LLM Alignments

Zhuoran Xu, Shihan Chen

April 29 2025

Preliminary: RLHF

- **Language Model**: Consider an LLM as a policy $\pi_\theta(y|x) = \prod_t \pi_\theta(y_t|x, y_{<t})$
- **Supervised Fine-Tuning (SFT)**: As an initial phase of alignment, the pre-trained model is enforced to imitate high-quality demonstration data.
- **Reinforcement Learning from Human Feedback (RLHF)**: To further align the SFT model π_θ with human preference by maximize:

$$L_r(\pi_\theta) = \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)}[r(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}]$$

  - Reward-based approach (Proximal Policy Optimization, PPO)
  - Reward-free approach (Direct Preference Optimization, DPO)

Preliminary: PPO

- **Reward-based approach (Proximal Policy Optimization, PPO**:
  - Collect a dataset of $D = \{(x, y_w, y_l)\}$
  - Train a reward model:

$$L_R(r_\phi) = -\mathbb{E}_{(x,y_w,y_l)}[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))]$$

  - Optimize:

$$L_r(\pi_\theta) = \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)}[r_\phi(x, y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}]$$

Preliminary: DPO

- **Reward-free approach (Direct Preference Optimization, DPO**:
  - Instead of learning a reward model, DPO optimizes the policy over preference data, the optimal policy is:

$$\pi^*(y|x) = \pi_{ref}(y|x)\frac{exp(r(x,y)/\beta)}{Z(x)}$$

  with the underlying reward being

$$r_\phi(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} + C(x)$$

  - The DPO loss is:

$$L_{DPO}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim D}[\log \sigma(\beta(\log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}))]$$

Online RLHF

- RLHF techniques for aligning LLMs with human preferences have traditionally been offline methods
- Offline methods: static preference dataset of $D_{off} = \{(x, y_w, y_l)\}$
- Limitation: the finite dataset D_{off} fails to cover the entire prompt-response space
- The resulting policy model often performs poorly when faced with out-of-distribution data.

Online RLHF

Online iterative RLHF [Dong et al., 2024]

- Based on *D*, update the policy (LLM)
- Sample random prompts and generate new preferences
- Expand dataset

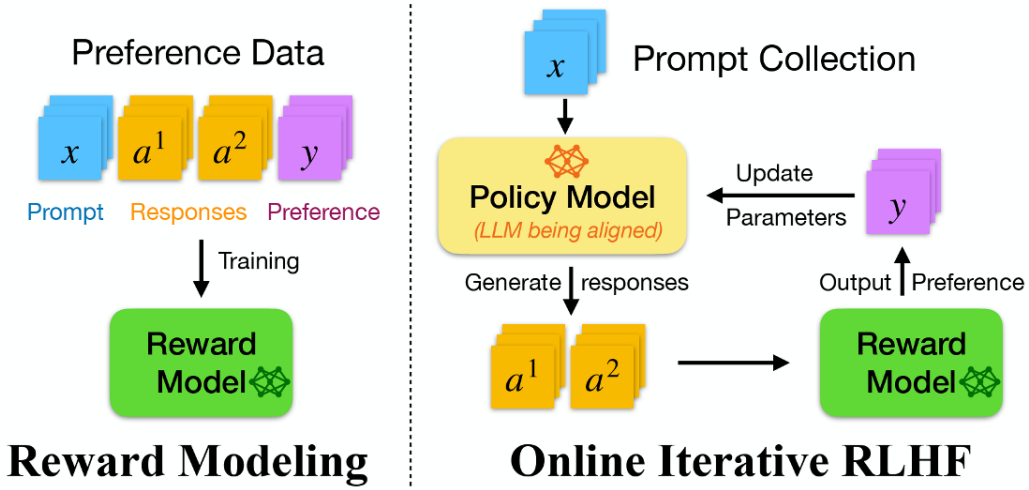


Figure 1: A simplified illustration of reward modeling and online iterative RLHF.

