ARM: Alignment with Residual Energy-Based Model

Bo Pang   **Caiming Xiong**   **Yingbo Zhou**
Salesforce AI Research
b.pang@salesfoce.com

Abstract

While large language models (LLMs) trained with large-scale unsupervised learning acquire a wide variety of world knowledge and skills, its behavior does not necessarily align with human preferences. RLHF methods achieve successes in aligning LLM responses with human preferences and improving the controllability of LLM behavior with human instruction. However, RLHF methods are considerably complicated to implement, computationally expensive to train, and notoriously tricky to tune. In this work, we propose Alignment with Residual Energy-Based Model (ARM), as a simple and flexible alternative to RLHF methods. Our method is driven by an observation that we can learn an aligned policy by minimizing a forward Kullback–Leibler (KL) divergence from a target policy (in the form of a residual energy-based model) to a parameteric policy (LLM), instead of a reverse KL as in RLHF methods. With samples from the energy-based target policy, we can leverage the power of DPO (or other offline methods) to learn an aligned policy efficiently. ARM is simple to implement and applicable in various data settings. Our extensive experiments demonstrate its strong performance across multiple datasets, compared to strong baselines like PPO, DPO.

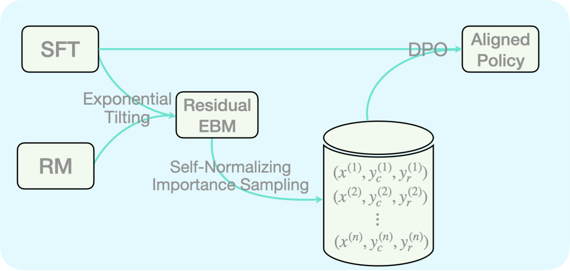
SFT
RM
Exponential Tilting
Residual EBM
Self-Normalizing Importance Sampling
$(x^{(1)}, y_c^{(1)}, y_r^{(1)})$
$(x^{(2)}, y_c^{(2)}, y_r^{(2)})$
$\vdots$
$(x^{(n)}, y_c^{(n)}, y_r^{(n)})$
DPO
Aligned Policy

Figure 1: Graphic illustration of ARM. In ARM, we sample from a target policy, as an exponential tilting of the SFT policy, with self-normalizing importance sampling. We can then learn an aligned policy from these samples with DPO (or any other offline methods).

1 Introduction

Large language models (LLMs) have become extremely powerful and demonstrated remarkable capacities in various domains (OpenAI, 2023; Anil et al., 2023). LLMs trained on very large unsupervised datasets acquire a wide range of capacities and skills, completing tasks zero-shot or few-shot (Radford et al., 2019; Brown et al., 2020). The large unsupervised corpus contains text with various goals and values, which are not necessarily aligned with human preferences.

After unsupervised learning, instruction tuning (Mishra et al., 2021; Sanh et al., 2021; Chung et al., 2022; Wei et al., 2021) is often applied to LLMs, which can significantly improve their capacities on instruction following and align their responses with human values or preferences. While instruct tuning is straightforward, the most successful class of methods for alignment is Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). To apply RLHF, human preferences data on model responses are first collected, and a reward function is learned with the preference data as a surrogate to human value. Given the surrogate reward function, RL methods can be applied, where language models are optimized to produce responses that receive high rewards while not drifting too far away from a reference model (Schulman et al., 2017).

Despite the success of RLHF, these methods are often complicated to implement, expensive to train, and tricky to tune. Recently, there is a surge of interest in developing simpler alternatives to RLHF methods such as DPO (Rafailov et al., 2023), RRHF (Yuan et al., 2023). These methods are straightforward to implement and easier to

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8225–8236
June 16-21, 2024 ©2024 Association for Computational Linguistics

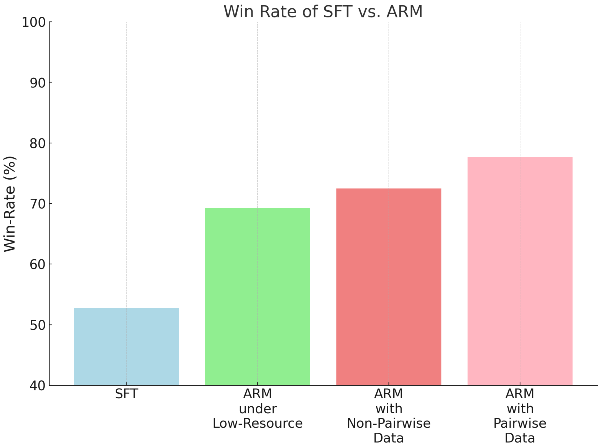

Figure 2: Improvements of ARM over SFT model under various settings. y-axis is the win-rate of LLM responses compared to human responses in the Anthropic-Helpful-Harmless dataset.

train, yet they maintain the performance of RLHF methods on human preference learning.

