The 38th Canadian Conference on Artificial Intelligence
DOI: 0

Canadian AI 2025
26—29 May 2025, Calgary, Alberta

Aligning Language Models Using Multi-Objective Deep Reinforcement Learning

First Author[†,*], Second Author[†], Third Author[‡]
[†] Affiliation One
[‡] Affiliation Two

Abstract

The alignment techniques used in state-of-the-art language models (LMs), e.g., reinforcement learning from human feedback (RLHF), have driven many successful natural language processing (NLP) tasks. RLHF uses human preferences based on the guidelines of being helpful and safe as a *single* reward signal to fine-tune language models. However, the trade-offs between helpfulness and safety are often found to be a problem, which makes it difficult for a model trained towards one objective to perform well on both. This paper proposes a new alignment technique, multi-objective language model alignment (MOLMA). The framework is based on *multi*-objective deep reinforcement learning to fine-tune language models. MOLMA can efficiently address the conflicting or the dominating learning signal issue caused by the trade-offs of inherent, often conflicting, multi-objectives underlying the language model alignment task. From the overall objective of achieving helpfulness and safety, our results show that MOLMA outperforms the other alignment techniques that rely on single-objective deep reinforcement learning.

Keywords: language model alignment, multi-objective deep reinforcement learning (MODRL), natural language processing (NLP).

1. **Introduction**

Language model alignment is a pivotal and intricate challenge in natural language processing (NLP). Aligning language models with human preferences tremendously improves usability by addressing these model's limitations on the expression of intended behaviors [1]. In this work, we view language model alignment from a novel perspective by treating it as a multi-objective optimization (MOO) task. We focus on developing a new technique using multi-objective deep reinforcement learning to train language models for better alignment.

As one of the most commonly used alignment techniques, reinforcement learning from human feedback (RLHF) [2] dramatically contributes to NLP research [1, 3–8]. RLHF uses single-objective deep reinforcement learning (SODRL) to optimize one objective based on human preferences. However, the most evident drawback of SODRL training is its problems in trade-offs among many NLP tasks with multiple, often conflicting, objectives [9]. Especially for the language model alignment task, the inherent multi-objectives, i.e., helpfulness and safety, are usually in conflict [4]. Single-objective training can adversely impact the learning process, making it hard for a model to perform well on both objectives. To address the conflicting learning signal problem underlying the single-objective training techniques, we introduce a novel multi-objective language model alignment (MOLMA) technique to train language models to optimize helpfulness and safety objectives.

To this end, we start with the phi-2 model [10], which is a small language model (SLM) trained using "textbook-quality" data and is a base model that has not undergone any alignment or fine-tuning yet. Despite having only 2.7 billion parameters, phi-2 can achieve state-of-the-art performance on various academic benchmarks among models with less than 10 billion parameters. The typical protocol to employ RLHF in the training pipeline of language models involves three stages: pre-training (PT), supervised fine-tuning (SFT),

[*] corresponding_author@example.ca

and RLHF. Since phi-2 can already be prompted for question answering (QA) and chat, the PT and SFT stages are omitted in this work. Instead of RLHF, we apply MOLMA to fine-tune the language model. For the MOLMA training, we adopt two reward models to predict scalar scores on helpfulness and safety, respectively. Rewards for helpfulness and safety are treated as equally important learning signals and are independently sent to MOLMA. We aim to eliminate the conflicting or dominating signals during the learning process to optimize both objectives. The key component of MOLMA is a multi-objective deep reinforcement learning (MODRL) algorithm that we devise to fine-tune the model.

This work's major contribution is that we treat language model alignment as an MOO task and develop a novel technique - MOLMA using MODRL to fine-tune language models.

2. Related Work

2.1. Applying RL to Align Language Models

Due to the risk of language models expressing unintended behaviors such as making up facts, generating biased or toxic text, or simply not following user instructions [11, 12], aligning language models with human values, i.e., helpful, truthful, and safe [1, 6, 13] is imperative. RL offers a direct approach to achieving this goal, as the agent requires minimal guidance from a reward model, similar to human proxies, and undergoes numerous iterations within the RL framework to adapt [7]. Due to the straightforward setting of RL, there is a significant amount of research developing alignment techniques using RL-based methods [14]. Besides the noted alignment technique RLHF, Second Thoughts [15] employs RL for text edits to learn alignment. [16] introduces reinforcement learning with synthetic feedback (RLSF), wherein the training data for the reward model are automatically generated, eliminating the need for human-annotated preference data. [17] presents directional stimulus prompting (DSP), a technique employing RL for black-box tuning of language models.

2.2. The Implementation of RL Algorithms for Language Model Alignment

There is a plethora of literature on adopting different RL algorithms to NLP tasks. Some work applies the REINFORCE algorithm for machine translation [18, 19] and text generation [20]. Paulus et al. [21] use the self-critical policy gradient training algorithm for text summarization. Jaques et al. [22] leverage Q-learning for dialog generation. With the advent of Proximal Policy Optimization (PPO) [23], it has been widely employed to improve the performance of language models due to numerous advantages, e.g., ease of implementation, sample efficiency, robustness, and so on [1, 24, 25]. However, in the language environment, PPO encounters challenges such as sparse rewards and ineffective exploration in the word space, rendering it sensitive to hyperparameter settings. For language model training, PPO is found to be unstable and slow in convergence, making it easy to yield ultimate inferior policies. There have been a few attempts to address the problem of instability and sensitivity to hyperparameters. Zheng et al. [7] propose the PPO-max, which assembles the most effective strategy for each component of PPO and is meticulously adjusted to prevent interference among them. Instead of PPO, our work accomplishes the MODRL algorithm for language model alignment based on Advantage-Induced Policy Alignment (APA) [8]. APA leverages squared error to directly regularize the deviation of model policy instead of estimating the importance ratio like PPO, which significantly improves stability and sample efficiency, thus hugely reducing the risk of model collapse. Furthermore, our preliminary experiments using PPO in MOLMA suffered from the reward hacking problem, an interesting task for us to address in the future.

