Several different RL algos and reward models

- RL algos:
  - REINFORCE
    - $J(\theta) = A^{KL}(x,y) \log P_\theta(x,y)$
    - A^{KL} is $r(x,y) - \beta D_{KL}(P_\theta \| P_{SFT})$
  - PPO
    - $J(\theta) = \min(\text{clip}(\rho_i) A^{KL}, \text{unclip}(\rho_i) A^{KL})$
      - ρ_i = policy ratio between P_θ and $P_{\theta_{OLD}}$
      - key diff: multiple updates per batch
      - baseline: trained value fn
  - GRPO
    - very similar to PPO
      - key diff: no value fn

↳ requires performing G rollouts per prompt x, not 1 like in PPO/REINFORCE

↳ advantage is normalized by group stats

$$A^{GR}(x, y_i) = \frac{r(x, y_i) - \mu_x}{\sigma_x + \varepsilon}$$

↳ you can use REINFORCE, PPO, GRPO with any kind of reward $r(x, y)$

Reward models

↳ Bradley Terry preference model (RLHF)

↳ train $r(x, y)$ on a big dataset of human pref judgments

↳ humans rate responses y_i gen. from the SFT model

↳ given prompt x and two or more responses y_i) human will produce a ranking $y_w > y_L$

↳ train reward model to make

$r(x, y_w) > r(x, y_l)$

↳ verifiable rewards (RLVR)

↳ cheaply / quickly obtainable

↳ no learned reward model, rewards come from environment

↳ correctness

↳ formatting

↳ execution-based rewards

e.g. coding

Examining GRPO advantage:

$r(x, y_i)$ = reward

$A_i^{raw} = r(x, y_i)$ ⇒ no baseline

$A_i^{N} = r(x, y_i) - N_x$

$A_i^{GR} = \frac{r(x, y_i) - N_x}{\sigma_x}$

rare success:

↳ rewards: $[1, 0, 0, 0, 0, 0, 0, 0]$

↳ $N_x = 0.125$, $\sigma_x = 0.33$

	A_i^{raw}	A_i^N	A_i^{GR}
winners	+1	+0.875	+2.647
losers	0 ~ no update	−0.125	−0.38

↳ A^{GR} puts more emphasis on rare success.

higher variance group:

rewards: $[6, 0, 0, 0, 0, 0, 0, 0]$

$N_x = 0.75$, $\sigma_x = 1.98$

	A_i^{raw}	A_i^N	A_i^{GR}
winners	+6	+5.25	+2.647
losers	0	−0.75	−0.378