Language Alignment via Nash-learning and Adaptive feedback

Ari Azarafrooz [1] **Farshid Faal** [1]

Abstract

Recent research has shown the potential of Nash Learning via Human Feedback for large language model alignment by incorporating the notion of a preference model in a minimax game setup.

We take this idea further by casting the alignment as a mirror descent algorithm against the *adaptive feedback* of an *improved opponent*, thereby removing the need for learning a preference model or the existence of an annotated dataset altogether.

The resulting algorithm, which we refer to as **L**anguage **A**lignment via **N**ash-learning and **A**daptive feedback (LANA), is capable of self-alignment without the need for a human-annotated preference dataset. We support this statement with various experiments and mathematical discussion.

1. Introduction

The standard approach for Large Language Model (LLM) alignment involves optimizing a reward function that is learned explicitly (RLHF) (Christiano et al., 2017) or implicitly (DPO) (Rafailov et al., 2024) by accessing human-generated feedback. Other alternatives integrate human-generated feedback by learning a preference model that takes two responses, denoted as y and y' (conditioned on a prompt x), as input and produces a preference score (a number between 0 and 1), indicating the preference of response y over response y' given the context/prompt x. These preference models are then cast as the utility of a game-theoretic framework, leading to the notion of Nash Learning via Human Feedback (NLHF) (Munos et al., 2023). The solution offered by the Nash equilibrium of the preference model is argued to be more aligned with the diversity of human preferences.

Within a similar minimax game setup, we propose another alternative that uses the *adaptive* feedback of an *improved* opponent without the need for a fixed/learned preference model or pre-generated preference data. This is similar to the transition from RLHF to self-reward DPO (Yuan et al., 2024). (Yuan et al., 2024) proposed an offline self-feedback procedure for generating new data for DPO to incorporate into further alignment training. However, no training methodology has yet been proposed to directly incorporate self-evaluating reward functions in the alignment training. This is because self-evaluation could still be noisy and lead to biased and sub-optimal demonstrations. Therefore, learning from such data directly does not guarantee a better optimal model. The main contribution of this paper is to show that self-reward training processes exist that are robust to sub-optimal and noisy iterative self-reward mechanisms.

Comparison with related works Two important distinctions of our work compared to related works (Munos et al., 2023; Rosset et al., 2024; Yuan et al., 2024; Wu et al., 2024) are:

- All the previous works assume that a preference model is learned in advance, analogous to the concept of reward models in RLHF. However, in our setup we assume we lack access to such a learned preference model or human-annotated preference dataset. Instead, the LLM policies improve through ***adaptive feedback from improved opponents.***
- All the existing works avoid a faithful game-theoretic implementation, such as the two-timescale update, to avoid complex hyperparameter tuning and unstable performance. While this might be true in a generic game-theoretic setup, it seems to be overthinking in the context of LLM alignment. This is because complex policy behaviors are inherently avoidable as a result of a shared common worldview learned in the initial foundation model.

Aside from the self-evaluating assumption, our work is also different from (Munos et al., 2023) in that it sets up a modified Mirror Descent algorithm to incorporate the KL regularization with respect to the reference policy. However, our proposed method is reference policy-free.

(Chen et al., 2024) proposed self-play in a supervised fine-tuning (SFT) context and not in alignment training.

[1]CA, USA. Correspondence to: Ari Azarafrooz <ari.azarafrooz@gmail.com>.

ICML 2024 Workshop on Models of Human Feedback for AI Alignment, Vienna, Austria.

2. Language Alignment via Nash-learning and Adaptive feedback

We derive the new alignment algorithm using mirror ascent algorithm with improved opponent (MAIO) (Munos et al., 2020). It defines a sequence of policies $(\pi_{i,t})_{t\geq 0}$ for a zero-sum game according to the following updates for all $i \in 1, 2$ and for all $t \geq 0$:

$$\pi_{i,t+1} = \arg \max_{\pi_i \in \Delta(\mathcal{Y})} [\gamma_t \pi_i . Q_i^{\tilde{\pi}_{-i,t}} - D_\phi(\pi_i, \pi_{i,t})] \quad (1)$$

where γ_t is a learning rate, D_ϕ is a Bregman divergence, more specifically a KL distance in our case and $Q_i^{\tilde{\pi}_{-i,t}}$ is the reward of the player i against the improved opponent $\tilde{\pi}_{-i,t}$

2.1. Sampling from improved Opponent $\tilde{\pi}$

Improved policies can take different forms, such as greedy, best response, MCTS, extra-gradient method, etc.