2.3. Multi-objective Optimization Methods

Language model alignment is inherently an MOO task since being helpful and safe is its goal. MOO involves seeking the optimal values for more than one desired objective, requiring simultaneous optimization of multiple objective functions. It is found that, reducing a multi-objective learning problem into a conventional single-objective approach, i.e., weighted sum [26] and piecewise combination [6] of the multiple objectives, makes it hard to solve [27–29]. In addition to the scalarization of the multi-objectives, there is work manually tuning the weights via grid search [30], which is computationally inefficient. Other methods involve optimizing weights using task-specific learning rates or random weighting [31, 32].

Among the various approaches addressing the MOO problem, the most promising outcomes arise from those employing explicit gradient modulation, where a conflicting gradient of one objective is substituted with a modified, non-conflicting gradient. There are many notable gradient modulation methods. PCGrad [33] performs gradient surgery that projects a task's gradient onto the normal plane of the gradient of any other task with a conflicting gradient. GradDrop [34] is a probabilistic masking procedure that samples gradients at an activation layer based on their level of consistency. CAGrad [35] looks for an update vector that maximizes the worst local improvement of any objective in a neighborhood of the average gradient. Nash-MTL [36] views the gradients combination step as a cooperative bargaining game, where tasks negotiate to reach an agreement on a joint direction of parameter update. Lee et al. [37] propose Parrot, a multi-reward RL framework for text-to-image generation where they only update the gradients of non-dominated data points. Among all the gradient modulation methods, AMTL [38] presents state-of-the-art performance on diverse multi-task learning (MTL) benchmarks, including the MTL reinforcement learning benchmarks MT10 in a Meta-World [39] environment. AMTL tries to mitigate the effects of conflicting and dominating gradients by aligning principal components of a gradient matrix. Inspired by AMTL, this work handles the MOO setting in language model alignment by modulating conflicting gradients caused by the underlying conflicting multi-objectives.

3. Multi-Objective Language Model Alignment (MOLMA)

MOLMA incorporates five models, *i.e.*, a reference model (Microsoft phi-2 in this work), a policy model, a value model, and two reward models (helpfulness and safety). The high-level methodology of MOLMA mainly involves reward modeling and MODRL fine-tuning. All models were trained based on phi-2. Our overall workflow is visualized in Figure 1. The following presents the MODRL preliminary, the reward modeling, and MODRL details.

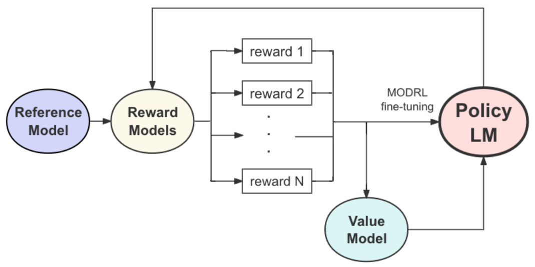


Figure 1. MOLMA training workflow, describing the sequential execution steps. The process includes reward modeling and MODRL fine-tuning.

3.1. Preliminary

The multi-objective language model alignment problem in this work can be formalized as a multi-objective Markov decision process (MOMDP) [9, 40] $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mu, \gamma, \boldsymbol{r} \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T} : (\mathcal{S} \times \mathcal{A}) \times \mathcal{S} \to [0, 1]$ is a probabilistic transition function, $\gamma \in [0, 1)$ is the discount factor, and $\mu : \mathcal{S} \to [0, 1]$ is a probability distribution over initial states. Different from the single-objective MDP, $\boldsymbol{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^K$ is a vector-valued reward function, specifying the immediate reward for each of the considered $K \geqslant 2$ objectives. The language model is the agent operating in the environment with state space $\mathcal{S}$ and action space $\mathcal{A}$. The interaction of the agent and the environment is considered over the sequence of steps: at each time step t, the agent takes action $a_t \in \mathcal{A}$ (actions are a sequence of tokens) in the state $s_t \in \mathcal{S}$ (state is the context) according to its policy π, the environment (reward models) returns an immediate vector-valued reward $\boldsymbol{r}_t$.

Unlike single-objective RL training, in which the objective is to maximize the expected accumulated reward, the learning of multi-objective RL (MORL) can be formulated as:

$$\max_{\phi} \boldsymbol{f}(\phi) = \mathbb{E}_{(s,a)\sim d^{\pi_\phi}}[\boldsymbol{r}(s,a)], \tag{3.1}$$

where π_ϕ is a policy with parameter ϕ, $f_k[\phi] = \mathbb{E}_{(s,a)\sim d^{\pi_\phi}}[r_k(s,a)]$ is the k-th objective, and $d^{\pi_\phi}(s,a) \doteq \sum_{t=1}^{\infty} \gamma^{t-1} p^{\pi}(S_t = s, A_t = a)$ is the discounted state-action occupation measure. The learning task of MORL may be formulated to minimize K losses. There are two families of MORL algorithms in terms of the number of policies: single-policy MORL (most common) and multi-policy MORL (less popular) [40, 41]. A single-policy MORL algorithm aims to learn a single policy using a given preference or importance of the objectives towards a point on the Pareto policy front. A multi-policy MORL algorithm involves learning multiple policies distributed on the Pareto policy front. Note that single-policy MORL with a preference is essentially *different* from single-objective RL with a weighted sum of rewards using the same preference because single-policy MORL requires resolution of gradient conflicts in the path towards the Pareto policy front. In contrast, single-objective RL lacks mechanisms to deal with gradient conflicts and is unlikely to reach the Pareto policy front. Since modern language models are huge in parameter size, single-policy MORL is more suitable for exploring language model alignment in this research.