Online RLHF

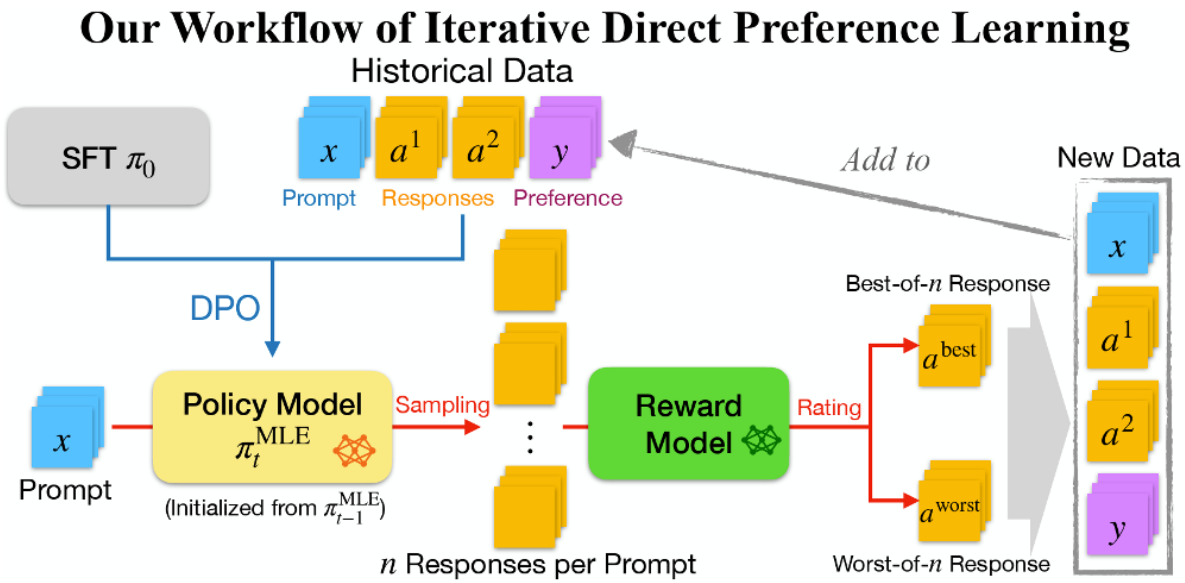


Figure 4: Illustration of our implementation of iterative direct preference learning. In iteration $t = 1$, the historical dataset is empty, and the resulting policy model π_1^{MLE} is the same as its initialization, π_0, which is the SFT model checkpoint. After that, the historical dataset grows with preference data collected from previous iterations.

Direct Human Preference Optimization

- Offline preference optimization methods have limitation of distribution mismatch between the training data and the data expected from the optimal policy.
- Rejection Sampling Optimization (RSO) [Liu et al., 2023]

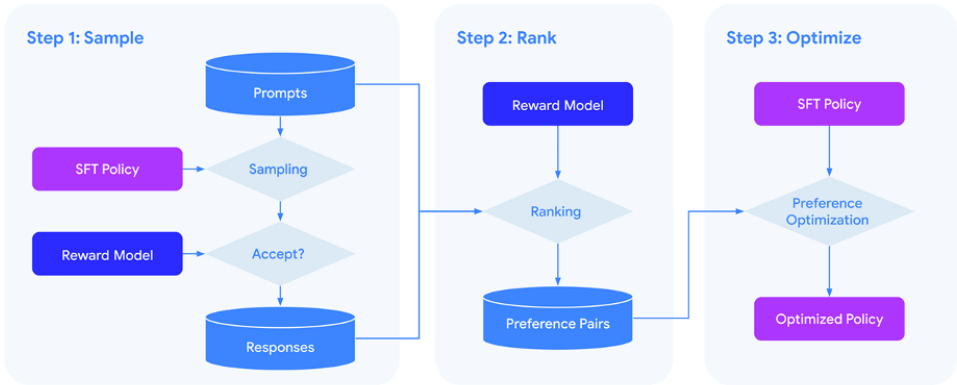
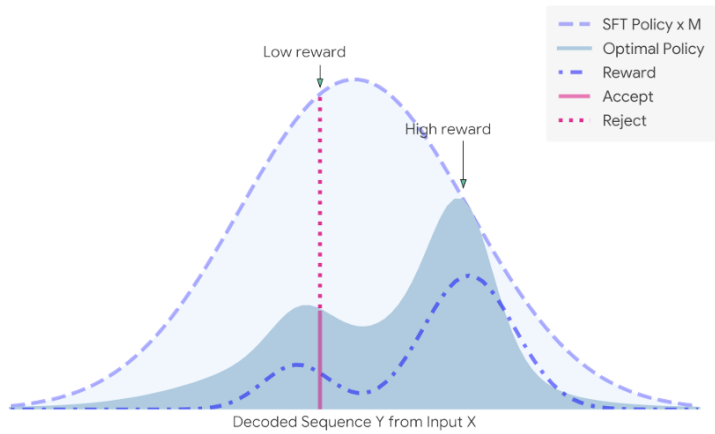


Figure 1: RSO first fits a pairwise reward-ranking model from human preference data. This model is later applied to generate preference pairs with candidates sampled from the optimal policy, followed by a preference optimization step to align sequence likelihood towards preferences.

Direct Human Preference Optimization

- Generate $y \sim \pi_{SFT}(y|x)$ and $u \sim U(0,1)$
- Calculate $M = min\{m|m \geq \frac{\pi_{SFT}(y|x)}{\pi_\theta(y|x)}\}$
- Accept y if $u < \frac{\pi_\theta(y|x)}{M\pi_{SFT}(y|x)}$
- Trained reward model to guide the sampling process.
- $\frac{\pi_\theta(y|x)}{M\pi_{SFT}(y|x)} \approx e^{\frac{r_\phi(x,y)-r_{max}}{\beta}}$
- $\beta \to \infty$ all accept
- $\beta \to 0$ accept highest reward



Direct Human Preference Optimization

Many methods aim to optimize LLM policies directly based on human preferences, without necessarily relying on a scalar reward signal.

- Sequence Likelikhood Calibration with Human Feedback (SliC-HF) [Zhao et al., 2023]

$$L_{SLiC-HF}(\pi_\theta) = max(0, \delta - \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}) - \lambda \log \pi_\theta(y_{ref}|X)$$

- δ serves as a margin to distinguish desired responses from undesired responses
- $\lambda \log \pi_\theta(y_{ref}|X)$ encourage the trained model to stay close to the initial SFT policy.

