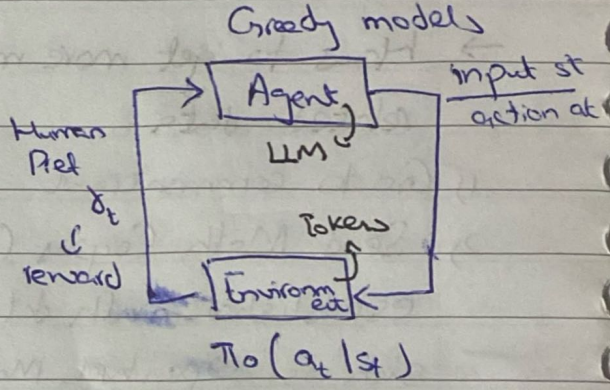
→ Proximal Policy Optimization:
- preference training.



- change the temp of LLM.
- with new temprature, a new output was calculated.
- we only made data for no, no label (preffered ones) right, wrong

→ For labelling:
- Human Pref
- LLM as judge
- Test case verification.

→ RLHF (Reinforcement Learning with human feedback)



→ tokens are set of vocabulary, out of which LLM will generate based on rule (policy).
→ action is choosing token.

* In RL we want to update weight of LLM based on human preference.
- slight update to meet human preference.

Date:

RLHF:

- 2 components: ① Reward model, ② RL

→ Objective of Reward model:
distinguish good from bad

- RM → Input (Prompt, HF)
- Output → $\gamma(x, y)$ a number 0-1

Issue → we cannot use these loss functions.
Cross entropy, RMSF, MSF ... ets.

- they only recognise the winner solution, they cannot distinguish b/w slight differences

solution → ~~Bretty~~ Bradley Terry Formulation: gives clear distinction

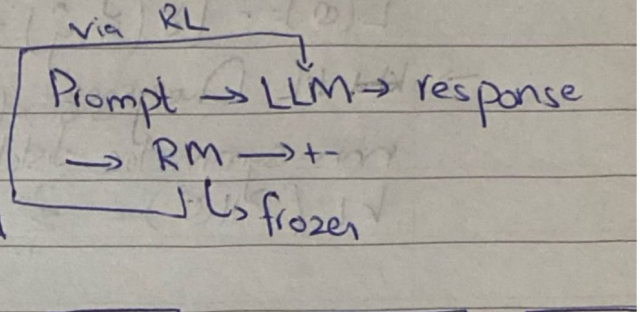
- Prob that y_i is better than y_j is given as:

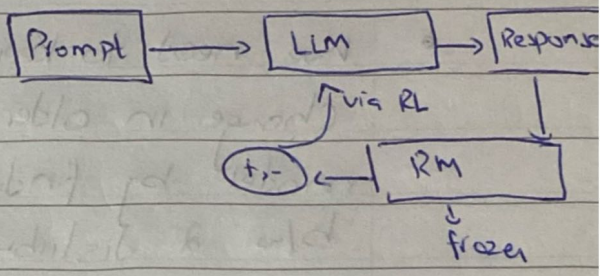
$$P(y_i > y_j) = \frac{e^{\gamma_i}}{e^{\gamma_i} + e^{\gamma_j}}$$

$$= \sigma(\gamma_i - \gamma_j)$$

- sigmoid gives value between 0 & 1.

Architecture:





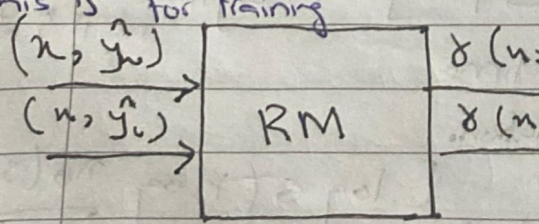
② Reinforcement Learning:

- The idea of RL step is to use RM to promote good answer.

→ Objective:
$L(\alpha)$ = maximize Reward + don't deviate too much from the base / Ref / old model.

- Reward hacking: too much deviation → learning instabili

This is for training



$$L(\alpha) = -E\left[\log\left(\sigma\left(\gamma(x, y_w) - \gamma(x, y_l)\right)\right)\right]$$

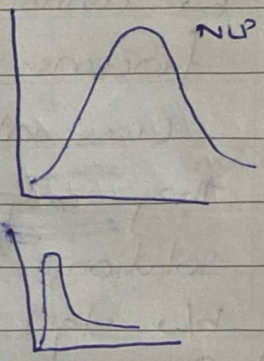
Date:

Proximal Policy Optimization: (β trainable controlling parameter)

$$L(\theta) = -\left[\gamma(x, \hat{y}) - \beta KL\left(\pi_\theta(\hat{y}/x) \,\|\, \pi_{\theta_{Ref}}(\hat{y}/x)\right)\right]$$

→ KL Divergence: sort of non-symmetric distribution b/w 2 dist

$$KL(P\|Q) = \sum_{i=1} P_i \log(P_i/Q_i) \rightarrow P_i(P_i - Q_i)$$



- we need to keep check b/w change in older & current model by finding distance b/w 2 distributions.

→ Issues with PPOs.

- we are calculating reward for each prompt.