3.2. **Reward Modeling**

This work intends to optimize the language model alignment task's inherent multiple objectives, i.e., helpfulness and safety. Hence, we trained two separate reward models from the phi-2 model so that reward signals on helpfulness and safety can be independently sent to the later MODRL training.

3.2.1. **Training Objective**

For the training of the two reward models, RMhelp and RMsafe, the language modeling head of the phi-2 model is replaced with a linear layer that generates a solitary output. Following [1, 6], we use a binary ranking loss that enforces the chosen response to obtain a higher score than the rejected response for both RMhelp and RMsafe training:

$$loss(\theta) = -\log\Big(\sigma\big(r_\theta(x, y_c) - r_\theta(x, y_r)\big)\Big), \tag{3.2}$$

where $r_\theta(x, y)$ is the predicted scalar score from the reward model given prompt x and corresponding completion y with respect to parameters θ. y_c is the chosen response and y_r is the rejected counterpart. σ is the sigmoid function.

3.3. **MODRL**

We propose an MODRL algorithm to fine-tune the policy initialized from the phi-2 model. Different from previous work that implements RLHF [1, 3, 5, 6, 26], instead of using PPO, we develop the algorithm based on APA [8] to enhance RL training stability. To improve the policy on all objectives that might inherently conflict, we use the gradient modulation approach to tackle the multi-objective optimization setting.

3.3.1. **Reward**

For the reward, following [3], a per-token Kullback–Leibler (KL) penalty from the original policy at each token is added to reduce the risk of the reward model being overly optimized, thus preventing the fine-tuned policy from moving too far from the original policy. The final adapted reward for MODRL can be uniformly modified as follows:

$$\bar{\boldsymbol{r}}^b\left(y|x\right) = \hat{\boldsymbol{r}}^b\left(y|x\right) - \beta \mathrm{KL}\left(\pi_\phi\left(y|x\right), \pi_0\left(y|x\right)\right), \tag{3.3}$$

where $\bar{\boldsymbol{r}}^b\left(y|x\right)$ is the adapted vector-valued reward in a training batch of size b given prompt x and the completion y. The lengths of $\bar{\boldsymbol{r}}^b\left(y|x\right)$ and $\hat{\boldsymbol{r}}^b\left(y|x\right)$ are equal to the number of objectives. $\pi_\phi\left(y|x\right)$ is the fine-tuned policy; $\pi_0(y|x)$ is the original policy initialized by the phi-2 model. β is the coefficient used to adjust the robustness of KL-penalty. The first term in Equation (3.3) is calculated by processing the raw reward vector $\boldsymbol{r}^b\left(y|x\right)$:

$$\hat{\boldsymbol{r}}^b\left(y|x\right) = \mathrm{WHITEN}\left(\mathrm{LOGIT}\left(\boldsymbol{r}^b\left(y|x\right)\right)\right). \tag{3.4}$$

Following [6], we reparameterizes the original vectored-valued reward $\boldsymbol{r}^b\left(y|x\right)$ by applying the logit function and then whitening within the batch to get $\hat{\boldsymbol{r}}^b\left(y|x\right)$, which helps increase stability and balance properly with the KL penalty term in Equation (3.3).

3.3.2. **APA loss estimation**

Based on the APA algorithm [8], instead of the clipped surrogate used in the PPO, the policy loss of MODRL for the k-th objective is computed as:

$$\widehat{\mathcal{L}}_k^{\mathrm{APA}} = \mathbb{E}_{(s,a)\sim d^{\pi_{\mathrm{old}}}}\left[\left(\log\frac{\pi_\phi\left(a|s\right)}{\pi_0\left(a|s\right)} - \widehat{A}_k^{\pi_{\mathrm{old}}}\left(s,a\right)/\lambda\right)^2\right], \tag{3.5}$$

where π_ϕ is the current policy with parameters ϕ. π_0 is the original policy. The action a (next token) and state s (context) are from the dataset $\mathcal{D} = \{(s_i, a_i) : i = 1, 2, ..., I\}$ sampled from the old policy distribution $d^{\pi_{\mathrm{old}}}$. $\widehat{A}_k^{\pi_{\mathrm{old}}}\left(s,a\right)$ is the old estimated advantage on the k-th objective computed from the reward given in Equation (3.3) based on the generalized advantage estimation (GAE) [23]. λ is a constant (0.95 in our experiments) imposing constraint on the KL regularization term (see [8] for details).

3.3.3. **Value Loss Estimation**

We fit an independent critic network in the MOLMA training process. The MOLMA critic model is trained from the reference model by replacing the language modeling head with a value head. The value function loss for the k-th objective is given as follows:

$$\widehat{\mathcal{L}}_k^V\left(\psi\right) = \mathbb{E}_{(s,a)\sim d^{\pi_{\mathrm{old}}}}\left[\left(V_{k,\psi}\left(a|s\right) - \widehat{A}_k^{\pi_{\mathrm{old}}}\left(s,a\right) - V_{k,\psi_{\mathrm{old}}}\left(a|s\right)\right)^2\right], \tag{3.6}$$

where $V_{k,\psi}\left(a|s\right)$ is the predicted value for objective k from the critic with parameters ψ.

