

Learning to Optimize Multi-Objective Alignment Through Dynamic Reward Weighting

Yining Lu^{◇*} Zilong Wang^{‡†} Shiyang Li[‡] Xin Liu[‡] Changlong Yu[‡]
Qingyu Yin[‡] Zhan Shi[‡] Zixuan Zhang[‡] Meng Jiang[◇]

[◇]University of Notre Dame
{ylu33, mjiang2}@nd.edu

[‡]Amazon
{syangli, xliucr, changlyu, qingyy, zzsamshi, zzx}@amazon.com

Abstract

Prior work in multi-objective reinforcement learning typically uses linear reward scalarization with fixed weights, which provably fails to capture non-convex Pareto fronts and thus yields suboptimal results. This limitation becomes especially critical in online preference alignment for large language models. Here, stochastic trajectories generated by parameterized policies create highly non-linear and non-convex mappings from parameters to objectives that no single static weighting scheme can find optimal trade-offs. We address this limitation by introducing dynamic reward weighting, which adaptively adjusts reward weights during the online reinforcement learning process. Unlike existing approaches that rely on fixed-weight interpolation, our dynamic weighting continuously balances and prioritizes objectives in training, facilitating effective exploration of Pareto fronts in objective space. We introduce two approaches of increasing sophistication and generalizability: hypervolume-guided weight adaptation and gradient-based weight optimization, offering a versatile toolkit for online multi-objective alignment. Our extensive experiments demonstrate their compatibility with commonly used online reinforcement learning algorithms, effectiveness across multiple datasets, and applicability to different model families, consistently achieving Pareto dominant solutions with fewer training steps than fixed-weight linear scalarization baselines.

models (LLMs) for complex reasoning tasks, such as mathematical problem solving (Shao et al., 2024; Zhang and Zuo, 2025), code generation (Chen et al., 2025; Yao et al., 2025), and logical reasoning (Xie et al., 2025; Liu et al., 2025). While this approach has demonstrated significant success in improving model performance, it predominantly focuses on optimizing accuracy while overlooking other essential objectives that are crucial for practical deployment. For instance, key factors such as response length and output clarity, which directly impact inference efficiency and user experience, are not included in the reward-ing process. In this paper, we focus on multi-objective online RL for LLMs that simultaneously optimizes auxiliary objectives alongside traditional accuracy metrics for reasoning tasks.

Existing multi-objective alignment studies rely on either fixed weights (Yao et al., 2025; Team et al., 2025) or heuristic rules for reward interpolation (Zhang and Zuo, 2025; Aggarwal and Welleck, 2025). However, these methods have three critical limitations: (1) Empirically, we find that different objectives vary in learning difficulty.¹ As a result, objectives that reach saturation quickly will plateau throughout the remaining training phases while continuing to receive equal gradient updates, leading to inefficient allocation of learning efforts. (2) Theoretically, static linear scalarization provably fails to cover non-convex regions of the Pareto front and thus yields suboptimal training results (Roijers et al., 2013; Hayes et al., 2022). (3) Heuristic rules for combining objectives lack flexibility and cannot generalize to new objectives or different tasks.

One natural way to address these limitations is to dynamically adjust objective weights throughout training based on the learning progress of each objective. This training paradigm, which we term

1 Introduction

Online reinforcement learning (RL) has become the de facto approach in aligning large language

^{*}Work done during an internship at Amazon.

[†]Work done at Amazon. Now at Google DeepMind.
Email: zlwang@ucsd.edu

¹We provide experimental justification in Appendix A.2.