Direct Human Preference Optimization

- DPO loss aimed to maximize the disparity between desired and undesired responses.
- This might lead to simultaneous increases or decreases in the rewards for both desired and undesired responses
- DPO-Positive (Smaug) [Pal et al., 2024]

$$L_{DPOP}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\{\log[\sigma(\beta\log(\frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}) - \beta\log(\frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}))]$$
$$- \lambda max(0, \log(\frac{\pi_{ref}(y_w|x)}{\pi_\theta(y_w|x)}))\}$$

- The second term could prevent the reduction in rewards for desired responses

Other Alignment Research

- Binary feedback: KTO [Ethayarajh et al., 2024]
- Merge SFT and Alignment: ORPO [Hong et al., 2024]
- Learning Dynamics of LLM Finetuning [Ren and Sutherland, 2025]

Binary Feedback

- Kahneman and Tversky's prospect theory: how humans made decisions under uncertain events did not maximize expected value owing to loss aversion.
- Value functions:

$$v(z) = \begin{cases} (z - z_0)^{\alpha} & \text{if } z \geq z_0 \\ -\lambda(z_0 - z)^{\alpha} & \text{if } z < z_0 \end{cases}$$

  α controls the curvature of the function, which reflects risk aversion; λ controls its steepness, which reflects loss aversion.
- Existence of a reference point that is used to get the relative gain or loss; concavity in relative gains (i.e., diminishing sensitivity away from z_0); and loss aversion (i.e., greater sensitivity to losses).

Human-aware Losses (HALOs)

Definition 3.4 (HALOs). Let θ denote the trainable parameters of the model $\pi_\theta : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ being aligned, π_ref the reference model, $l : \mathcal{Y} \to \mathbb{R}^+$ a normalizing factor, and

$$r_0(x, y) = l(y) \log \left(\frac{\pi_\theta(y \mid x)}{\pi_\text{ref}(y \mid x)} \right)$$

the implied reward, where $Q(Y' \mid x)$ is a reference point distribution over $\mathcal{Y}$ and $v : \mathbb{R} \to \mathbb{R}$ is non-decreasing everywhere and concave in $(0, \infty)$. The *human value* of (x, y) is

$$v\left(r_0(x, y) - \mathbb{E}_Q[r_0(x, y')]\right)$$

A function f is a *human-aware loss* for v if there exists $a_{x,y} \in \{-1, +1\}$ such that:

$$f(\pi_\theta, \pi_\text{ref}) = \mathbb{E}_{x,y\sim\mathcal{D}}\left[a_{x,y} v\left(r_\theta(x, y) - \mathbb{E}_Q\left[r_\theta(x, y')\right]\right)\right] + C_\mathcal{D} \tag{1}$$

where $\mathcal{D}$ is the feedback data and $C_\mathcal{D} \in \mathbb{R}$ is a data-specific constant.

DPO and PPO-Clip are human-aware losses

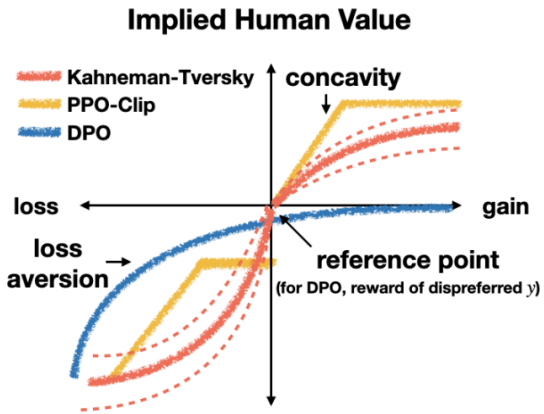


Figure 1. The utility that a human gets from the outcome of a random variable, as implied by different human-aware losses (HALOs). Notice that the implied value functions share properties such as loss aversion with the canonical human value function in prospect theory (Tversky & Kahneman, 1992).

Does being a HALO matter?

- HALOs either match or outperform non-HALOs at every scale.
- Up to a scale of 7B parameters, alignment provides virtually no gains over SFT alone.
- Despite only using dummy +1/-1 rewards, our offline PPO variant performs as well as DPO

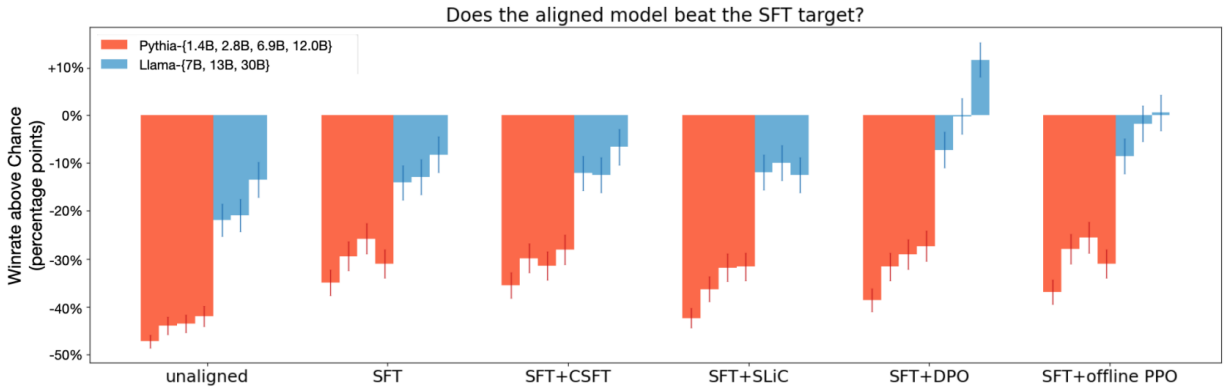


Figure 2. HALOs (DPO, offline PPO variant) outperform non-HALOs (SLiC, CSFT), as measured by the `GPT-4-0613`-judged winrate of the aligned model's generations against a hard-to-beat baseline: the outputs that would have been used as the targets for SFT. The y-axis here plots the winrate above chance (i.e., the winrate – 50%). The difference between methods is only significant ($p < 0.05$) at 13B+ parameters, and only the HALO-aligned Llama-{13B, 30B} models are able to match the baseline and yield a winrate at or above chance.