The reliance on complicated online RL methods is because the reward maximization (with some conservative constraint) in preference learning amounts to minimizing a reverse Kullback–Leibler (KL) divergence $\mathbb{D}_{\mathrm{KL}}(\pi_\theta \mid\mid \pi^*)$, where π^* is the target response distribution or policy that aligns with human preference, and π_θ is parameteric policy (e.g., LLMs) we aim to learn (see Section 4 for more details). Optimizing the reverse KL is not straightforward since sampling from π_θ is not differentiable, and we have to resort to online RL methods to optimize this objective.

In this work, we propose to optimize the forward KL, $\mathbb{D}_{\mathrm{KL}}(\pi^* \mid\mid \pi_\theta)$. As we will show later (Section 4), the target distribution π^* is a residual energy-based model with a reference distribution (usually the SFT distribution) as the base model and the surrogate reward function as the (negative) residual energy term. We can sample from π^* given a learned reference distribution and reward function, and let's denote it as $\mathcal{D}_{\pi^*}$. If we learn π_θ from $\mathcal{D}_{\pi^*}$ with maximum likelihood estimation (MLE), it is a variant of expert iteration (Anthony et al., 2017). Besides MLE, we can learn it with any other offline methods such as DPO. This work focuses on DPO since it is simple and performs well (Rafailov et al., 2023). We call our method of learning policy from $\mathcal{D}_{\pi^*}$ as Alignment with Residual Energy-Based Model (ARM) due to the central role of EBM in our method. See Figure 1 for a illustration of ARM.

We conduct extensive experiments and demonstrate that our method, ARM, yields substantial improvements over SFT policies and outperforms competitive baselines such as PPO and DPO. In addition to standard benchmarks, we also examine ARM when only non-pairwise preference data are available and in low-resource settings. These experiments highlight the applicability of our method to diverse settings due to its simplicity and flexibility. As a preview, Figure 2 displays the win-rate of SFT and ARM policy responses under various conditions, as compared to human preferred responses, on the Anthropic Helpful-Harmless dataset (Bai et al., 2022a).

Our contributions are summarized as follows:

- We propose a new learning method named ARM for aligning LLMs with human preferences.
- ARM is simple to implement and flexible to accommodate various data settings.
- Our experiments show that ARM outperforms strong baselines such as PPO-based RLHF, state-of-the-art RL-free method DPO, in tasks including instruction following, summarization, and dialogue.

2 Related Works

Since unsupervised LLMs have demonstrated unprecedented potentials in a wide variety of tasks and domains (Radford et al., 2019; Brown et al., 2020), much research has dedicated to study how to improve the controllability of LLM behavior, in order to align it with human value and ensure it to follow human instructions.

One line of work focuses on instruction tuning. Early works leverage academic datasets by transforming them into instructional formats with human-written prompt templates (Wei et al., 2021; Sanh et al., 2021; Wang et al., 2022). This approach is scalable and exhibits potentials in making unsupervised LLMs follow instructions. However, its performance significantly lags behind proprietary models like GPT-4 (OpenAI, 2023) which most likely collects a large scale of human-written high quality instruction data. Recently, some researchers attempt to collect high-quality instruction data from strong proprietary models and use them to improve open-sourced models' instruction following capacities (Xu et al., 2023a,b).

Besides instruction tuning, RLHF is another class of methods that demonstrate success in human preference learning (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). This approach learns a surrogate reward function from human preference data and considers LLMs as policy models, and then applies RL methods to maximize rewards assigned to the policy without excessively drifting from some reference model. One popular method, PPO (Schulman et al., 2017), is often used in this setting. Motivated by the complexity of RL methods, many recent works attempt to develop simpler alternatives. DPO (Rafailov et al., 2023) and RRHF (Yuan et al., 2023) are developed along this line. Our work shares the same motivation.

Since collecting human feedback is expensive, some works explore to use AI models to provide feedback, and they offer potential solutions to the scalability limitations of RLHF (Lee et al., 2023; Bai et al., 2022b). Our method, once an SFT model and a reward function are trained with human (or other AI) data, our method characterizes sampling instruction data and preference data from AI models and learning from these samples. Thus, our work is also in line with these RLAIF works on this aspect.

3 Preliminaries

Large language model training starts with unsupervised learning where it is trained on very large datasets with next token prediction. With scaling, LLMs gain wide knowledge and capacities after unsupervised training (Radford et al., 2019; Brown et al., 2020).