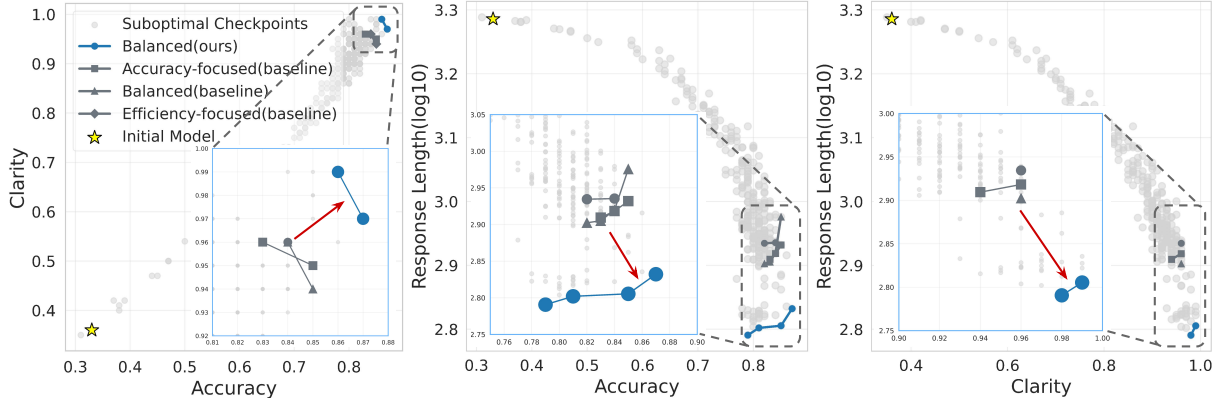


Figure 1: Pareto fronts obtained by our gradient-based weight optimization compared to three baselines using fixed-weight reward interpolation. We train the Qwen3-8B model (Yang et al., 2025) on the Math500 dataset (Lightman et al., 2023) using GRPO (Shao et al., 2024). The three training configurations, *accuracy-focused*, *balanced*, and *efficiency-focused*, correspond to different weight distributions initialized to our optimization objectives: *accuracy*, *conciseness*, and *clarity*. We aim to train models that achieve strong problem-solving ability (higher accuracy) with computational efficiency (fewer tokens) while maintaining interpretable reasoning processes (better clarity). Gray dots indicate Pareto suboptimal checkpoints generated along training. **Clearly, our dynamic reward weighting consistently builds superior Pareto fronts that dominate baselines across all objectives, demonstrating its effectiveness in multi-objective alignment.**

dynamic reward weighting, embodies a core principle that has proven effective in other optimization domains (Liu and Vicente, 2024; Fan et al., 2024): *Redirecting learning effort towards objectives with the greatest potential for improvement*. In this work, we demonstrate that applying this principle to multi-objective LLM alignment could yield substantial improvements over existing static weighting schemes. As shown in Figure 1, our gradient-based method outperforms all three baselines with varying weight configurations.

Specifically, we propose two progressively sophisticated approaches, spanning from strict to flexible training constraints, to suit different multi-objective alignment scenarios. (1) **Hypervolume-guided weight adaptation** (§4): When user preferences for different objectives are given, this method encourages the policy to discover new non-dominated solutions at each training step. It rewards new checkpoints that demonstrate positive hypervolume contributions, thereby proactively pushing the Pareto front in the desired optimization direction. (2) **Gradient-based weight optimization** (§5): When user preferences are not available, the method computes how learning each objective contributes to improving overall model performance through gradient analysis and dynamically reallocates weights accordingly.

Extensive experiments demonstrate that our methods outperform static linear scalarization baselines and a concurrent work PAMA (He and Maghsudi, 2025), achieving superior Pareto fronts with fewer training steps across multiple online RL algorithms (GRPO, REINFORCE, and RLOO), datasets (Math500, MATH, and SafeSQL), and model families (Qwen3, Deepseek, Mistral, and Llama3). These results highlight the importance of dynamic reward weighting for multi-objective RL.² In summary, our contributions are threefold:

- We identify the importance of dynamic reward weighting in multi-objective LLM alignment and formalize this optimization challenge.
- We propose a comprehensive toolkit spanning hypervolume and gradient-based methods to address scenarios with and without human preference priors.
- We demonstrate the effectiveness of our toolkit across multiple online RL algorithms, datasets, and models, showing improved training efficiency and Pareto optimal performance.

²Code to reproduce our results: github.com/yining610/dynamic-reward-weighting.