The improved policy might also be an optimally aligned model π_{Expert}. However, we assume that we haven't learned such a model yet. In other words, we haven't trained a preference/reward model in advance, and we are learning this as the game progresses. We hypothesize that one might rely on the self-evaluation of the LLM policy to derive samples from such an improved policy. For example, for a given prompt x, we sample two responses y and y'.

Each player generates two answers and queries the *opponent* using the following evaluation prompt template:
"User
Given a piece of instruction and two of its possible responses, output 1 or 2 to indicate which response is better.
Instruction: instruction,
Response 1: y
Response 2: y'?
Assistant
Preferred response is -".

y is the *preferred* answer for user i if:
π_{-i}(eval prompt)$[-1][tokenizer("1")] >$
π_{-i}(eval prompt)$[-1][tokenizer("2")]$ and y' otherwise.

Every player then treats the preferred answer as the sample of an *improved* opponent and the *rejected* answer as their own, aiming to maximize their expected win rate under the setup described in equation 1 setup.

We show that while this setup leads to noisy outcomes (e.g., the opponent may be wrong) and changing utilities (the preference measure for the exact same response is not identical over the course of the game since the parameters get updated at every step), with the correct choice of proxy reward and slower learning dynamics, the game converges to a better policy.

Algorithm 1 LANA

Input: prompt distribution $\mathcal{X}$, An instruct-tuned LLM model policy π, eval prompt
Initialize $\pi_1, \pi_2 \leftarrow \pi$
repeat
  $x \sim \mathcal{X}$
  for $i = 1$ **to** 2 **do**
    $(y, y') \sim \pi_i(x)$
    Optimize π_i using SGD with loss:
    if π_{-i}(eval prompt) indicates $(y' \succ y)$ **then**
      loss $:= \log(\pi_{-i}(x, y')/\pi_i(x, y))$
    else
      loss $:= \log(\pi_{-i}(x, y)/\pi_i(x, y'))$
    end if
  end for
until Convergence or out of available compute resource

3. Algorithm

Let the reward be *adaptively* evaluated at each time t using the policies of the players as follows:

$$Q_i^{\tilde{\pi}_{-i,t}} = \log(\pi_{i,t}/\tilde{\pi}_{-i,t})/\gamma_t \quad (2)$$

In section 5.1, we show that if we plug this into the optimization Eq. 1, we end up with Algorithm 1 which we refer to as LANA, short for **L**anguage **A**lignment via **N**ash-learning and **A**daptive feedback. Note how it can alternatively be viewed as an online, two-player, simplified (reference-free, sigma-free) version of Direct Preference Optimization (DPO) without the need for a human-annotated preference dataset.

4. Experiments

4.1. Experiment Setup

4.1.1. Data

We randomly selected 3K prompts from (Argilla & Face, 2024). We **only** use the prompts and not the generated responses. For testing aside from common methodology we also utilized pre-processed version of the UltraFeedback test dataset (Cui et al., 2023) and different categories of (Huang et al., 2024) for deeper understanding.

4.1.2. Base model

We used the Phi-3-mini-4k-instruct (Abdin et al., 2024) for most of our experiments. This model is a mini model containing only 3.8b parameters which is helpful for faster

Table 1. Alpca-eval after training on only 3K instructions

MODEL	LC WIN RATE	WIN RATE	STD
LANA	**22.50**	**21.35**	1.26
PHI-MINI	20.84	19.98	1.20

Table 2. Unlike other Self-rewarding LM, LANA don't face a significant drop in reasoning tasks.

TASK	LANA	BASE
GSM8K-5SHOT	0.7703	0.7756
HELLASWAG	0.7822	0.7841
ARC-CHALLENGE	0.5674	0.5785
HELLASWAG	0.7822	0.7841

experimentation and lack of resources.

4.1.3. HYPERPARAMETERS

The y, y' were generated using a temperature of 0.1 with maximum length of 128 tokens. We also limited the prompt to a maximum of 256 tokens. These choices were due to resource limitations. The learning rate for SGD optimization was set to 0.0003, and each batch size was 4. Both players parameterized their base models using LoRA (Hu et al., 2021) with a rank of 16.