To improve LLMs' instruction following capacity, the next step is instruction tuning or supervised finetuning (SFT), where models are finetuned on instructions and human-written completions (Mishra et al., 2021; Sanh et al., 2021; Wei et al., 2021). Given a dataset, $\mathcal{D}_{\text{sft}} = \{(x, y)\}$ where x is an instruction or a prompt and y is a human-written completion, SFT is often done by

$$\pi_{\text{sft}} = \min_{\pi} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{sft}}} \big[-\log \pi(y|x) \big] \qquad (1)$$

To align model behavior with human value, RLHF is applied after learning the SFT policy. This framework assumes 1) a latent reward function, $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, that captures human preference, and 2) preference models, Bradley-Terry model (Bradley and Terry, 1952) or more general Plackett-Luce model (Plackett, 1975; Luce, 2012), that specify preference probability given the reward function. Assume we have access to $\{(x, y_0, y_1)\}$ where $y_0, y_1 \sim \pi_{\text{sft}}(y|x)$, the Bradely-Terry model assumes human preference is captured by the following distribution,

$$p(y_1 \succ y_0|x) = \frac{\exp\left(r(x, y_1)\right)}{\exp\left(r(x, y_1)\right) + \exp\left(r(x, y_0)\right)}. \qquad (2)$$

We define $z \sim \text{Bernoulli}(p(y_1 \succ y_0 \mid x))$, then we can generate a preference dataset, $\mathcal{D}_{\text{pref}} = \{(x, y_0, y_1, z)\}$. Given a parameteric form of reward model, $r_\phi(x, y)$, it can be learned with the negative log-likelihood loss:

$$\begin{aligned}\mathcal{L}(\phi) = -\mathbb{E}_{(x,y_0,y_1,z)\sim\mathcal{D}_{\text{pref}}}\Big[& \\ & z \log \sigma(r_\phi(x, y_1) - r_\phi(x, y_0)) + \\ & \Big(1 - z\Big)\Big(1 - \log \sigma(r_\phi(x, y_1) - r_\phi(x, y_0))\Big)\Big]\end{aligned} \qquad (3)$$

Given $\pi_{\text{sft}}(y|x)$ and $r_\phi(x, y)$, we would like to learn a policy, $\pi_\theta(y \mid x)$, with feedback from the reward model. The objective is often formulated as reward maximization with KL-constraint:

$$\begin{aligned}\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \big[r_\phi(x, y) \big] - & \\ & \beta \mathbb{D}_{\text{KL}} \big[\pi_\theta(y \mid x) \mid\mid \pi_{\text{ref}}(y \mid x) \big],\end{aligned} \qquad (4)$$

where π_{ref} is often set to be π_{sft}. This objective is often optimized with online RL methods such as PPO (Schulman et al., 2017).

As an alternative, Rafailov et al. (2023) proposes direct preference optimization (DPO) where they bypass direct reward modeling via a change of variables to define the preference loss as a function of the policy directly. Therefore, the policy can be trained with the preference loss directly. In particular, the DPO objective is as follows,

$$\begin{aligned}\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[& \\ \log \sigma \Bigg(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - & \\ \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \Bigg) \Bigg],\end{aligned} \qquad (5)$$

where y_w is the preferred response given the prompt x and y_l is the dispreferred response.

4 Methods

4.1 Motivations

As shown in prior works (Peters and Schaal, 2007; Korbak et al., 2022; Go et al., 2023), the KL-constrained reward maximization objective defined in Equation 4 is equivalent to minimizing a reverse KL divergence, $\mathbb{D}_{\text{KL}}\left(\pi_\theta(y \mid x) \mid\mid \pi^*(y \mid x)\right)$ where $\pi_\theta(y \mid x)$ is the parametric policy that we are trying to align with human value and $\pi^*(y \mid x) = \frac{1}{Z(x)}\pi_{\text{sft}}(y \mid x)\exp\left(\frac{1}{\beta}r_\phi(x,y)\right)$.

We may learn π_θ by minimizing $\mathbb{D}_{\text{KL}}\left(\pi_\theta \mid\mid \pi^*\right)$. However, this approach faces two challenges. First, optimizing a reverse KL leads to mode collapsing. Second, it cannot be optimized end-to-end due to the non-differentiability of sampling from π_θ (which has a discrete output space), and this is why researchers resort to RL-based methods such as PPO. While they produce language models with impressive capacities, these methods are considerably complicated to implement, tricky to tune, and computationally expensive to train (e.g., four LLMs need to be fit in GPU memory in PPO training).

4.2 Alignment with Residual Energy-Based Model

In this work, we propose to learn π_θ with the forward KL, $\mathbb{D}_{\text{KL}}\left(\pi^*(y \mid x) \mid\mid \pi_\theta(y \mid x)\right)$. Following this principle, we develop a simple, efficient, flexible, and highly-performant method, and recast several heuristic-driven methods in a probabilistic framework.

4.2.1 Residual Energy-Based Model

Our proposal is driven by the observation that the target distribution π^* is a residual energy-based model (EBM) (Deng et al., 2019; Bakhtin et al., 2021),

$$\pi^*(y \mid x) = \frac{1}{Z(x)}\pi_{\text{sft}}(y \mid x)\exp\left(\frac{1}{\beta}r_\phi(x,y)\right), \tag{6}$$

where $Z(x)$ is a normalizing factor known as partition function, π_{sft} is the SFT-learned distribution, and $\frac{1}{\beta}r_\phi(x,y)$ is the negative energy or the residual in the residual EBM framework (r_ϕ is the learned surrogate reward function, see Equation 3).