KTO

- Default KTO loss

$$L_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{x,y\sim D}\left[\lambda_y - v(x,y)\right] \tag{8}$$

where

$$r_\theta(x,y) = \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}$$

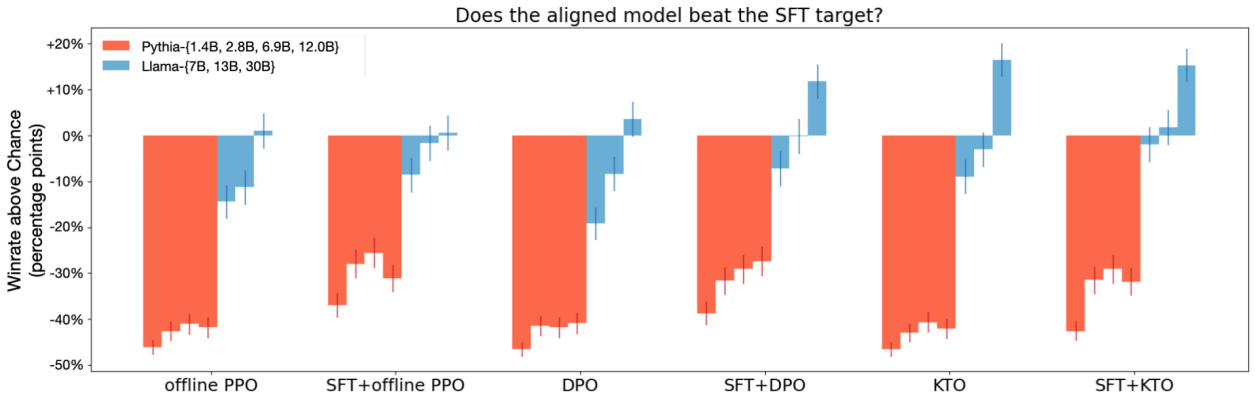
$$z_0 = \mathrm{KL}\left(\pi_\theta(y' \mid x) \,\|\, \pi_{\text{ref}}(y' \mid x)\right)$$

$$v(x,y) = \begin{cases} \lambda_D\, \sigma\left(\beta\left(r_\theta(x,y) - z_0\right)\right) & \text{if } y \sim y_{\text{desirable}} \mid x \\ \lambda_U\, \sigma\left(\beta\left(z_0 - r_\theta(x,y)\right)\right) & \text{if } y \sim y_{\text{undesirable}} \mid x \end{cases}$$

- Replace *withlogisticfunction*σ in the value function for numerical stability; β controls the degree of risk aversion; λ_D, λ_U are hyperparameters for desirable and undesirable outputs respectively.