Advantage ≅ reward − baseline.

PPO-KL:

$$L(\theta) = E\left[\frac{\pi_\theta(a_t/s_t)}{\pi_{\theta_{ref}}(a_t/s_t)} A_t - \beta KL\left(\pi_{ref}(\cdot/s_t), \pi_\theta(\cdot/s_t)\right)\right]$$

(1: π_{ref}, 2: π_θ, 3: A_t, 4: ratio)

→ 4 models → issue.

Solution ⇒ Best of N.

$x \rightarrow$ [SFT] $\rightarrow y_1, y_2, y_3 \cdots \hat{y}_n \rightarrow$ [RM, frozen] $\rightarrow k$ $\underset{i \in \|\hat{y}, N\|}{\arg\max}\ \delta(x, y)$

- super complex at inference time

→ DPO: Direct Preference Optimization.

$$L(\pi_\theta; \pi_{\theta_{ref}}) = -E_{(x, y_w, y_l) \sim D}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w/x)}{\pi_{old}(y_w/x)} - \beta \frac{\pi_\theta(y_l/x)}{\pi_{old}(y_l/x)}\right)\right.$$

Date:

usually perform better

PPO ⟷ DPO → This perform acceptable

computationally expensive

- distribution shift due to RM less examples.

⇒ DeepSeek R1 → they changed responses with thinking steps → solution

Order:

→ Preference Training RLHF

→ Proximal Policy Optimization ← actor-critic algo

→ Bag of N

→ Direct preference Optimization

→ DeepSeek Math.

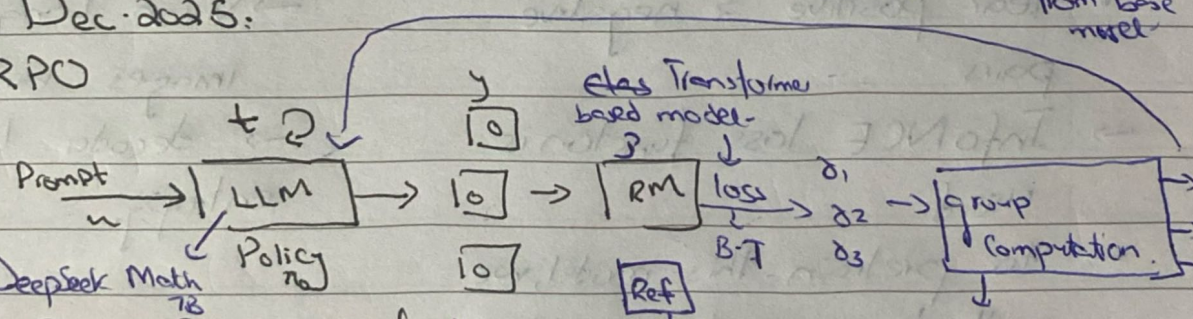
Group Relative Policy Opt

1st Dec. 2025:

GRPO

KL divergence

$K(\theta)$ = ~~maximize~~ maximize reward + don't deviate too much from base model



any scoring criteria → average, mean to low

* RM: Quantification of the input observation (o-1)
* if we have multiple output for the same source:

$x \to f \to \begin{matrix} O_1 \\ O_2 \\ O_3 \end{matrix} \to P(O_1 > O_2)$ B.T

$\ell(\theta) = \sum (\delta_1 - \delta_2)$

* classification & Quantification only 0 or 1
cat or dog

$$L_{GRPO}(\theta) = E\left[q \sim P(Q),\ \{o\}_{i=1}^{G} \in \pi_\theta(O|q)\right]$$

prompts — output Data

$$= \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{i=1}^{|o_i|}\left\{\min\left\{\frac{\pi_\theta(o|q)}{\pi_{old}(o|q)}\hat{A},\ clip\left(\frac{\pi_\theta(o|q)}{\pi_{old}(o|q)},\ 1+\varepsilon, 1-\varepsilon\right)A\right\} - \beta\, KL(\pi_\theta/\pi_{\theta_{old}})\right\}$$

this — or this

$|o_i|$ → average size of prompt output

select among min, the change,

→ Thinking models:

→ 3rd December, 2026:

Meme related → BLIP2

ViT, CNN

Encoder/Decoder base LLM

→ VLM vision Language models.

→ Contrastive Learning:

- lear positive & negative pairs

→ InfoNCE loss function is used.

→ to pretrain this model, we will have input pair like this (text, img) — label

→ CLIP. dual encoder model.

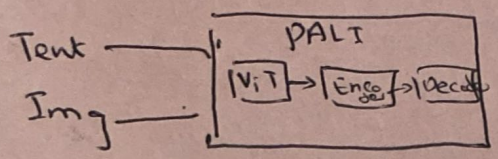
→ Florence., SAM — segmentation

→ MFCC → Cepstreal coefficients.

→ Vision Language Model: images with text

→ decoder loss is → cross entropy.

→ WAN - Animate.
→ Generative Adversarial Networks
→ Variational autoencoders
→ Diffusion networks.
=> PALI :



=> mixture of object.
- Scaling 2 loss functions

⇒ VQAS
- Visual, Questions & Answer
  Image (input) text (input) output