4.2.2 Self-Normalizing Importance Sampling

Since all the components in $\pi^*(y \mid x)$ is known, we can directly sample from it. Sampling from EBM, especially discrete EBM, is still under active research (Grathwohl et al., 2021). In this work, we use self-normalizing importance sampling (Shapiro, 2003; Grover et al., 2019). Deng et al. (2019) has shown that it works well with language-model-based EBM. The sampling proceeds in two steps: 1) sampling from the autoregressive language model $\pi_{\text{sft}}(y \mid x)$, 2) resampling according to the negative energy term, $\frac{1}{\beta}r_\phi(x,y)$. This sampling procedure is detailed in Algorithm 1.

Algorithm 1 Self-Normalizing Importance Sampling.

```
Input: $\pi_{\text{sft}}(y \mid x)$, instruction $x$, number of proposal samples to be drawn, $n$.
Output: completion $y$.
1. proposal sampling: Sample $n$ samples $\{y^{(1)}, ..., y^{(n)}\}$ from $\pi_{\text{sft}}(y \mid x)$.
2. resampling with residual energy: Sample $y \sim p(y|x) = \frac{\exp\left(r_\phi(x,y)/\beta\right)}{\sum_{i=1}^{n}\exp\left(r_\phi(x,s_i)/\beta\right)}$.
```

Given this particular choice of sampling method (self-normalizing importance sampling) and the fact the negative energy is defined by a surrogate reward function, sampling from $\pi^*(y \mid x)$ resembles the well-known best-of-n inference (Dubois et al., 2023) where it draws n responses from the SFT model and returns the response with the highest surrogate reward. The difference is that sampling from π^* is a probabilistic approach while best-of-n is greedy.

4.2.3 Expert Iteration

With samples from $\pi^*(y \mid x)$, we can learn θ by minimizing the forward KL, $\mathbb{D}_{\text{KL}}\left(\pi^* \mid\mid \pi_\theta\right)$, which amounts to maximum likelihood estimation (MLE) of θ. That is,

$$\max_\theta \mathbb{E}_{x,y\sim\mathcal{D}_{\pi^*}} \log \pi_\theta(y|x), \tag{7}$$

where $\mathcal{D}_{\pi^*} = \{x, y \mid x \sim \mathcal{D}_{\text{prompt}}, y \sim \pi^*(y \mid x)\}$ and $\mathcal{D}_{\text{prompt}}$ is a collection of prompts. This is a variant of expert iteration considering that responses from π^* can be considered as "expert" responses. This approach is summarized in Algorithm 2.

4.2.4 ARM: Bradley-Terry

In our experiments, expert iteration works well and consistently produces policy that outperforms SFT policy. Considering the advantage of DPO

Algorithm 2 Expert Iteration.

```
Input: $\pi^*(y \mid x)$, prompt dataset $\mathcal{D}_{\text{prompt}} = \{x\}$.
Output: $\pi_\theta(y|x)$.
1. sampling: Sample a completion $y$ from $\pi^*(y|x)$ given $x \sim \mathcal{D}_{\text{prompt}}$ with self-normalizing importance sampling (Algorithm 1), resulting in $\mathcal{D}_{\pi^*} = \{(x, y)\}$.
2. learning: Learn $\theta$ from $\mathcal{D}_{\pi^*}$ via MLE, see Equation 7.
```

Algorithm 3 ARM.

```
Input: $\pi^*(y \mid x)$, prompt dataset $\mathcal{D}_{\text{prompt}} = \{x\}$.
Output: $\pi_\theta(y|x)$.
1. sampling: Sample completions $y_0$ and $y_1$ from $\pi^*(y|x)$ given $x \sim \mathcal{D}_{\text{prompt}}$ with self-normalizing importance sampling (Algorithm 1), resulting in $\mathcal{D}_{\pi^*} = \{(x, y_0, y_1)\}$.
2. scoring: Score $y_0$ and $y_1$, yielding $r_\phi(x, y_0)$ and $r_\phi(x, y_1)$, and then compute the preference probability  (see Equitation 8), giving us $D_{\text{preference}} = \{(x, y_0, y_1, \rho)\}$.
3. learning: Learn $\theta$ from $\mathcal{D}_{\text{preference}}$ via DPO, see Equation 9.
```

over MLE (or the advantage of offline RL methods over behavior cloning in general), the flexibility of our framework allows us to do simple modifications on expert iteration to leverage DPO, which results in alignment with residual energy-based model (ARM). Expert iteration (see Algorithm 2) follows two steps: 1) sampling and 2) MLE learning. In ARM, we 1) add a scoring step as the second step where we collect preference scores using the learned surrogate reward function, $r_\phi(x, y)$, and 2) use DPO instead of MLE to train π_θ.