2 Related Works

Multi-objective RL seeks to balance multiple, sometimes conflicting objectives when optimizing policies (Fleming et al., 2005; Emmerich and Deutz, 2018; Hayes et al., 2022; Huang et al., 2025). This is crucial because optimizing under each criterion can lead to significantly different policies being learned (Rădulescu et al., 2020). We divide the notable prior works on multi-objective RL for LLMs into two groups: methods designed for training steerable multi-objective policies (§2.1) and those focused on general reasoning preference alignment (§2.2).

2.1 Steerable Multi-Objective Preference Finetuning

Most multi-objective RL works in the LLM era explore the problem from a steerability perspective, aiming to train policies that can be steered to generate desirable outputs across a continuum of user-preferred reward weights. Representative approaches include off-policy RL methods (Zhou et al., 2024; Guo et al., 2024; Xiong and Singh, 2025), post-hoc policy mixing (Rame et al., 2023; Wang et al., 2025), and conditional training that incorporates preference weights as input (Yang et al., 2024a,b; Wang et al., 2024; Zhong et al., 2024).

However, these methods typically treat objectives in a static manner, either by using fixed-weight linear scalarization for rewards or by merging trained policies with fixed weights. While they enable inference-time flexibility for users to navigate trade-offs between objectives, they suffer from a fundamental limitation: linear scalarization provably fails to capture non-convex regions of the Pareto front (Hayes et al., 2022; Roijers et al., 2013), resulting in suboptimal policies.

From a geometric perspective, linear scalarization is equivalent to sweeping a hyperplane whose normal vector is the given weights over the objective space. According to the supporting hyperplane theorem (Boyd and Vandenberghe, 2004), for each weight vector, there exists a non-zero supporting hyperplane that is tangent to the convex hull of the Pareto front at some point. Consequently, linear scalarization can only recover Pareto optimal solutions on the convex portions of the Pareto front but fails on those in non-convex (concave) regions (Wei and Niethammer, 2022).

This limitation becomes particularly critical in

online RL settings, where stochastic trajectories create highly non-linear and non-convex mappings from policy parameters to objectives. We therefore replace static weighting with a dynamic reward weighting mechanism that continuously rebalances and reprioritizes objectives during training. This approach facilitates a more thorough exploration of the objective space, enabling the final policy to approximate Pareto optimal solutions in regions that static linear methods cannot reach.

2.2 General Multi-Objective Preference Finetuning

A more straightforward application of multi-objective RL is to train a policy to acquire multiple capabilities in a single training run (Mukherjee et al., 2024), which is the direction we aim to improve. Since the introduction of PPO (Ouyang et al., 2022) and GRPO (Shao et al., 2024), on-policy RL has become the de facto approach for LLM preference alignment. Wu et al. (2023) have applied RLHF to train LLMs that generate more relevant, factual, and informative text. Beyond accuracy, GRPO has been used to train LLMs for high-quality code generation (Yao et al., 2025) and to produce more concise responses in mathematical reasoning tasks (Zhang and Zuo, 2025; Team et al., 2025; Aggarwal and Welleck, 2025). Yet these methods still compute rewards with static linear scalarization or rely on human-defined interpolation rules, limiting their generalizability to new objectives.

To the best of our knowledge, no unified toolkit currently exists for automatically guiding LLM training across arbitrary objective sets. The most relevant line of work is the multi-gradient descent algorithm (MGDA) for multi-objective optimization Désidéri (2012); Zhou et al. (2022); Liu and Vicente (2024); Gu et al. (2024); Xu et al. (2025); Li et al. (2025a). These methods compute gradients for each objective and combine them using various aggregation strategies to guide parameter updates. However, existing approaches primarily focus on relatively simple supervised finetuning tasks, such as logistic regression, and study on classical two-objective Fonseca problems (Fonseca and Fleming, 1996). They are not directly applicable to LLM alignment.

Most closely related to our work, the concurrent study PAMA (He and Maghsudi, 2025) adopts the minimum-norm optimization principle from

MGDA to LLM alignment. While PAMA represents an important step toward multi-objective LLM training, it has only been evaluated on general preference alignment tasks with two objectives (e.g., generating positive sentiment and longer responses on the IMDb dataset (Maas et al., 2011)) using Noon PPO. Its generalizability to more complex reasoning tasks and to different RL algorithms remains unclear.