4.2. Results

Alpaca evaluation (Li et al., 2023; Dubois et al., 2024; 2023) results are shown in Table 1. It shows noticeable improvement after training using LANA on 3K prompts without access to any human-annotated preference dataset.

We also measured the model's performance using LLM-Evaluation Harness (Gao et al., 2023), with results shown in Table 2 and MT-Bench (Zheng et al., 2023), shown in the spider plot in Fig. 2. The most significant observation is that, unlike other self-reward mechanism such as (Yuan et al., 2024), not only is there no noticeable drop in reasoning tasks, but it also seems to perform noticeably better in GSM8K task in MT-bench, matching that of GPT-3.5-turbo, despite being a mini model of 3.8b parameters.

Next we utilized the annotated preference data set in the pre-processed version of the UltraFeedback test dataset (Cui et al., 2023) and different categories in (Huang et al., 2024) to gain a better understanding across different task categories. The results are demonstrated in Fig. 1. They show that LANA helped improve the win rate across all tasks, with some more noticeable improvements in categories such as riddle, theory of mind and plan category, again confirming our assertion regarding the improvements in logical reasoning tasks using LANA without an annotated dataset.

4.3. Ablation study

We tested two additional models, Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) and Gemma-2b-it (Team et al., 2024). Both led to negligible improvements as shown in Fig. 3. While we cannot rule out other reasons, we suspect this implies that the choice of the base model is crucially important. This could mean that the lower quality of training data in these models leads to noisier self-evaluations, which in turn seem to cancel out the progress made during alignment. In contrast, the Phi model training data appears to be collected using the "TextBook all you need" methodology (Gunasekar et al., 2023).

5. Mathematical Discussions

Two points need to be addressed:

- How LANA loss function is derived from Eq. 1?
- Does the game converge, and it what sense?

5.1. LANA loss derivation

An alternative way to define Eq. 1 is through mirror maps (Bubeck et al., 2015). A mirror map is a mapping induced by the convex function ϕ, which maps primal variables to dual variables. Given a convex function ϕ, the mirror map is essentially the gradient ∇_ϕ. For the KL divergence case, the mirror map associated with the negative entropy function is given by the following gradients:

$$\nabla_{\phi(\pi)} = \log(\pi) \tag{3}$$

Then the alternative definition for Eq. 1 is:

$$\pi_{i,t+1} = \arg\min_{\pi_i \in \Delta(\mathcal{Y})} D_\phi(\pi_i, z_{i,t+1}) \tag{4}$$

where $z_{i,t+1}$ is such that:

$$\nabla_{\phi(z_{i,t+1})} = \nabla_{\phi(\pi_{i,t})} + \gamma Q_i^{\tilde{\pi}_{-i,t}} \tag{5}$$

In other words, gradient descent steps are performed in mirror space (policy log-likelihood) instead of in LLM weights space.

Combining the above equations with reward Eq.2 yields:

$$\pi_{i,t+1} = \arg\min_{\pi_i \in \Delta(\mathcal{Y})} \mathbb{E}_{\pi_i}[\log(\tilde{\pi}_{-i,t}/\pi_{i,t})] - H(\pi_i) \tag{6}$$

We instead minimize the upper bound by ignoring the entropy term:

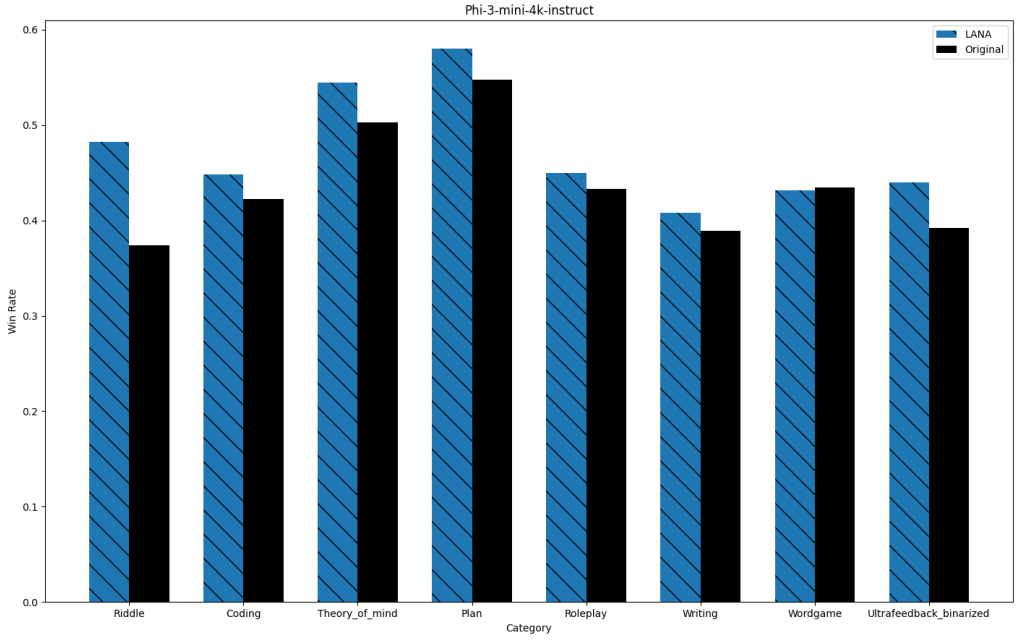

Figure 1. Preference Data set are pre-processed version of the UltraFeedback test dataset (Cui et al., 2023) and different categories in (Huang et al., 2024).

$$loss := \mathbb{E}_{\pi_i} \log(\tilde{\pi}_{-i,t}/\pi_{i,t}) \tag{7}$$

which we pass as the loss to SGD for optimization.

We also note that the choice of Q is essential in deriving such a bound, and it is not simply the result of ignoring the KL term in equation 1. The KL term in Eq. equation 1 $-D_{\mathrm{KL}}(\pi, \pi_{i,t}) = H(\pi) - H(\pi, \pi_{i,t})$ has two goals: first, to encourage exploration for π via the entropy maximization term $H(\pi)$; and second, to avoid reward hacking so that π doesn't deviate too much from the past policy $\pi_{i,t}$ by minimizing the cross-entropy $H(\pi, \pi_{i,t})$. The cross-entropy term is still captured in the objective of Eq. 7. However, our experiments show that entropy term is not important, and therefore for the sake of simplicity is removed from the loss term.

5.2. Convergence

Let's drop the player index notation as the game is symmetric.

Lemma 5.1. *(Munos et al., 2020) Let $p \geq 1$ and $q \geq 1$ such that $1/p + 1/q = 1$. Let ϕ be a strongly convex function with respect to the ℓ_p-norm $\|\cdot\|_p$ with some modulus σ, i.e., for any π, π',*

$$\phi(\pi) \geq \phi(\pi') + \nabla\phi(\pi') \cdot (\pi - \pi') + \frac{\sigma}{2}\|\pi - \pi'\|^2.$$

Write D_ϕ the associated Bregman divergence: for π, π',

$$D_\phi(\pi, \pi') \stackrel{\text{def}}{=} \phi(\pi) - \phi(\pi') - \nabla\phi(\pi') \cdot (\pi - \pi').$$

Let δ be a vector of dimension $|\mathcal{Y}|$. Define π_{t+1} as

$$\pi_{t+1} = \arg\max_{\pi \in \Delta(\mathcal{Y})} \left[\pi.\delta_t - D_\phi(\pi, \pi_t)\right], \tag{8}$$

Then for any $\pi \in \Delta(\mathcal{Y})$, we have,

$$D_\phi(\pi, \pi_{t+1}) \leq D_\phi(\pi, \pi_t) + (\pi_t - \pi).\delta_t + (2/\sigma)\|\delta_t\|_q^2.$$

Using the last lemma with the choice of Q in eq. 2 and D_ϕ being a D_{KL} distance, we have