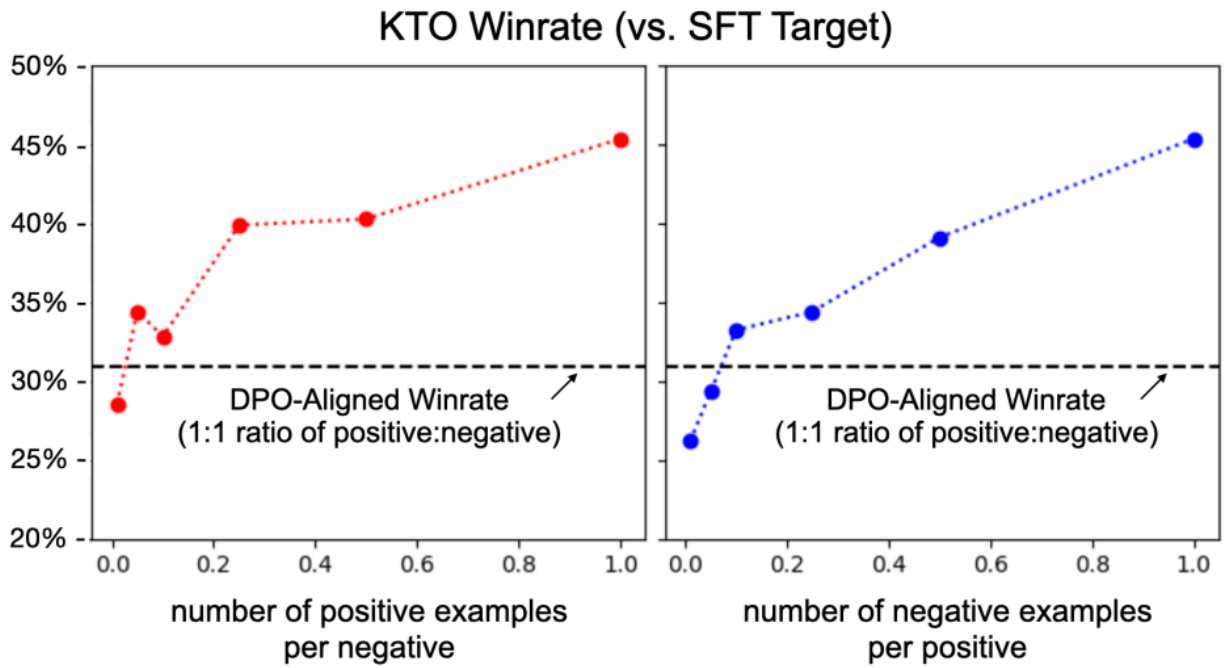
KTO vs. DPO

- KTO ≥ DPO
- At sufficient scale, KTO does not need SFT. A KTO-aligned Llama-13B, 30B model is competitive with its SFT+KTO counterpart.
- KTO data need not come from preferences.



KTO vs. DPO

- The optimal range for $\frac{\lambda_D n_D}{\lambda_U n_U}$ between 1 and 4/3 could deal with data imbalance optimally, where n_D and n_U represented the quantities of desired and undesired samples.



Merge SFT and Alignment

- Previous research primarily concentrated on sequentially applying SFT and alignment, a method that proved to be laborious and led to catastrophic forgetting. The subsequent studies either integrated these two processes into a single step or performed fine-tuning in parallel and merged the two model at the end.
- Odds Ratio Preference Optimization (ORPO) [Hong et al., 2024]

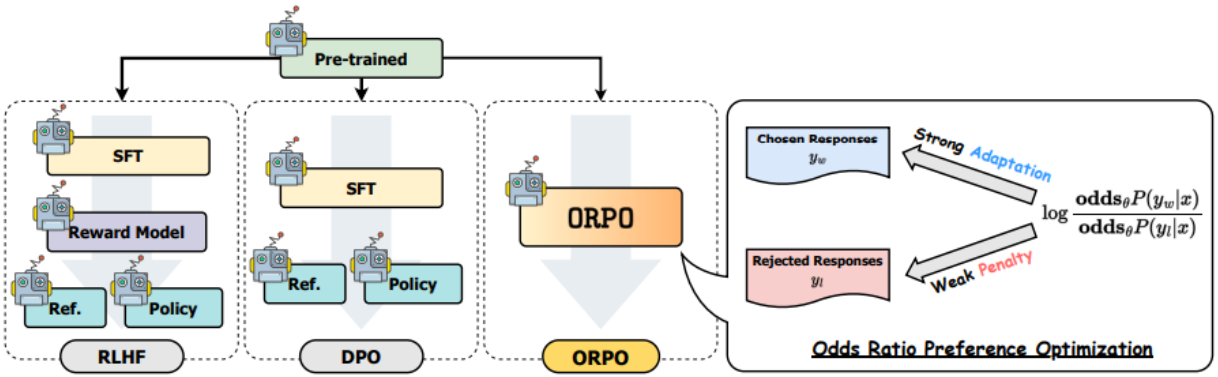


Figure 2: Comparison of model alignment techniques. ORPO aligns the language model *without a reference model* in a single-step manner by assigning a weak penalty to the rejected responses and a strong adaptation signal to the chosen responses with a simple log odds ratio term appended to the negative log-likelihood loss.

Absence of Penalty in Cross-Entropy Loss

$$\mathcal{L} = -\frac{1}{m}\sum_{k=1}^{m}\log P(x^{(k)}, y^{(k)}) \tag{2}$$

$$= -\frac{1}{m}\sum_{k=1}^{m}\sum_{i=1}^{|V|} y_i^{(k)} \cdot \log(p_i^{(k)}) \tag{3}$$

y_i indicates if *ith* token is a label token; p_i refers to the probability of the *ith* token

Odds Ratio Preference Optimization

- The likelihood for the model θ to generate the output sequence y than not generating it.

$$\text{odds}_\theta(y \mid x) = \frac{P_\theta(y \mid x)}{1 - P_\theta(y \mid x)} \tag{4}$$

- $\text{OR}_\theta(y_w, y_l)$ indicates how much more likely it is for the model θ to generate y_w than y_l given input x.

$$\text{OR}_\theta(y_w, y_l) = \frac{\text{odds}_\theta(y_w \mid x)}{\text{odds}_\theta(y_l \mid x)} \tag{5}$$