3 Background

3.1 Notations

We now formulate common notations used throughout the paper. We consider a multi-objective optimization problem with K objectives, where each objective is associated with a weight w_i for $i = 1, 2, \dots, K$. The weight vector is denoted as $\mathbf{w} = (w_1, w_2, \dots, w_K)$, and the corresponding reward vector is $\mathbf{r} = (r_1, r_2, \dots, r_K)$. Following standard practice in RL, we denote the policy parameterized by θ as π_θ , and the objective function as $J(\theta)$. We use superscript w to denote weighted quantities (e.g., G_t^w for weighted discounted return and r^w for weighted reward) and i to denote single-objective quantities (e.g., w_i and $J_i(\theta)$).

3.2 Pareto Front and Hypervolume Contribution

Definition 3.1 (Pareto Front). Let $X \subseteq \mathbb{R}^n$ be the feasible set and $f : X \rightarrow \mathbb{R}^K$, $f(x) = (f_1(x), \dots, f_K(x))$, the vector of objectives to be maximized. Write $u \succ v$ if $u_i \geq v_i$ for every i and $u \neq v$. The Pareto set and Pareto front are formally defined as:

Pareto set $:= P = \{x \in X \mid \nexists y \in X : f(y) \succ f(x)\}$

Pareto front $:= f(P) = \{f(x) \mid x \in P\}$.

Intuitively, the Pareto front consists of all objective vectors for which no objective can be improved without sacrificing at least one other objective. Points in the Pareto set are called *Pareto optimal* or *non-dominated* solutions.

Definition 3.2 (Hypervolume Indicator). Let $\mathbf{A} = \{\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(n)}\} \subset \mathbb{R}^K$ be a finite set of objective vectors, where each $\mathbf{a}^{(i)} = (a_1^{(i)}, \dots, a_K^{(i)})$. Let $\mathbf{r} = (r_1, \dots, r_K) \in \mathbb{R}^K$ be a reference point satisfying $r_j \leq a_j^{(i)}$ for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, K\}$ (i.e., $\mathbf{r}$ is dominated by every

$\mathbf{a}^{(i)}$). The hypervolume of $\mathbf{A}$ with respect to $\mathbf{r}$ is:

$$\text{HV}(\mathbf{A}; \mathbf{r}) = \Lambda\left(\bigcup_{\mathbf{a} \in \mathbf{A}} [\mathbf{a}, \mathbf{r}]\right), \text{ where } [\mathbf{a}, \mathbf{r}] = \{x \in \mathbb{R}^K \mid r_j \leq x_j \leq a_j, \forall j \in \{1, \dots, K\}\}$$

Λ denotes the K -dimensional Lebesgue measure. The hypervolume indicator is the K -dimensional volume of the region dominated by $\mathbf{A}$ and bounded below by $\mathbf{r}$; higher hypervolume indicates larger coverage of the objective space.

Definition 3.3 (Hypervolume Contribution). For any point $\mathbf{a} \in \mathbb{R}^K$, its hypervolume contribution to a set $\mathbf{A}$ is defined as the change in hypervolume when $\mathbf{a}$ is presented and therefore it is always non-negative:

$$\Delta\text{HV}(\mathbf{a}, \mathbf{A}) = \text{HV}(\mathbf{A} \cup \{\mathbf{a}\}) - \text{HV}(\mathbf{A} \setminus \{\mathbf{a}\}) \geq 0.$$

4 Hypervolume-Guided Weight Adaptation

A natural and straightforward way to improve multi-objective alignment is to encourage pushing the Pareto front at each training step. Therefore, we introduce hypervolume-guided weight adaptation, which leverages human-specified weights $\mathbf{w}$ through two hierarchical components: (1) the standard reward vector $\mathbf{r}$, and (2) a meta-level reward $r_{\text{pareto}}(\mathbf{r}, \mathbf{B})$ that encourages policies to achieve new Pareto optimality. Note that we do not directly modify the given human-specified weights $\mathbf{w}$, instead we use the meta-level signal to amplify rewards when training discovers new Pareto fronts. We design $r_{\text{pareto}}(\mathbf{r})$ as:³

$$r_{\text{pareto}}(\mathbf{r}, \mathbf{B}) = 0.5 + 1.5 \tanh(\Delta\text{HV}(\mathbf{r}, \mathbf{B})). \quad (1)$$