$\delta_t = \gamma_t \log(\tilde{\pi}_t/\pi_t)$

It follows that

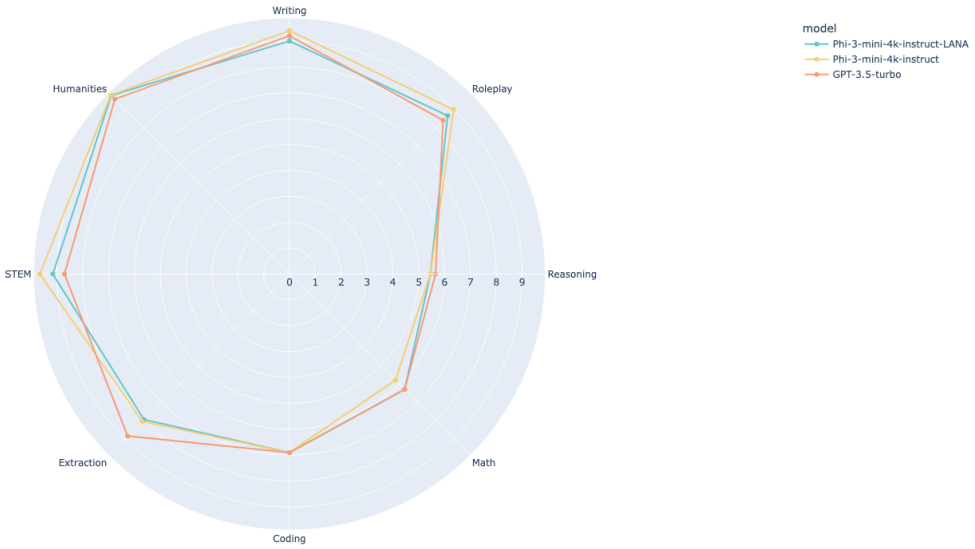

Figure 2. MT-Bench: Unlike other self-rewarding LMs, not only is there no drop in reasoning tasks, but there is also a significant increase in GSM8k, matching that of GPT-3.5-turbo. In other tasks, performance seems to be affected slightly.

$$
\begin{aligned}
&D_{\mathrm{KL}}(\pi, \pi_{t+1}) \leq \\
&D_{\mathrm{KL}}(\pi, \pi_t) + (\pi - \pi_t).\gamma_t \log(\pi_t/\tilde{\pi}_t) \ + (2/\sigma)\|\delta_t\|_q^2 \\
&\leq D_{\mathrm{KL}}(\pi, \pi_t) + (\pi.\gamma_t \log(\pi_t/\tilde{\pi}_t) \ + (2/\sigma)\|\delta_t\|_q^2 \\
&\leq (1-\gamma_t) D_{\mathrm{KL}}(\pi, \pi_t) + \gamma_t D_{\mathrm{KL}}(\pi, \tilde{\pi}_t) + (2/\sigma)\|\delta_t\|_q^2 \quad (9)
\end{aligned}
$$

where the second inequality is the result of $KL(\pi_t, \tilde{\pi}_t) > 0$ and the last inequality is the result of re-writing $\log(\pi_t/\tilde{\pi}_t) = \log(\pi/\tilde{\pi}_t) * \log(\pi_t/\pi)$

By iterating the inequality and assuming the norm is bounded (for example, by ensuring that the policy probability does not have zero support), we can make the following conclusion for a fixed learning rate $\gamma_t = \gamma$:

$$
\begin{aligned}
&D_{\mathrm{KL}}(\pi^*, \pi_T) \leq \\
&D_{\mathrm{KL}}(\pi^*, \tilde{\pi}_c) + (1-\gamma)^T D_{\mathrm{KL}}(\pi^*, \pi_0) + (2/\gamma\sigma)\|\delta_t\|_q^2 \\
&\leq D_{\mathrm{KL}}(\pi^*, \tilde{\pi}_c) + e^{-\gamma T} D_{\mathrm{KL}}(\pi^*, \pi_0) + (2/\gamma\sigma)\|\delta_t\|_q^2 \quad (10)
\end{aligned}
$$

where π^* is Nash equilibrium, $c = \arg\max_{t \in 0,\ldots,T} D_{\mathrm{KL}}(\pi^*, \tilde{\pi}_t)$. From it, we can conclude convergence on average but last iterate convergence is not guaranteed (unless $D_{\mathrm{KL}}(\pi^*, \tilde{\pi}_c) = 0$).

5.3. Future works

An important future direction is to test LANA across different and larger models to assess the role of the base model. Additionally, we didn't have enough compute to train on more than 3k prompts. The question remains how much more improvement LANA alignment would have provided with continued training on more instructions.