3.3.4. **Final Loss**

The final loss function for learning MOLMA can be given as follows:

$$\mathcal{L}_k\left(\mathcal{D}\right) = \widehat{\mathcal{L}}_k^{\mathrm{APA}}\left(\mathcal{D}\right) + \widehat{\mathcal{L}}_k^V\left(\mathcal{D}\right),\ k = 1, \cdots, K. \tag{3.7}$$

3.3.5. **MODRL Algorithm**

For the MODRL training, we aim to use the gradient modulation method for policy learning. Each loss associated with the objectives (helpfulness and safety) is computed by

Equation (3.7). The gradient modulation method specifically addresses the multi-task optimization challenges, i.e., gradient dominance and gradient conflicts, by aligning principal components of a gradient matrix. The existence of conflicting or dominating gradients disrupts the stability of the training process and leads to a deterioration in overall performance.

Algorithm 1: Multi-Objective Deep Reinforcement Learning (MODRL)

```
Require: $\pi_0$: original policy; $K$: number of objectives; $\boldsymbol{\omega}$: task importance (all objectives are
         deemed equal importance in this work); $\eta$: learning rate;
1  Let $\pi_\phi = \pi_0$;
2  foreach epoch do
3      foreach minibatch do
4          foreach $k = 1, 2, ..., K$ do
5              Compute loss $\mathcal{L}_k(\phi)$;
6              Compute gradient $\boldsymbol{g}_k = \nabla_\phi \mathcal{L}_k(\phi)$;
7          end
8          Get the gradient matrix $\boldsymbol{G} = \{\boldsymbol{g}_1, ..., \boldsymbol{g}_K\}$; // playing objective-specific gradient
               vectors as columns in $\boldsymbol{G}$
9          Compute task space Gram matrix $\boldsymbol{M} \leftarrow \boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$;
10         Get eigen-values and eigen-vectors $(\boldsymbol{\lambda}, \boldsymbol{V}) \leftarrow \mathrm{eigen}(\boldsymbol{M})$; // eigen-decomposition such
               that $\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$ where $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$
11         $\boldsymbol{\Sigma}^{-1} \leftarrow \mathrm{diag}\left(\sqrt{\frac{1}{\lambda_1}}, ..., \sqrt{\frac{1}{\lambda_K}}\right)$;
12         Balance transformation $\boldsymbol{B} \leftarrow \sqrt{\lambda_n}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}^T$;
13         Get new aligned gradient matrix $\widehat{\boldsymbol{G}} = \boldsymbol{G}\boldsymbol{B}$; Updated gradient $\nabla\phi = \widehat{\boldsymbol{G}}\boldsymbol{\omega}$;
14         Update policy parameter $\phi = \phi - \eta\nabla\phi$;
15     end
16 end
17 Return policy $\pi_\phi$;
```

It is acknowledged that gradient dominance can be measured with a gradient magnitude similarity [33], and a cosine distance between vectors can measure the gradient conflicts [35]. However, the two metrics cannot offer a comprehensive assessment if taken in isolation. One of the key components of AMTL is the proposal of the condition number, a stability criterion that can indicate the presence of both challenges. The value of the condition number is the ratio of the maximum and minimum singular values of the corresponding matrix. Minimizing the condition number of the linear system of gradients, a linear combination of gradients for all objectives, mitigates dominance and conflicts within this system. If we apply singular value decomposition (SVD), we can have $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$, where $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_K)$ with the eigen-values arranged in decreasing order. One can easily obtain $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$, where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_K)$ and we know that $\sigma_k = \sqrt{\lambda_k}$. According to AMTL, a gradient matrix with a minimal condition number (i.e., the singular values are equal to the last positive singular value) can be decomposed as: $\widehat{\boldsymbol{G}} = \boldsymbol{U}\widehat{\boldsymbol{\Sigma}}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{U}\sigma\boldsymbol{I}\boldsymbol{V}^{\mathrm{T}} = \sigma\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}} = \sigma\boldsymbol{G}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}^{\mathrm{T}}$, where $\sigma = \sqrt{\lambda_K}$ and $\boldsymbol{U} = \boldsymbol{G}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}$, and $\widehat{\boldsymbol{G}}$ is the aligned gradient matrix. A linear combination of the aligned objective-specific gradient vectors using the objective importance would be $\widehat{\boldsymbol{G}}\boldsymbol{\omega} = \sum_{k=1}^{K} \omega_k \widehat{\boldsymbol{g}}_k$. The pseudocode for the MODRL fine-tuning algorithm proposed in this work to align the language model is given in Algorithm 1.

4. Experiments

4.1. Data

The datasets used to train the two reward models were collected from the open-source preferences datasets: the Stanford SHP dataset [42], the Anthropic Helpful and Harmless dataset [4], and the PKU-SafeRLHF dataset [43]. These datasets comprise pairwise human preference data, a chosen and a rejected response given the same prompt. The dataset used for MODRL training comprises sampled prompts without desired responses from the Cleaned Alpaca dataset [44] and Anthropic Harmless dataset [4]. The proportion of the prompts for helpfulness to the prompts for safety is 60/40, because in our previous experiments we found that providing more safety prompts is conducive to improving the model's safety performance without hurting the helpfulness performance. Table 1 summarizes the data for training the reward models, and training and testing MOLMA.