$\mathbf{B}$ represents the performance buffer storing validation performance of the current Pareto set. In practice, we compute $\Delta\text{HV}(\mathbf{r}, \mathbf{B})$ using the recursive dimension-sweep algorithm of Fonseca et al. (2006), which achieves computational complexity $\mathcal{O}(n^{k-2} \log n)$ for n points in k dimensions. The complete procedure is detailed in Algorithm 1.

4.1 Experiment Setup

We evaluate our proposed method across multiple online RL algorithms, including natural policy gradient methods REINFORCE (Sutton et al.,

³We experimented with multiple other activation functions with varying range and found that Eq. 1 is the most stable one for our scenario. We provide more details in Appendix A.4.

Method	PAMA		Accuracy-focused		Balanced		Efficiency-focused	
	Baseline		Baseline	Hypervolume-guided	Baseline	Hypervolume-guided	Baseline	Hypervolume-guided
Training RL				<i>Accuracy</i> $\uparrow$ / <i>Response Length</i> $\downarrow$ / <i>Clarity</i> $\uparrow$				
GRPO	0.360 / 1936 / 0.385		0.832 / 701 / 0.962	0.850 / 619 / 0.970	0.832 / 687 / 0.935	0.825 / 683 / 0.955	0.837 / 650 / 0.970	0.840 / 731 / 0.967
REINFORCE	0.380 / 1921 / 0.400		0.789 / 837 / 0.977	0.805 / 827 / 0.988	0.798 / 872 / 0.994	0.788 / 837 / 0.978	0.778 / 676 / 0.985	0.790 / 618 / 1.000
RLOO	0.340 / 1939 / 0.365		0.827 / 677 / 0.965	0.829 / 813 / 0.940	0.820 / 720 / 0.940	0.823 / 639 / 0.955	0.830 / 565 / 0.967	0.843 / 700 / 0.980

Table 1: Pareto fronts obtained from hypervolume-guided weight adaptation compared to PAMA and fixed-weight baselines. $\uparrow$ and $\downarrow$ indicate optimization direction. Each value represents the average performance across the entire Pareto front under a given training setup. **Hypervolume-guided weighting outperforms baselines across most objectives, weight configurations, and RL algorithms**, and in certain cases, demonstrates full dominance with superior results on all three objectives.

Algorithm 1 Hypervolume-Guided Weight Adaptation

```

1: Input: Training set  $\mathcal{D}_{\text{train}}$ , validation set  $\mathcal{D}_{\text{val}}$ , initial policy parameters  $\theta_0$ , human specified weights  $\mathbf{w} \in \mathbb{R}_{\geq 0}^K$ 
2: Hyperparameters: batch size  $B$ , rollout size  $G$ , maximum training steps  $T$ 
3: Evaluate  $\mathbf{r}_{\theta_0} = \text{Evaluate}(\theta_0, \mathcal{D}_{\text{val}})$ 
4: Initialize Pareto set  $\mathbf{B} \leftarrow \{\mathbf{r}_{\theta_0}\}$ 
5:  $r_{\text{pareto}} \leftarrow 1$   $\triangleright$  No hypervolume contribution yet
6: for  $t = 1$  to  $T$  do
7:   Sample mini-batch  $\mathcal{D}_b \subset \mathcal{D}_{\text{train}}$  of size  $B$ 
8:   for all query  $q \in \mathcal{D}_b$  do
9:     Sample  $G$  answers  $\{y_i\}_{i=1}^G \sim \pi_{\theta_{t-1}}(\cdot | q)$ 
10:    Compute reward vectors  $\{\mathbf{r}_i\}_{i=1}^G$  for each  $y_i$ 
11:    Compute scalar rewards  $r_i \leftarrow \mathbf{w}^\top \mathbf{r}_i$ 
12:    Adaptation:  $\tilde{r}_i \leftarrow r_{\text{pareto}} \cdot r_i$ 
13:   end for
14:   Update  $\theta_{t-1}$  via chosen RL objective function  $J(\theta_{t-1})$  using  $\tilde{r}_i$ 
15:    $\mathbf{r}_{\theta_t} \leftarrow \text{Evaluate}(\theta_t, \mathcal{D}_{\text{val}})$ 
16:    $r_{\text{pareto}} \leftarrow 0.5 + 1.5 \tanh(\Delta\text{HV}(\mathbf{r}_{\theta_t}, \mathbf{B}))$   $\triangleright$  Update meta weight based on hypervolume contribution
17:   if  $\Delta\text{HV}(\mathbf{r}_{\theta_t}; \mathbf{B}) > 0$  then
18:      $\mathbf{B} \leftarrow \mathbf{B} \cup \{\mathbf{r}_{\theta_t}\}$   $\triangleright$  Update Pareto set
19:   end if
20: end for