Moreover, sampling during training leads to highly inefficient training. Tricks such as PagedAttention (Kwon et al., 2023) are also not applicable during training. Improving the training efficiency of LANA is another important area for future work.

References

Abdin, M., Jacobs, S. A., Awan, A. A., Aneja, J., Awadallah, A., Awadalla, H., Bach, N., Bahree, A., Bakhtiari, A., Behl, H., et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Argilla and Face, H. Dibt 10k prompts ranked, 2024. URL `https://huggingface.co/datasets/DIBT/10k_prompts_ranked`.

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

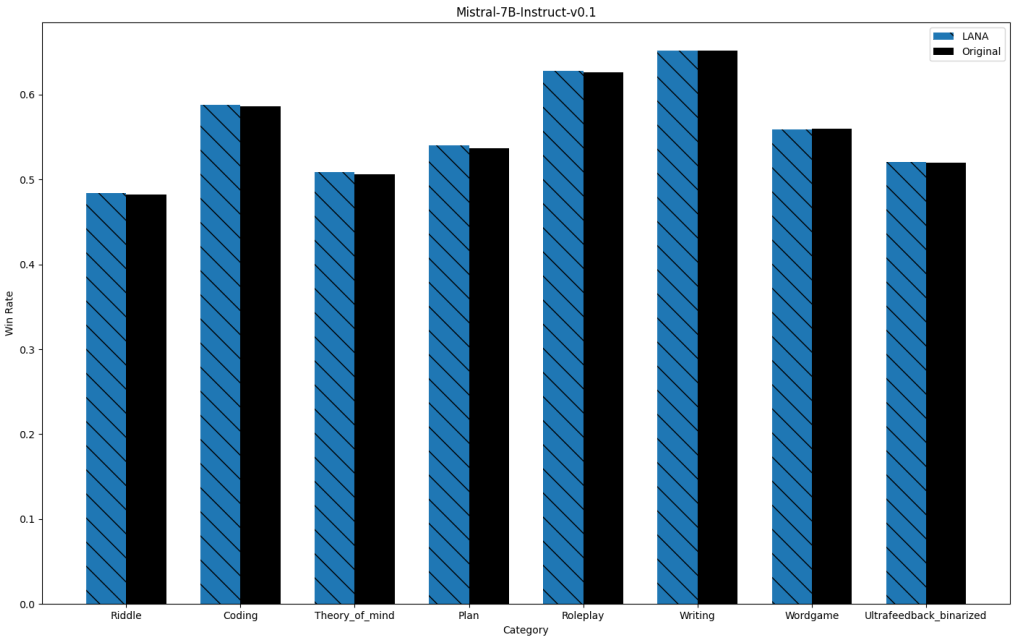

Figure 3. **LANA provides no benefit with Mistral-v1 as foundation model**. Preference datasets are pre-processed version of the UltraFeedback test dataset (Cui et al., 2023) and different categories in (Huang et al., 2024).

Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., and Sun, M. Ultrafeedback: Boosting language models with high-quality feedback, 2023.

Dubois, Y., Li, X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacafarm: A simulation framework for methods that learn from human feedback, 2023.

Dubois, Y., Galambosi, B., Liang, P., and Hashimoto, T. B. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Huang, S. C., Piqueres, A., Rasul, K., Schmid, P., Vila, D., and Tunstall, L. Open hermes preferences. https://huggingface.co/datasets/argilla/OpenHermesPreferences, 2024.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 2023.

Munos, R., Perolat, J., Lespiau, J.-B., Rowland, M., De Vylder, B., Lanctot, M., Timbers, F., Hennes, D., Omidshafiei, S., Gruslys, A., et al. Fast computation of nash equilibria in imperfect information games. In *International Conference on Machine Learning*, pp. 7119–7129. PMLR, 2020.

Munos, R., Valko, M., Calandriello, D., Azar, M. G., Rowland, M., Guo, Z. D., Tang, Y., Geist, M., Mesnard, T., Michi, A., et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Rosset, C., Cheng, C.-A., Mitra, A., Santacroce, M., Awadallah, A., and Xie, T. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Wu, Y., Sun, Z., Yuan, H., Ji, K., Yang, Y., and Gu, Q. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024.

Yuan, W., Pang, R. Y., Cho, K., Sukhbaatar, S., Xu, J., and Weston, J. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.