Table 1. Data used for the MOLMA training and test (data size is counted in terms of number of prompts). See Table 2 for test data used for RMHelp and RMsafe evaluation.

Dataset	Split	Size	Composition	Source	Proportion
RMHelp	train valid	356,811 19,253	chosen&rejected text pairs	PKU-SafeRLHF (help) Anthropic Helpful Stanford SHP	77% 11.5% 11.5%
RMsafe	train valid	322,934 16,997	chosen&rejected text pairs	PKU-SafeRLHF (safe) Anthropic Harmless	87.5% 12.5%
MOLMA	train valid test	34,206 2,159 1,000	sampled prompts only	Cleaned Alpaca Anthropic Harmless	60% 40%

4.2. **Reward Models Evaluation**

To prove the validity of the reward models trained in this work, RMhelp and RMsafe were evaluated for accuracy on various open-source human preference benchmarks. It is reckoned as correct if the reward model assigns a higher score to the preferred response than its counterpart within a text pair.

4.3. **MOLMA Evaluation**

To validate the new alignment technique developed in this work, we compared the performances of the MOLMA against the reference model (phi-2 model) and the other four models trained via SODRL using the same APA method as in MOLMA. The four SODRL models are SOhelp, SOsafe, SOweighted, and SOpiecewise. SOhelp and SOsafe are trained for ablation study. SOhelp is trained to optimize the helpfulness objective alone, and SOsafe is trained to optimize the safety objective alone. The training of SOhelp and SOsafe is the same as the standard training procedure that employs RLHF [1]. SOweighted aims to maximize a weighted sum of the reward for helpfulness and the reward for safety [26]. SOpiecewise uses a piecewise combination of helpfulness and safety rewards, following the training procedure of Llama Chat [6]. The training objectives for each SODRL model are listed below:

SOweighted objective:

$$\arg\max_{\phi} \mathbb{E}_{x\sim\mathcal{D},y\sim\phi}\left[\sum_{k=1}^{K}\frac{1}{K}\bar{r}_k\left(y|x\right)\right], \tag{4.1}$$

where $\bar{r}_k\left(y|x\right)$ is the k-th value of the vector-valued $\bar{\boldsymbol{r}}\left(y|x\right)$ in Equation (3.3) given prompt x and its completion y. The importance weights of helpfulness and safety are equal to make a fair comparison with MOLMA, which is trained evenly toward both objectives.

SOpiecewise objective:

$$\arg\max_{\phi} \mathbb{E}_{x\sim\mathcal{D},y\sim\phi}\left[\bar{r}_p\left(y|x\right)\right] \tag{4.2}$$

$$r_p\left(y|x\right) = \begin{cases} r_{\text{safe}}\left(y|x\right), \; r_{\text{safe}}\left(y|x\right) < \delta \\ r_{\text{help}}\left(y|x\right), \text{ otherwise}\end{cases} \quad , \quad \bar{r}_p(y|x) = \text{WHITEN}\left(\text{LOGIT}\left(r_p^b\left(y|x\right)\right)\right),$$

where $r_{\text{safe}}(y|x)$ is the reward on safety, and $r_{\text{help}}(y|x)$ is the reward on helpfulness. δ is a threshold filtering unsafe responses, which is set according to the accuracy of the RMsafe.

In evaluation, we provide the same prompts to MOLMA, the reference model, and the four SODRL models. RMhelp and RMsafe then respectively assign scalar scores to the responses from these models. Performance is evaluated by comparing scores. The higher the scores, the better the performance in the objective.

All models were trained using FSDP [45] with the AdamW optimizer [46], with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $eps = 10^{-5}$, and gradient clipping of 1.0. For reward models RMhelp and RMsafe, this work used a cosine annealing learning rate schedule down to 10% of the initial learning rate 1×10^{-5}, a weight decay of 0, a batch size of 28, and training for 1 epoch. To make the performances comparable, the training of the four SODRL models used the same hyperparameters as MOLMA training. We used a constant learning rate of 1×10^{-6} and a weight decay of 0.1. This work trained MOLMA and SODRL models for 100 APA iterations with an experience memory size of 64, a KL penalty $\beta = 0.01$, a mini-batch size of 8, and took one gradient step per mini-batch for each iteration. Each training batch for one APA iteration was randomly sampled from the MOLMA training set.

4.4. **Results of Reward Models**

Table 2. **Reward models evaluation accuracy.** The reward models for helpfulness (RMhelp) and safety (RMsafe) were evaluated on human preference benchmarks. Evaluation results of the other open-source reward models on the same data are provided as a reference. The data for evaluation were not included in our training and validation.

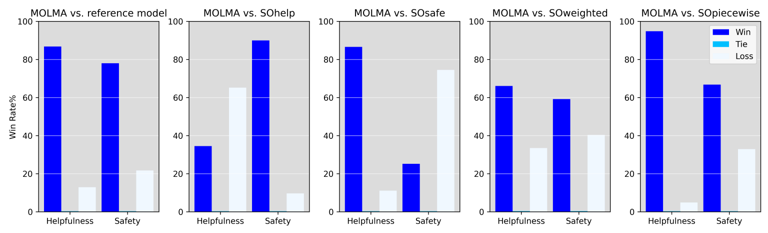
	Stanford SHP	Anthropic Helpful	Anthropic Harmless	PKU-SafeRLHF (helpfulness)	PKU-SafeRLHF (safety)	Avg
SteamSHP-XL	**76.9**	66.8	63.2	63.4	47.2	63.5
Open Assistant	47.6	**71.9**	**69.0**	46.2	57.4	58.4
RMhelp	61.5	60.8	-	**73.4**	-	**65.2**
RMsafe	-	-	60.6	-	**62.7**	61.7