```

1999) and RLOO (Ahmadian et al., 2024), and heuristic policy gradient method GRPO (Shao et al., 2024). The main experiments are conducted on the Math500 dataset (Lightman et al., 2023) using the Qwen3-8B model (Yang et al., 2025). To provide a more comprehensive evaluation of our method, we then extend our experiments to additional datasets spanning mathematical and coding reasoning tasks (Math algebra problems (Hendrycks et al., 2021) and SafeSQL (Yao et al., 2025)), as well as additional model families, including Deepseek-7B (DeepSeek-AI et al., 2024), Mistral-7B (Jiang et al., 2023), and Llama3-8B

(Grattafiori et al., 2024), in §6. All results are reported from the test set.

For math tasks, we design the following three objectives: (1) generating correct solutions (*accuracy*), (2) producing efficient solutions that minimize computational steps and tokens (*conciseness*), and (3) demonstrating clear reasoning with step-by-step thinking process (*clarity*). We denote the weight vector as $\mathbf{w} = [w_{\text{accuracy}}, w_{\text{conciseness}}, w_{\text{clarity}}]$. Each objective is evaluated using heuristic rules to ensure the corresponding reward is verifiable (0 or 1). For easier analysis, in the following tables and figures, we present accuracy and clarity performance based on their reward scores, while conciseness is reported directly through response length.

Following common practice in multi-objective RL (Rame et al., 2023; Li et al., 2025b), we use fixed-weight linear scalarization as one of our baselines. Specifically, we evaluate three carefully chosen weight configurations that represent different optimization priorities: 1) *Accuracy-focused* ($\mathbf{w} = [0.5, 0.25, 0.25]$) prioritizing accuracy over conciseness and formatness; 2) *Balanced* ($\mathbf{w} = [0.334, 0.333, 0.333]$) weighting all objectives equally; and 3) *Efficiency-focused* ($\mathbf{w} = [0.25, 0.375, 0.375]$) emphasizing conciseness and clarity. Additionally, we compare our method against the concurrent work PAMA (He and Maghsudi, 2025).⁴ More experiment setup details can be found in Appendix A.3.

4.2 Results

We compare our hypervolume-guided weight adaptation against fixed-weight baselines and PAMA in Table 1. Across all three online RL algorithms, there is consistently at least one weight configuration where our method outperforms the

⁴As their code is not publicly available, we reimplement PAMA following the methodology and hyperparameters reported in the paper.

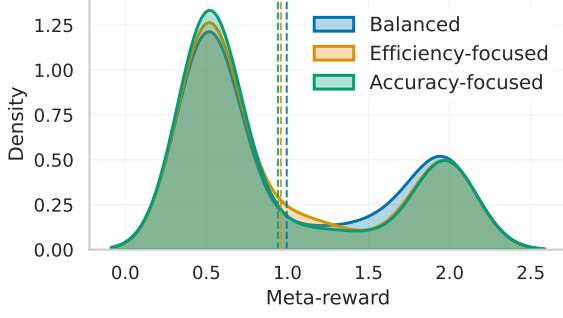


Figure 2: Meta-reward r_{pareto} distributions, with vertical dashed lines indicating the average value.

baselines on all objectives. For example, under both accuracy- and efficiency-focused settings, REINFORCE equipped with hypervolume-guided weighting achieves Pareto fronts that have better average scores across all three objectives.