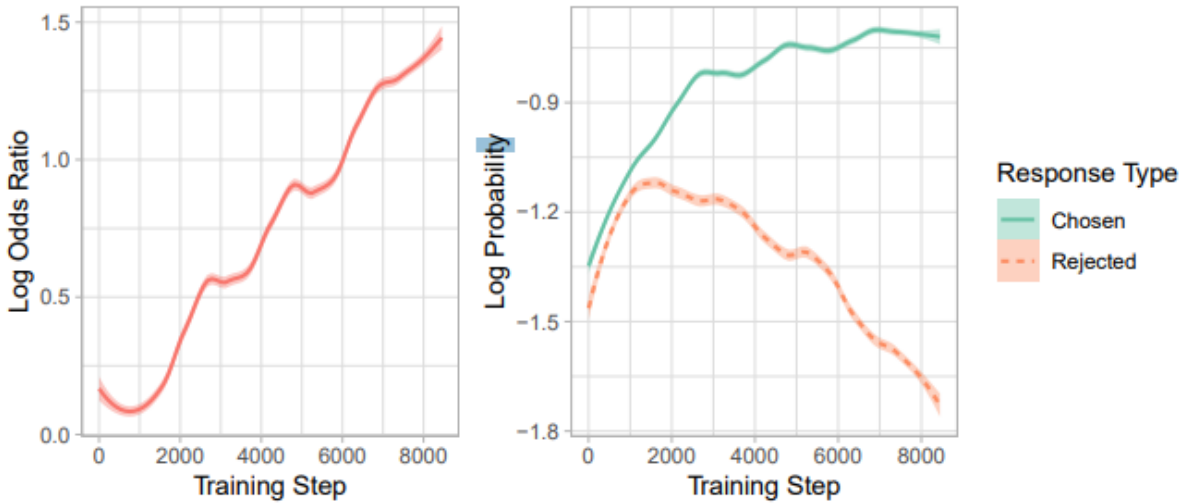
- Objective Function of ORPO

$$\mathcal{L}_{\text{ORPO}} = \mathbb{E}_{(x, y_w, y_l)} \left[\mathcal{L}_{\text{SFT}} + \lambda \cdot \mathcal{L}_{\text{OR}} \right] \tag{6}$$

$$\mathcal{L}_{\text{OR}} = -\log \sigma \left(\log \frac{\text{odds}_\theta(y_w \mid x)}{\text{odds}_\theta(y_l \mid x)} \right) \tag{7}$$

ORPO

- Lower the likelihood of unwanted generations with ORPO.
- This approach is ineffective for SFT datasets where only y_w (Desired responses).
- Some experiments on Mistral and Llama-3 indicated that the performance of ORPO is inferior to that of DPO is present.



Learning Dynamics

- How the change in model's parameter θ influences the corresponding change in $f(\theta)$? After an GD update on x_u, how does the model's prediction on x_o change?
- Gradient descent:

$$\Delta\theta = -\eta \cdot \nabla \mathcal{L}(\pi_\theta(x_u), y_u)$$

- First-order Taylor expansions:

$$\log \pi_{\theta_{t+1}}(y \mid \mathbf{x}_o) - \log \pi_{\theta_t}(y \mid \mathbf{x}_o) = \langle \nabla_\theta \log \pi_{\theta_t}, \Delta\theta \rangle + O$$

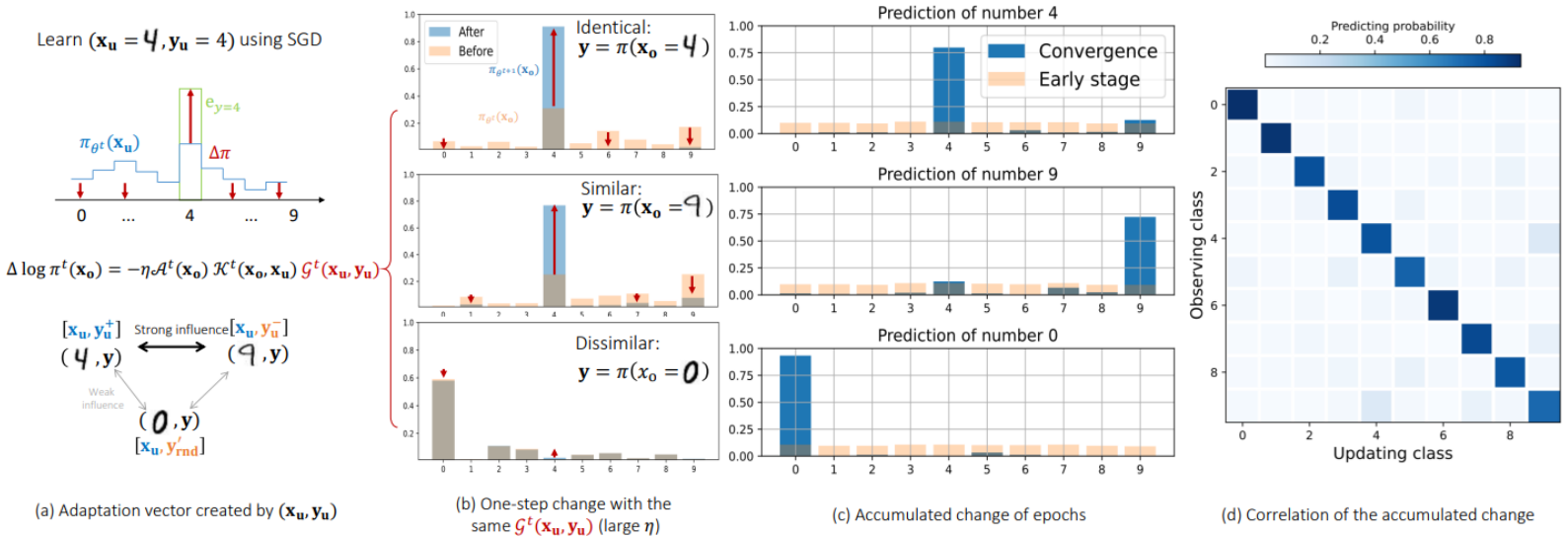
- One-step learning dynamics decomposition:

$$\Delta \log \pi_{\theta_t}(y \mid \mathbf{x}_o) = -\eta \mathcal{A}^t(\mathbf{x}_o)\mathcal{K}^t(\mathbf{x}_o, \mathbf{x}_u)\mathcal{G}^t(\mathbf{x}_u, y_u)$$

- $\mathcal{K}^t$ is the empirical neural tangent kernel of the logit network z (model-specific similarity measurement); $\mathcal{G}^t$ provides the energy and direction for the model's adaptation

A Warm-up MNIST Example

- A simple example of training a LeNet on the MNIST dataset
- Predicted probability would be pulled up of x_u and x_o are similar (when making predictions on images coming from class 4, the model tends to assign higher confidence on class 9).