Given an instruction x, we can sample two responses, y_0 and y_1, from $\pi^*(y|x)$. Bradley-Terry model (Equation 2) is then used to assign preference scores with the surrogate reward model, $r_\phi(x, y)$. In particular, for y_1 being preferred over y_0, $y_1 \succ y_0$, the preference probability, ρ, is

$$\begin{aligned}\rho &= p(y_1 \succ y_0|x) \\ &= \frac{\exp\left(r_\phi(x, y_1)\right)}{\exp\left(r_\phi(x, y_1)\right) + \exp\left(r_\phi(x, y_0)\right)},\end{aligned} \tag{8}$$

and for $y_0 \succ y_1$, we have $1 - \rho$. As such, we build a preference dataset by sampling from $\pi^*(y \mid x)$ and the Bradley-Terry preference model, and let's denote it as $\mathcal{D}_{\text{preference}} = \{(x, y_1, y_2, \rho)\}$. With the preference dataset, π_θ can be learned by minimizing the following objective,

$$\begin{aligned}&\mathcal{L}_{\text{ARM}}(\theta) = -\mathbb{E}_{(x, y_0, y_1, \rho) \sim \mathcal{D}_{\text{preference}}}\Bigg[\\ &\rho \log \sigma \left(\beta \log \frac{\pi_\theta(y_1 \mid x)}{\pi_{\text{sft}}(y_1 \mid x)} - \beta \log \frac{\pi_\theta(y_0 \mid x)}{\pi_{\text{sft}}(y_0 \mid x)}\right) + \\ &(1 - \rho) \log \sigma \Bigg(\beta \log \frac{\pi_\theta(y_0 \mid x)}{\pi_{\text{sft}}(y_0 \mid x)} - \\ &\beta \log \frac{\pi_\theta(y_1 \mid x)}{\pi_{\text{sft}}(y_1 \mid x)}\Bigg)\Bigg].\end{aligned} \tag{9}$$

It is a modified version of the original DPO objective (Rafailov et al., 2023). Notice that is a probability value or soft label. This is accessible because we have the learned surrogate reward model (instead of the latent reward function of human). This learning approach is summarized in Algorithm 3.

The original DPO learning is off-policy since it's trained with samples from a variety of policies (e.g., human, other LLMs). In contrast, our method is more on-policy since it learns from samples of an exponentially-tilted SFT-policy. In addition, ARM directly learns from surrogate reward value instead of chosen-versus-rejected binary feedback as in the original DPO. Given a well-learned reward function, our approach has an advantage.

4.2.5 ARM: Plackett-Luce

The Bradley-Terry model is one choice of reward model. The Plackett-Luce model (Plackett, 1975; Luce, 2012) is a generalization of the Bradley-Terry model when the number of responses is more than two. One practical reason why Bradley-Terry is chosen over Plackett-Luce is because it is more expensive to collect preference data over multiple responses. In our framework, preference data used to train $\pi_\theta(y|x)$ [1] are collected from a learned reward function. Thus, it is trivial to collect preference over multiple responses given a prompt. We next briefly introduce the Plackett-Luce model and show how our method can be extended to the case with Plackett-Luce as the reward model.

As the Bradley-Terry model, the Plackett-Luce model also assumes that human preference is proportional to the value of each choice under some latent reward function, when presented a set of choices. In the LLM context, given a prompt x and

[1] Note that the preference data used to train the surrogate reward function is still collected from human feedback.

	AlpacaFarm	**TL;DR Summarization**	**Anthropic-HH**
SFT	36.7	43.7	52.7
Expert Iteration	41.9	57.2	62.4
PPO	46.8	63.5	63.6
DPO	46.8	65.8	67.3
Best-of-n	45.0	64.6	70.1
Ours	**50.1**	**71.0**	**77.7**

Table 1: Win-rates on AlpacaFarm, TL;DR Summarization, and Anthropic-HH.

	Non-Pairwise Preference Data		
	AlpacaFarm	**TL;DR Summarization**	**Anthropic-HH**
SFT	36.7	43.7	52.7
Expert Iteration	41.7	45.0	57.1
DPO	N/A	N/A	N/A
Best-of-n	43.3	46.2	68.4
Ours	**48.2**	**47.4**	**72.5**

Table 2: Win-rates on AlpacaFarm, TL;DR Summarization, Anthropic-HH when we only have access to non-pairwise preference data.

a set of K LLM responses $\{y_1, \ldots, y_K\}$, human would give a permutation $\tau : [K] \to [K]$, based on their ranking of the responses. The Plackett-Luce model states the distribution of the permutations (rankings) is,

$$p(\tau | y_1, \ldots, y_K, x) = \prod_{k=1}^{K} \frac{\exp(r(x, y_{\tau(k)}))}{\sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}))}, \tag{10}$$

where $y_{\tau(1)}$ is the highest ranked response. Notice that when $K = 2$, Equation 10 reduces to the Bradley-Terry model (Equation 2). Rafailov et al. (2023) shows that DPO can be generalized to the Plackett-Luce model too by parameterizing the reward function $r(x, y)$ as log-ratios of policies. In particular,

$$p_\theta(\tau | y_1, \ldots, y_K, x) = \prod_{k=1}^{K} \frac{\exp\left(\beta \log \frac{\pi_\theta(y_{\tau(k)}|x)}{\pi_{\text{sft}}(y_{\tau(k)}|x)}\right)}{\sum_{j=k}^{K} \exp\left(\beta \log \frac{\pi_\theta(y_{\tau(j)}|x)}{\pi_{\text{sft}}(y_{\tau(j)}|x)}\right)} \tag{11}$$