It is important to note that better average scores of Pareto fronts do not necessarily guarantee true Pareto optimality. Due to space constraints, we include the full Pareto front visualizations for Table 1 in Appendix A.5 (Figure 9). These plots provide compelling visual evidence that REINFORCE with our hypervolume-guided weight adaptation achieves clearly superior Pareto fronts, dominating baselines under both accuracy- and efficiency-focused settings.

To better understand the performance differences between weight configurations, we analyze the meta-reward r_{pareto} during REINFORCE training. As shown in Figure 2, the balanced weight setting yields higher meta-rewards that encourage more aggressive exploration, paradoxically, leading to suboptimal performance compared to two other configurations. This suggests that effective hypervolume-guided weighting requires careful calibration of meta-reward to avoid overly aggressive updates that can hinder convergence.

5 Gradient-Based Weight Optimization

We now consider a more sophisticated yet flexible setting where reward weights are not fixed but dynamically adapted in real-time. In this scenario, human-specified preferences are not assumed. We formulate the online weights adaptation during training as an optimization problem and begin our study with the natural policy-gradient method REINFORCE:

$$\nabla J(\theta) = \mathbb{E}[\nabla \log \pi_{\theta}(a_t | s_t) G_t^w], \quad (2)$$

which exhibits linearity to the per-objective gradients given by $\nabla J(\theta) = \sum_{i=1}^K w_i \nabla J_i(\theta)$. We prove this linearity through a simple nested summation

$$\begin{aligned} G_t^w &= \sum_{l=0}^{\infty} \gamma^l r_{t+l+1}^w = \sum_{l=0}^{\infty} \gamma^l \sum_{i=1}^K w_i r_{t+l+1}^i \\ &= \sum_{i=1}^K w_i \sum_{l=0}^{\infty} \gamma^l r_{t+l+1}^i = \sum_{i=1}^K w_i G_t^i. \end{aligned}$$

Then there exists a constant $C \in \mathbb{R}$ such that $J(\theta) = \sum_{i=1}^K w_i J_i(\theta) + C$, which implies

$$\arg \min_{\mathbf{w} \in \Delta^K} J(\theta(\mathbf{w})) = \arg \min_{\mathbf{w} \in \Delta^K} \sum_{i=1}^K w_i J_i(\theta).$$

Inspired by domain adaptation (Fan et al., 2024) and based on the above linearity assumption, we can derive the following reward weight update rule, where η is the learning rate and μ is the regularization factor (we provide full proof in Appendix A.1 and the whole procedure in Algorithm 2):

$$\begin{aligned} \mathbf{w}^{(t)} &= \frac{\mathbf{w}^{(t-1)}}{\sum_k w_k^{(t-1)}}, \quad \mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} \odot \exp\left(\frac{\eta^{(t)} \mathbf{I}^{(t)}}{\mu}\right), \\ I_i^{(t)} &= \langle \nabla J_i(\theta^{(t)}), \sum_{k \in [K]} \nabla J_k(\theta^{(t)}) \rangle. \end{aligned} \quad (3)$$

Remark. The $I_i^{(t)}$ measures the total influence that learning objective i has on the remaining $K - 1$ objectives ($\langle \nabla J_i, \sum_{k \neq i} \nabla J_k \rangle$) plus the magnitude of its own gradient ($\|\nabla J_i\|_2^2$). Intuitively, similar to prior studies in data influence (Koh and Liang, 2017) and domain adaptation (Fan et al., 2024), we upweight an objective when it has high learning potential on other objectives (high influence) and has not been learned enough yet (high gradient magnitude).

5.1 Convergence Analysis

A natural concern in weight updating is whether the weights may collapse to zero or explode to the boundary value of one within a single update step. To address this concern, we provide a convergence analysis for Eq. 3. We first make some standard assumptions used across existing RL (Wang et al., 2017; Wu et al., 2020; Kumar et al., 2023) and multi-objective optimization studies (Zhou et al., 2022; Liu and Vicente, 2024; Xu et al., 2025).