(a) Adaptation vector created by $(\mathbf{x_u}, \mathbf{y_u})$

(b) One-step change with the same $\mathcal{G}^t(\mathbf{x_u}, \mathbf{y_u})$ (large η)

(c) Accumulated change of epochs

(d) Correlation of the accumulated change

Learning Dynamics of LLM's Finetuning

- Auto-regressive!

$$\mathcal{L}_{\text{SFT}} \triangleq -\log \mathbf{z} = -\log \pi_\theta(\mathbf{y}|\mathbf{x}) = -\sum \log \pi_\theta(y_l \mid \mathbf{x}, \mathbf{y}_{<l})$$

- Simplified using teacher forcing:

$$\chi = [\mathbf{x}; \mathbf{y}], \quad \mathbf{z} = h_\theta(\chi), \quad \pi_\theta(\mathbf{y}|\chi) = \text{Softmax}(\mathbf{z})$$

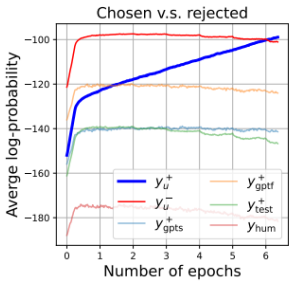
- Similar decomposition:

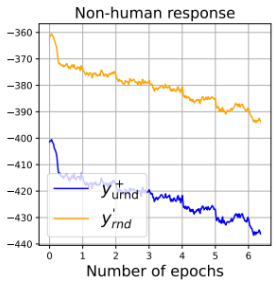
$$[\Delta \log \pi^t(y|\chi_o)]_m = -\sum_{l=1}^{L} \eta [\mathcal{A}^t(\chi_o)]_m [\mathcal{K}^t(\chi_o, \chi_u)]_l [\mathcal{G}(x_u)]_l + \mathcal{O}(\eta^2)$$

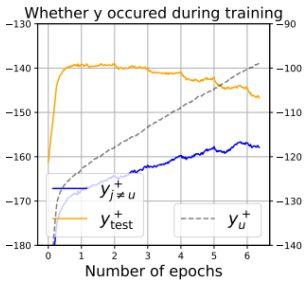
- For a prompt x_u, how does learning the response y_u^+ influence the model's belief about a response $y_u^{'}$?

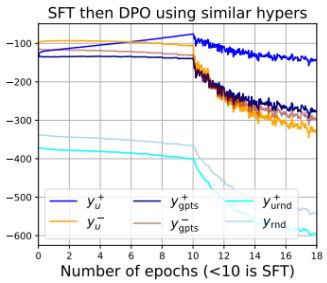
Learning Dynamics of SFT

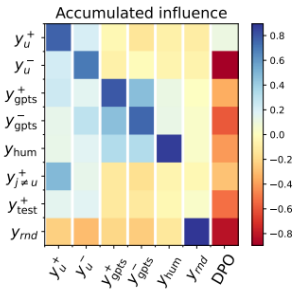
- Desired response increases; others increase than decrease.
- $[x_u, y_{j\neq u}]$ increases. Hallucination.
- All responses generated by ChatGPT are considered very similar to each other, regardless of how semantically different they are.











A commen framework

- The gradients of various LLM finetuning algorithms have a similar structure.

$$\nabla_\theta \log \pi_\theta(y|\chi_u)$$

1. SFT: quite straight forward
2. DPO and its variants: need a bit calculation

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(\mathbf{x}_u, \mathbf{y}_u^+, \mathbf{y}_u^-)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_{\theta^t}(\mathbf{y}_u^+ \mid \chi_u^+)}{\pi_{\text{ref}}(\mathbf{y}_u^+ \mid \chi_u^+)} - \beta\log\frac{\pi_{\theta^t}(\mathbf{y}_u^- \mid \chi_u^-)}{\pi_{\text{ref}}(\mathbf{y}_u^- \mid \chi_u^-)}\right)\right]$$

$$\mathcal{L}_{\text{SLiC}} = \mathbb{E}_{(\mathbf{x}_u, \mathbf{y}_u^+, \mathbf{y}_u^-)\sim\mathcal{D}}\left[\max\left[0, \delta - \log\frac{\pi_{\theta^t}(\mathbf{y}_u^+ \mid \chi_u^+)}{\pi_{\theta^t}(\mathbf{y}_u^- \mid \chi_u^-)}\right] - \beta\cdot\log\pi_{\theta^t}(\mathbf{y}_{\text{ref}} \mid \chi_{\text{ref}})\right]$$