Similar to ARM based on the Bradley-Terry model (see Section 4.2.4 and Algorithm 3), we can learn the aligned policy $\pi_\theta(y \mid x)$ with the following three steps: 1) sample K model responses, $\{y_1^{(i)}, \ldots, y_K^{(i)}\}$, from $\pi_{\text{sft}}(y \mid x)$, given a prompt $x^{(i)}$; 2) score the K responses with $r_\phi(x, y)$, yielding $\{r_\phi(x, y_1^{(i)}), \ldots, r_\phi(x, y_K^{(i)})\}$; 3) update θ by minimizing a generalized DPO loss,

$$\mathbb{CE}(p_{\text{EBM}}(\tau | y_1, \ldots, y_K, x), p_\theta(\tau | y_1, \ldots, y_K, x)), \tag{12}$$

where $\mathbb{CE}(p, q)$ is the cross-entropy from p to q, and $p_{\text{EBM}}(\tau | y_1, \ldots, y_K, x)$ is the ranking distribution computed following Equation 10 with surrogate reward values from step 2) $(\{r_\phi(x, y_1^{(i)}), \ldots, r_\phi(x, y_K^{(i)})\})$.

5 Experiments

In this section, we empirically evaluate our proposed method, ARM. We first examine its performance on three datasets. After validating its performance in standard settings, we next explore how ARM works in two other interesting settings: 1) we only have access to non-pairwise human feedback 2) we only have access to a limited amount of pairwise human feedback. These experiments aim to demonstrate the flexibility of our method and applicability to realistic scenarios. In the aforementioned experiments, we focus on ARM based on the Bradley-Terry model (see Section 4.2.4). We then compare ARM based on Bradley-Terry versus Plackett-Luce. In the end, we do an ablation on the number proposal samples used in the self-normalizing importance sampling, n (see Algorithm 1).

	TL;DR Summarization		Anthropic-HH	
	2k	8k	2k	8k
SFT	43.7	43.7	52.7	52.7
Expert Iteration	46.3	47.8	56.4	59.7
DPO	47.5	52.6	57.2	60.6
Best-of-n	48.2	51.9	58.1	62.3
Ours	**51.9**	**64.2**	**69.2**	**73.9**

Table 3: Win-rates on TL;DR Summarization and Anthropic-HH when we only have access to a limited amount of preference data (2k and 8k).

5.1 Experiment setup

We conduct experiments on three datasets. Each dataset contains two subsets: 1) an SFT dataset $\mathcal{D}_{\text{sft}} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$; 2) a human preference dataset $\mathcal{D}_{\text{pref}} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^{N}$. We first learn an SFT model with $\mathcal{D}_{\text{sft}}$ and a reward model with $\mathcal{D}_{\text{pref}}$, and then train a policy model with our proposed method, ARM (see Algorithm 3).

We first consider AlpacaFarm (Dubois et al., 2023). It provides a suite of datasets and evaluation methods that enables research and development for learning from feedback. The datasets build upon Alpaca data (Taori et al., 2023) by splitting it into multiple subsets and collecting pairwise feedback data. We use the SFT split (10k) and pairwise preference split (10k) as $\mathcal{D}_{\text{sft}}$ and $\mathcal{D}_{\text{pref}}$ respectively in our experiments. Alpaca data cover diverse topics and models trained on it has shown non-trivial instruction following capacities.

The second dataset is the Reddit TL;DR summarization dataset (Völske et al., 2017). In TL;DR, x is a post from reddit.com with a variety of topics (sbureddits), and y is summary written the original poster (TL;DR). We use the filtered version by Stiennon et al. (2020). It has 123k posts as $\mathcal{D}_{\text{sft}}$. Stiennon et al. (2020) also collected 64k summary comparison on the TL;DR dataset, which we use as $\mathcal{D}_{\text{pref}}$.

Our third dataset is Anthropic's Helpful and Harmless (HH) dataset where each instance consists of a conversation between a human and an AI assistant. In HH, x is a human query (potentially with some conversation history), and y is a response generated by a large (unknown) language model. HH has 170k examples. It does not have a separate $\mathcal{D}_{\text{sft}}$ set, while each instance has a query and two responses (chosen and rejected). We use the collection of query and chosen response as our $\mathcal{D}_{\text{sft}}$.

In AlpacaFarm experiments, we use the pretrained SFT and reward models by Dubois et al. (2023) in order to ensure a fair comparison between our results and reported results. Their models are based on LLama-1-7B-base (Touvron et al., 2023a). In our experiments, our base model is LLama-2-7B-base (Touvron et al., 2023b).