We evaluated RMhelp for helpfulness on the benchmarks - Anthropic Helpful, Stanford SHP, and PKU-SafeRLHF. RMsafe was evaluated for safety on the Anthropic Harmless and PKU-SafeRLHF benchmarks. Each evaluation dataset contains 1,000 randomly collected data. The evaluation accuracies of the reward models are reported in Table 2. We also provide the results of existing open-source reward models - SteamSHP-XL [42] and Open Assistant [5] based on DeBERTa V3 Large V2 [47], on the same data as a reference. RMhelp has the best average performance and highest accuracy on the PKU-SafeRLHF helpfulness benchmark. RMsafe has the highest accuracy on the PKU-SafeRLHF safety benchmark. Thus, the reward models trained in this work are eligible for DRL training.

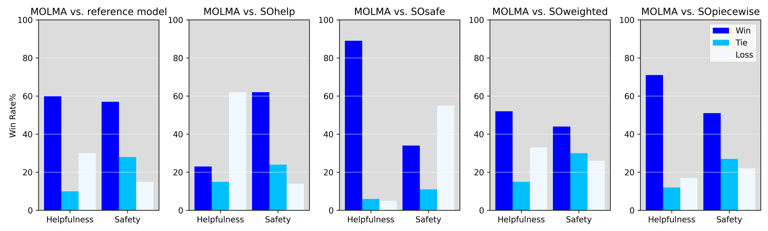
4.5. **Results of MOLMA**

4.5.1. **Model-based Evaluation Results**

After policy learning, MOLMA, was compared with the reference model (phi-2), SOhelp, SOsafe, SOweighted, and SOpiecewise. This work randomly sampled 1,000 prompts from the MOLMA test set for evaluation in terms of helpfulness and safety. As shown in Figure 2a, MOLMA dramatically outperforms the reference model with a win rate in helpfulness reaching 87% and an approximate 78% win rate in safety. MOLMA performs well in both objectives instead of being biased against one. MOLMA outperforms SOweighted in helpfulness and safety, with both win rates around 60%. MOLMA possesses a nearly 95% win rate in helpfulness but only a 67% win rate in safety against SOpiecewise, which the uneven proportion of MOLMA training prompts for helpfulness and safety can cause. The



(a) Model-based evaluation.



(b) Human evaluation.

Figure 2. Model-based evaluation (a) and human evaluation (b) results of MOLMA versus baselines. MOLMA is evaluated on helpfulness and safety by comparing with the reference model, SOhelp, SOsafe, SOweighted, and SOpiecewise.

distributions of rewards for the generated responses to the 1,000 test prompts are shown in Figure 3. We can see that MOLMA's rewards are concentrated at the top right corner. In contrast, the reference model and SOweighted's rewards are inferior to MOLMA's, SOpiecewise's rewards spread in a larger area, and SOhelp and SOsafe's rewards are skewed to emphasize one objective only.

4.5.2. **Human Evaluation Results**

In addition to the model-based evaluations above, five non-relevant student volunteers were recruited at our university to conduct human evaluations for the helpfulness and safety of each model's responses. We randomly sampled 100 prompts from the test dataset. Based on each prompt, each human evaluator was asked to rank the six responses (generated from the six models) from the 1st to the 6th (ties allowed) from the perspective of helpfulness and safety, respectively. All human evaluators were well-informed with the definitions of being helpful and safe for a generated response. Responses were presented randomly to ensure the evaluators were unaware of the corresponding source models. For each prompt, a response wins over another in one objective if it obtains a higher average rank. All human evaluators verbally consented to publish the evaluations. The results in Figure 2b shows that MOLMA outperforms all other models without being biased against one objective.

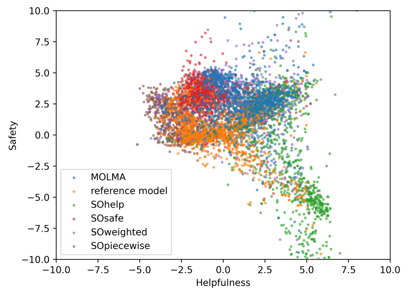


Figure 3. Distributions of rewards of generated responses to test prompts.

4.6. **Evaluation Result of SOweighted with Different Importance Weights**

We explored the performance of SOweighted (weighted sum of rewards to apply SODRL) using a range of importance weights in helpfulness and safety, and compared with our MOLMA which used equal importance weights. The model-based evaluation results are shown in Figure 4. SOweighted struggles with solving the conflicts between helpfulness and safety, while MOLMA shows better balanced performance. Thus, it is convincing that the multi-objective alignment strategy is superior to the weighted-sum scalarization strategy.

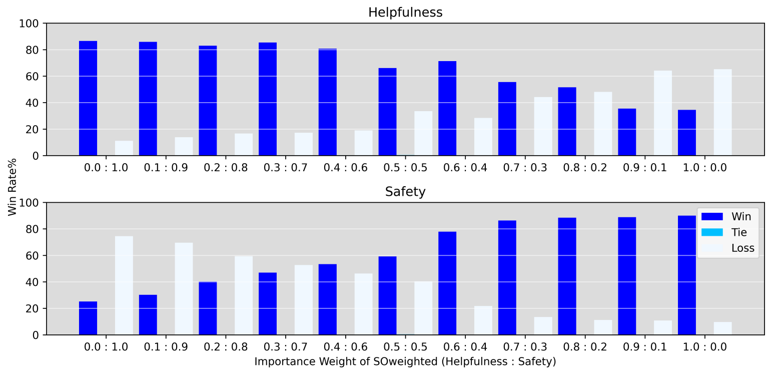


Figure 4. MOLMA (equal weights) versus SOweighted (a range of weights).