3. RL (PPO, GRPO): under approximation

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q\sim P(Q), o\sim\pi_{\theta_{old}}(O|q)]\frac{1}{|o|}\sum_{t=1}^{|o|}\min\left[\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}A_t, \text{clip}\left(\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}, 1-\varepsilon, 1+\varepsilon\right)A_t\right].$$

$$\frac{\nabla\pi_\theta(y|\chi)}{\pi_{old}(y|\chi)} = \frac{\pi_\theta(y|\chi)}{\pi_{old}(y|\chi)}\nabla_\theta \log\pi_\theta(y|\chi_u)$$

Negative Gradients and Squeezing Effect

- Analysis of DPO:

$$-\sum_{l=1}^{L}\eta\left[\mathcal{A}^t(\chi_o)\right]_m\left[\mathcal{K}^t(\chi_o,\chi_u^+)\mathcal{G}^t_{\mathrm{DPO}+}-\mathcal{K}^t(\chi_o,\chi_u^-)\mathcal{G}^t_{\mathrm{DPO}-}\right]_l$$

$$\mathcal{G}^t_{\mathrm{DPO}+}=\beta(1-\alpha)(\pi_{\theta_t}(y\mid\chi_u^+)-y_u^+),\quad \mathcal{G}^t_{\mathrm{DPO}-}=\beta(1-\alpha)(\pi_{\theta_t}(y\mid\chi_u^-)-y_u^-)$$

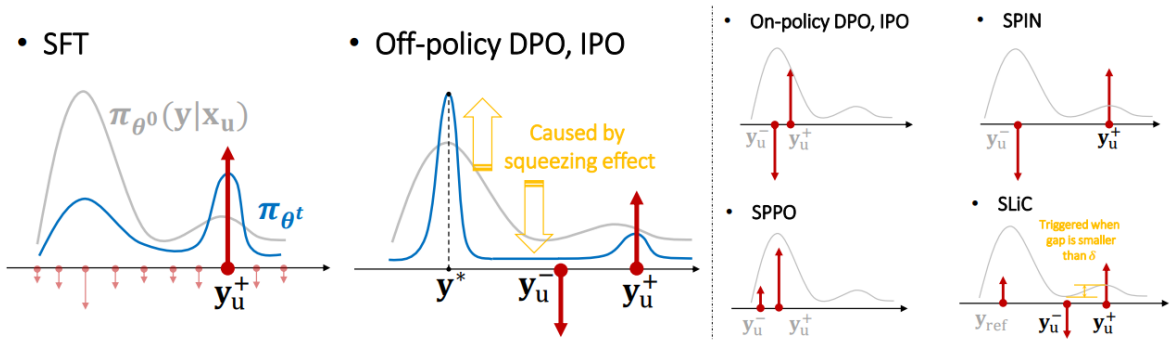
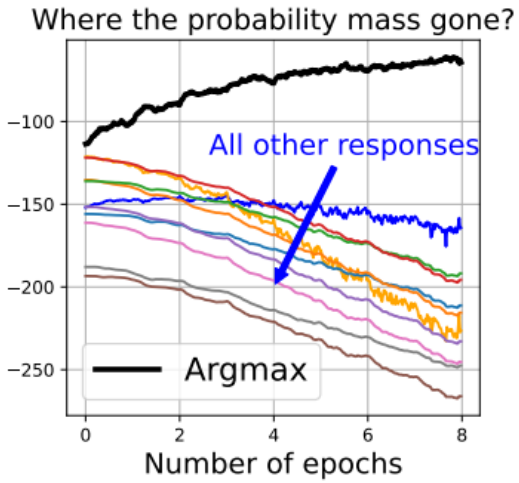


Figure 2: The updating vector provided by the residual term $\mathcal{G}^t$ of different algorithms. The gray **y** are responses *sampled* from π in an on-policy way. In the second panel, we demonstrate the "squeezing effect" caused by imposing a big negative gradient on a "valley" region of a distribution. For more details about this counter-intuitive effect, please refer to Section 3.3 and Appendix E. Other panels demonstrate on-policy DPO (and IPO), SPIN (Z. Chen et al. 2024), SPPO (Y. Wu et al. 2024), and SLiC (Y. Zhao et al. 2023).

Learning Dynamics of DPO

- Verification of the squeezing effect. This could be a reasonable explanation of the "degeneration" reported in recent work (e.g. Holtzman et al. 2020): as θ^t becomes more peaky at its most confident predictions, it is easier to sample sequences with repeated phrases.



Dong, H., Xiong, W., Pang, B., Wang, H., Zhao, H., Zhou, Y., Jiang, N., Sahoo, D., Xiong, C., and Zhang, T. (2024).
Rlhf workflow: From reward modeling to online rlhf.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. (2024).
Kto: Model alignment as prospect theoretic optimization.

Hong, J., Lee, N., and Thorne, J. (2024).
Orpo: Monolithic preference optimization without reference model.

Liu, T., Zhao, Y., Joshi, R., Khalman, M., Saleh, M., Liu, P. J., and Liu, J. (2023).
Statistical rejection sampling improves preference optimization.

Pal, A., Karkhanis, D., Dooley, S., Roberts, M., Naidu, S., and White, C. (2024).
Smaug: Fixing failure modes of preference optimisation with dpo-positive.

Ren, Y. and Sutherland, D. J. (2025).
Learning dynamics of llm finetuning.

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., and Liu, P. J. (2023).
Slic-hf: Sequence likelihood calibration with human feedback.