To evaluate our methods, we compute the win-rate of model responses against preferred responses by human. The comparison is done by GPT-4 (gpt-4-0314). Dubois et al. (2023) has demonstrated that the consistency between GPT-4 and humans on model ranks. The prompts we use in evaluation are provided in the Appendix B.

We use the original implementation of DPO by the authors (Rafailov et al., 2023) and the TRLX for PPO training (Castricato et al., 2023). Otherwise, our model training is based on Huggingface transformers (Wolf et al., 2020). For DPO, we use $\beta = 0.1$ in all experiments. For best-of-n and self-normalizing importance sampling, we use $n = 32$ (see Section 5.6 for an ablation). Additional experiment details are given in Appendix A.

5.2 Main results

Our primary results across the three datasets are summarized in Table 1. In comparison to SFT, all methods show sizeable advancements. Simple training method, Expert Iteration, achieves a 14% to 30% improvement over SFT. The standard RLHF method, PPO, and the recently-popularized simpler alternative, DPO, show even greater enhancements beyond Expert Iteration. Notably,the inference-based method, best-of-n , performs surprisingly well, yielding comparable or superior win rates when compared to both PPO and DPO.

Last but not least, our proposed method, ARM, improves the win-rates significantly. In comparison to the previously top-performing methods, PPO, DPO, and best-of-n , our method also exhibits substantial improvements, ranging from 7% to 15%.

5.3 Learn from non-pairwise preference

RLHF methods and recently-developed alternatives assume there exists at least of two responses given an instruction or a prompt. Nevertheless, this is not always the case. In most deployed-chatbot settings, human users interact with an LLM-based assistant and may provide a binary feedback (e.g., thumbs-up versus thumbs-down) to an LLM response. The same instruction or prompt almost never appears twice. Therefore, we may end up with a non-pairwise preference dataset, $\mathcal{D}_{\text{non-pairwise}} = \{(x^{(i)}, y^{(i)}, z^{(i)})\}_{i=1}^{N}$ where $z \in \{0, 1\}$ or $z \in \{\text{like}, \text{dislike}\}$. Methods like DPO is not applicable in the setting without pairwise preference data. Our method, however, is flexible and can be applied.

We only need to make a simple modification by training a surrogate reward function from the non-pairwise data $\mathcal{D}_{\text{non-pairwise}}$. In particular, the reward function can be trained with the following loss function,

$$\mathcal{L}(\phi) = -\mathbb{E}_{(x,y,z)\sim\mathcal{D}_{\text{non-pairwise}}}\Big[z \log \sigma(r_\phi(x, y)) + (1 - z)(1 - \log \sigma(r_\phi(x, y)))\Big] \tag{13}$$

Then a policy can be learned via the same ARM procedure as defined in Algorithm 3.

We conduct experiments based on pairwise datasets (AlpacaFarm, TL;DR summarization, Anthropic-HH). First, we simulate non-pairwise preference data by randomly sampling y from $\{y_w, y_l\}$ and denote $z = 1$ if $y = y_w$ and $z = 0$ if $y = y_l$, resulting in $\mathcal{D}_{\text{non-pairwise}} = \{(x^{(i)}, y^{(i)}, z^{(i)})\}_{i=1}^{N}$. Then we train a reward function according to Equation 13. Following steps proceed exactly the same as the pairwise setting.

The experiment results are displayed in Table 2. First, ARM outperforms Expert Iteration and best-of-n . Second, our method still yields substantial enhancements over the SFT model, especially on AlpacaFarm and Anthropic-HH.

5.4 Learn from limited amount pairwise data

In this section, we investigate our method in low-resource settings. In particular, for TL;DR summarization and Athropic-HH, we sample 2k and 8k pairwise preference data. The results are summarized in Table 3. First, ARM is able to produce significant improvements over SFT policy, and the improvements are larger compared to baselines. Second, although in the 2k setting, the ARM performance is clearly weakened compared to the full dataset performance (see Table 1), it is quite surprising that our method with 8k data (accounting for 10% or less of the full dataset) can recover a large proportion of the performance of the models trained with the full dataset, especially on Anthropic-HH.

5.5 Bradley-Terry versus Plackett-Luce

In this section, we compare the performance of ARM with Bradley-Terry or Plackett-Luce as the preference model. We conduct experiments with both TL;DR summarization and Anthropic-HH. As shown in Table 4, In both datasets, ARM with Plackett-Luce slightly underperforms ARM with Bradley-Terry. This is an intriguing observation. First, intuitively ARM with Plackett-Luce learns from more data and should potentially perform better. Second, a previous theoretical work also shows the advantage of Plackett-Luce (Zhu et al., 2023). Our current hypothesis to the weaker performance of Plackett-Luce is that Plackett-Luce requires more preference accurate labels compared to Bradley-Terry, since it learns from more nuanced comparisons, while the surrogate reward function is noisy. We invite future work to further explore this interesting issue.