4.7. **Evaluation Result of MOLMA Fine-tuned Based on Llama-2 7B**

We experimented with Llama-2 7B [6] and show the model-based evaluation results in Figure 5. MOLMA outperforms the baselines, consistent with the performance based on phi-2. Thus, MOLMA as a general strategy is applicable to improve other language models.

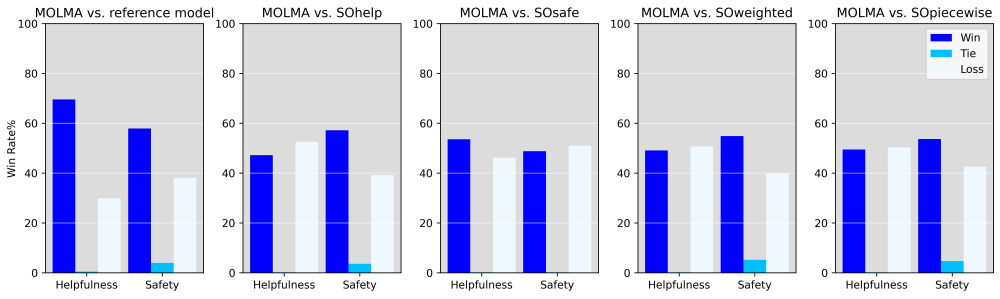


Figure 5. Model-based evaluation result of MOLMA fine-tuned based on Llama-2 7B.

4.8. **Training Time**

All experiments in this paper were executed using 8 NVIDIA A100s. With the base model phi-2 having 2.7 billion parameters, the specific training time for each model, i.e., MOLMA and four SODRL models, is given in Table 3. The time consumed for models using SO-DRL methods in training is roughly the same. MOLMA is relatively disadvantageous in terms of time efficiency.

Table 3. Training time comparison.

Model	**Time (s)**
MOLMA	64204.30
SOhelp	34128.23
SOsafe	34096.60
SOweighted	36767.07
SOpiecewise	37084.33

5. **Conclusion**

We study language model alignment in the multi-objective setup to make the models helpful and safe, which often conflict. Transforming language model alignment into a single-objective optimization task can potentially induce conflicting or dominating learning signals in the learning process, which makes it hard for a model to perform well on both objectives. We develop a multi-objective language model alignment (MOLMA) technique to optimize both objectives simultaneously. By comparing it with other models trained through SO-DRL strategies, we demonstrate the effectiveness of MOLMA. Future work will focus on comparing with other alignment strategies [48], improving MOLMA for adaptive preference alignment [49], and generalizing MOLMA for reasoning as in DeepSeek [50]. Our source code and trained models are available at https://anonymous.4open.science/r/Aligning-Language-Models-Using-Multi-Objective-Deep-Reinforcement-Learning-4FCF.

References

[1] L. Ouyang, J. Wu, et al. "Training language models to follow instructions with human feedback". In: *NeurIPS*. 2022, pp. 27730–27744.

[2] P. F. Christiano, J. Leike, et al. "Deep reinforcement learning from human preferences". In: *NeurIPS*. 2017.

[3] D. M. Ziegler, N. Stiennon, et al. "Fine-tuning language models from human preferences". In: *arXiv preprint arXiv:1909.08593* (2019).

[4] Y. Bai, A. Jones, et al. "Training a helpful and harmless assistant with reinforcement learning from human feedback". In: *arXiv preprint arXiv:2204.05862* (2022).

[5] A. Köpf, Y. Kilcher, et al. "OpenAssistant conversations-democratizing large language model alignment". In: *arXiv preprint arXiv:2304.07327* (2023).

[6] H. Touvron, L. Martin, et al. "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288* (2023).

[7] R. Zheng, S. Dou, et al. "Secrets of RLHF in large language models part I: PPO". In: *arXiv preprint arXiv:2307.04964* (2023).

[8] B. Zhu, H. Sharma, et al. "Fine-tuning language models with advantage-induced policy alignment". In: *arXiv preprint arXiv:2306.02231* (2023).

[9] C. F. Hayes, R. Rădulescu, et al. "A practical guide to multi-objective reinforcement learning and planning". In: *Autonomous Agents and Multi-Agent Systems* 36.1 (2022), p. 26.

[10] S. Gunasekar et al. *Textbooks are all you need*. 2023. URL: https://www.microsoft.com/en-us/research/publication/textbooks-are-all-you-need/.

[11] R. Bommasani, D. A. Hudson, et al. "On the opportunities and risks of foundation models". In: *arXiv preprint arXiv:2108.07258* (2021).

[12] Z. Kenton, T. Everitt, et al. "Alignment of language agents". In: *arXiv preprint arXiv:2103.14659* (2021).

[13] R. Thoppilan, D. De Freitas, et al. "LaMDA: Language models for dialog applications". In: *arXiv preprint arXiv:2201.08239* (2022).

[14] T. Shen, R. Jin, et al. "Large language model alignment: A survey". In: *arXiv preprint arXiv:2309.15025* (2023).

[15] R. Liu, C. Jia, et al. "Second thoughts are best: Learning to re-align with human values from text edits". In: *NeurIPS*. 2022, pp. 181–196.