5.6 How many proposal samples we need?

As shown in Algorithm 1, to sample from $\pi^*(y \mid x)$, we need first sample n proposal samples form $\pi_{\text{sft}}(x)$. In this experiment, we ablate on the number proposal needed for good performance of our method. As shown in Table 5, as n increases from 16 to 32, we observe a clear improvement on win rate. However, further increasing n yields no improvement.

6 Conclusion

In this work, we propose Alignment with Residual Energy-Based Model (ARM) as a simple alternative to complicated RLHF methods. The core idea is to learn from samples drawn from a target policy in the form of a residual energy-based model, with powerful offline methods like DPO. Our proposed method is characterized by its simplicity and high performance, as demonstrated in diverse tasks and various data settings.

	TL;DR Summarization	Anthropic-HH
Plackett-Luce	69.8	76.3
Bradley-Terry	71.0	77.7

Table 4: Win-rates on TL;DR Summarization and Anthropic-HH with Bradley-Terry versus Plackett-Luce as the human preference model.

n	win-rate
16	47.8
32	50.1
64	49.7
128	50.6

Table 5: Ablation on the number of proposal samples, n, in the self-normalizing importance sampling.

7 Limitations

ARM with Plackett-Luce is a potentially powerful method since it leverages the flexibility of ARM and enables $\pi_\theta(y|x)$ to learn from multiple comparisons. However, the current work did not elucidate the reason that it does not outperform ARM with Bradley-Terry. We find that this question intriguing and hope future work can further investigate it.

Even though the goal of this paper is to align LLMs with human preference, the resulting model still have the risk of producing harmful content. The resulting model should be extensively tested before it is deployed in real-world settings.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Thomas Anthony, Zheng Tian, and David Barber. 2017. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Anton Bakhtin, Yuntian Deng, Sam Gross, Myle Ott, Marc'Aurelio Ranzato, and Arthur Szlam. 2021. Residual energy-based models for text. *Journal of Machine Learning Research*, 22(40):1–41.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Louis Castricato, Alex Havrilla, Shahbuland Matiana, Duy V. Phung, Aman Tiwari, Jonathan Tow, and Maksym Zhuravinsky. 2023. trlX: A scalable framework for RLHF.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 2019. Residual energy-based models for text generation. In *International Conference on Learning Representations*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman.

2023. Aligning language models with preferences through f-divergence minimization. *arXiv preprint arXiv:2302.08215*.

Will Grathwohl, Kevin Swersky, Milad Hashemi, David Duvenaud, and Chris Maddison. 2021. Oops i took a gradient: Scalable sampling for discrete distributions. In *International Conference on Machine Learning*, pages 3831–3841. PMLR.

Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric J Horvitz, and Stefano Ermon. 2019. Bias correction of learned generative models using likelihood-free importance weighting. *Advances in neural information processing systems*, 32.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. 2022. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

R Duncan Luce. 2012. *Individual choice behavior: A theoretical analysis*. Courier Corporation.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. In *Annual Meeting of the Association for Computational Linguistics*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jan Peters and Stefan Schaal. 2007. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pages 745–750.

Robin L Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan D. Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng-Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *ArXiv*, abs/2110.08207.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Alexander Shapiro. 2003. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022.

Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023b. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears.

Banghua Zhu, Jiantao Jiao, and Michael I Jordan. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint arXiv:2301.11270*.

A Additional Experiment Details

In both SFT and reward function training, we first warm up the learning rate to 2e-5 and cosine decay it to 2e-6, and models are trained with 3 epochs. DPO and PPO training follow the default settings in Rafailov et al. (2023) and Castricato et al. (2023). All model training are done with 8 Nvidia A100 GPUs. AlpacaFarm experiments cost 1-2 hours, while TL;DR and Anthropic experiments cost 4-6 hours.

B Evaluation Prompts

We use the same evaluation prompts as in (Rafailov et al., 2023).

GPT-4 win rate prompt for TL;DR summarization.

```
Which of the following summaries does a better job of summarizing the most \
important points in the given forum post, without including unimportant or \
irrelevant details? A good summary is both precise and concise.

Post:
<post>

Summary A:
<Summary A>

Summary B:
<Summary B>

FIRST provide a one-sentence comparison of the two summaries, explaining which \
you prefer and why. SECOND, on a new line, state only "A" or "B" to indicate your \
choice. Your response should use the format:
Comparison: <one-sentence comparison and explanation>
Preferred: <"A" or "B">
```

GPT-4 win rate prompt for Anthroptic-HH.

```
For the following query to a chatbot, which response is more helpful?

Query: <the user query>

Response A:
<either the test method or baseline>

Response B:
<the other response>

FIRST provide a one-sentence comparison of the two responses and explain \
which you feel is more helpful. SECOND, on a new line, state only "A" or \
"B" to indicate which response is more helpful. Your response should use \
the format:
Comparison: <one-sentence comparison and explanation>
More helpful: <"A" or "B">
```