[16] S. Kim, S. Bae, et al. "Aligning large language models through synthetic feedback". In: *arXiv preprint arXiv:2305.13735* (2023).

[17] Z. Li, B. Peng, et al. "Guiding large language models via directional stimulus prompting". In: *arXiv preprint arXiv:2302.11520* (2023).

[18] M. Ranzato, S. Chopra, et al. "Sequence level training with recurrent neural networks". In: *arXiv preprint arXiv:1511.06732* (2015).

[19] J. Kreutzer, J. Uyheng, and S. Riezler. "Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning". In: *arXiv preprint arXiv:1805.10627* (2018).

[20] P. Tambwekar, M. Dhuliawala, et al. "Controllable neural story generation via reinforcement learning". In: *arXiv preprint arXiv:1809.10736* (2018).

[21] R. Paulus, C. Xiong, and R. Socher. "A deep reinforced model for abstractive summarization". In: *arXiv preprint arXiv:1705.04304* (2017).

[22] N. Jaques, A. Ghandeharioun, et al. "Way off-policy batch deep reinforcement learning of implicit human preferences in dialog". In: *arXiv preprint arXiv:1907.00456* (2019).

[23] J. Schulman, F. Wolski, et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[24] N. Stiennon, L. Ouyang, et al. "Learning to summarize with human feedback". In: *NeurIPS*. 2020, pp. 3008–3021.

[25] R. Nakano, J. Hilton, et al. "WebGPT: Browser-assisted question-answering with human feedback". In: *arXiv preprint arXiv:2112.09332* (2021).

[26] J. Li, W. Monroe, et al. "Deep reinforcement learning for dialogue generation". In: *arXiv preprint arXiv:1606.01541* (2016).

[27] J.-A. Désidéri. "Mutiple-gradient descent algorithm for multiobjective optimization". In: *ECCOMAS*. 2012.

[28] E. Parisotto, J. L. Ba, and R. Salakhutdinov. "Actor-Mimic: Deep multitask and transfer reinforcement learning". In: *arXiv preprint arXiv:1511.06342* (2015).

[29] A. Kendall, Y. Gal, and R. Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics". In: *ICCV*. 2018, pp. 7482–7491.

[30] A. Kendall, M. Grimes, and R. Cipolla. "PoseNet: A convolutional network for real-time 6-dof camera relocalization". In: *ICCV*. 2015, pp. 2938–2946.

[31] Z. Chen, V. Badrinarayanan, et al. "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks". In: *ICML*. 2018, pp. 794–803.

[32] S. Liu, E. Johns, and A. J. Davison. "End-to-end multi-task learning with attention". In: *CVPR*. 2019, pp. 1871–1880.

[33] T. Yu, S. Kumar, et al. "Gradient surgery for multi-task learning". In: *NeurIPS*. 2020, pp. 5824–5836.

[34] Z. Chen, J. Ngiam, et al. "Just pick a sign: Optimizing deep multitask models with gradient sign dropout". In: *NeurIPS*. 2020, pp. 2039–2050.

[35] B. Liu, X. Liu, et al. "Conflict-averse gradient descent for multi-task learning". In: *NeurIPS* 34 (2021), pp. 18878–18890.

[36] A. Navon, A. Shamsian, et al. "Multi-task learning as a bargaining game". In: *arXiv preprint arXiv:2202.01017* (2022).

[37] S. H. Lee, Y. Li, et al. "Parrot: Pareto-optimal multi-reward reinforcement learning framework for text-to-image generation". In: *arXiv preprint arXiv:2401.05675* (2024).

[38] D. Senushkin, N. Patakin, et al. "Independent component alignment for multi-task learning". In: *CVPR*. 2023, pp. 20083–20093.

[39] T. Yu, D. Quillen, et al. "Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning". In: *CoRL*. 2020, pp. 1094–1100.

[40] S. Parisi, M. Pirotta, et al. "Policy gradient approaches for multi-objective sequential decision making". In: *IJCNN*. 2014, pp. 2323–2330.

[41] P. Vamplew, R. Dazeley, et al. "Empirical evaluation methods for multiobjective reinforcement learning algorithms". In: *Machine Learning* 84 (2011), pp. 51–80.

[42] K. Ethayarajh, Y. Choi, and S. Swayamdipta. "Understanding dataset difficulty with $\mathcal{V}$-usable information". In: *ICML*. 2022, pp. 5988–6008.

[43] J. Ji, M. Liu, et al. "BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset". In: *arXiv preprint arXiv:2307.04657* (2023).

[44] R. Taori, I. Gulrajani, et al. *Stanford Alpaca: An instruction-following LLaMA model.* `https://github.com/tatsu-lab/stanford_alpaca`. 2023.

[45] Y. Zhao, A. Gu, et al. "Pytorch FSDP: experiences on scaling fully sharded data parallel". In: *arXiv preprint arXiv:2304.11277* (2023).

[46] I. Loshchilov and F. Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[47] P. He, X. Liu, et al. "DeBERTa: Decoding-enhanced BERT with disentangled attention". In: *arXiv preprint arXiv:2006.03654* (2020).

[48] R. Rafailov, A. Sharma, et al. "Direct preference optimization: Your language model is secretly a reward model". In: *NeurIPS*. 2023.

[49] Y. Zhong, C. Ma, et al. "Panacea: Pareto alignment via preference adaptation for LLMs". In: *NeurIPS*. 2024.

[50] DeepSeek-AI et al. "DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning". In: *arXiv preprint arXiv:2501.12948* (2025).