Algorithm 2 Online Reward Weights Optimization

```

1: Input: Training set  $\mathcal{D}_{\text{train}}$ , initial policy parameters  $\theta_0$ , initial objective weights  $w^{(0)} \in \mathbb{R}_{\geq 0}^K$ 
2: Hyperparameters: batch size  $B$ , rollout size  $G$ , learning rate  $\eta$ , regularization factor  $\mu$ , maximum training steps  $T$ 
3: for  $t=1$  to  $T$  do
4:   Sample mini-batch  $\mathcal{D}_b \subset \mathcal{D}_{\text{train}}$  of size  $B$ 
5:   for all query  $q \in \mathcal{D}_b$  do
6:     Sample  $G$  answers  $\{y_i\}_{i=1}^G \sim \pi_{\theta_{t-1}}(\cdot | q)$ 
7:     Compute reward vectors  $\{r_i\}_{i=1}^G$  for each  $y_i$ 
8:   end for
9:   Obtain gradients  $\{\nabla J_i(\theta_{t-1}, \mathcal{D}_b)\}_{i=1}^K$ 
10:  Compute influence signal  $I^{(t)}$  ▷ See Eq. 3
11:  Update weights:
12:     $w^{(t)} \leftarrow w^{(t-1)} \odot \exp\left(\frac{\eta^{(t)} I^{(t)}}{\mu}\right)$ 
13:     $w^{(t)} \leftarrow w^{(t)} / \sum_i w_i^{(t)}$ 
14:    Compute scalar rewards  $\{w^{(t)\top} r_i\}_{i=1}^G$ 
15:    Update  $\theta_{t-1}$  via chosen RL objective function  $J(\theta_{t-1})$  using scalarized rewards
16: end for

```

Assumption 5.1 (Lipschitz Continuity). *The policy π_θ is differentiable with respect to θ and the output $\log \pi_\theta(a | s)$ is L -Lipschitz and has bounded norm $\|\nabla \log \pi_\theta(a | s)\| \leq B$ for any θ .*

Assumption 5.2 (Bounded Reward). *The reward is bounded in the range $[0, r_{\max}]$.*

Assumption 5.3 (Learning Rate). *The learning rate $\eta^{(t)}$ is non-increasing and non-negative.*

Assumption 5.4 (Bounded Policy Gradient). *The true gradient (Eq. 2) of the objective function is bounded $\|\nabla J_k(\theta)\| \leq C \leq \infty$ which is established by $\|\nabla \log \pi_\theta(a | s)\| < B$ (Assumption 5.1) and $G_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l+1} \leq \sum_{l=0}^{\infty} \gamma^l r_{\max} = \frac{r_{\max}}{1-\gamma}$ (Assumption 5.2).*

Lemma 5.1. *Recall the weight update rule defined in Eq. 3, its true weight takes the closed form:*

$$w_i^{(T)} = \frac{w_i^{(0)} \exp\left(\sum_{t=1}^T \tau^{(t)} I_i^{(t)}\right)}{\sum_{k \in [K]} w_k^{(0)} \exp\left(\sum_{t=1}^T \tau^{(t)} I_k^{(t)}\right)},$$

where $\tau^{(t)} = \frac{\eta^{(t)}}{\mu}$

Proof. The lemma holds when $T = 1$:

$$w_i^{(1)} = \frac{w_i^{(0)} \exp(\tau^{(1)} I_i^{(1)})}{\sum_{k \in [K]} w_k^{(0)} \exp(\tau^{(1)} I_k^{(1)})}.$$

Suppose the lemma holds for all values up to $T-1$, for step at T , we have:

$$\begin{aligned} w_i^{(T)} &= \frac{w_i^{(T-1)} \exp(\tau^{(T)} I_i^{(T)})}{\sum_k w_k^{(T-1)} \exp(\tau^{(T)} I_k^{(T)})} \\ &= \frac{\frac{w_i^{(0)} \exp\left(\sum_{t=1}^{T-1} \tau^{(t)} I_i^{(t)}\right)}{\sum_{k \in [K]} w_k^{(0)} \exp\left(\sum_{t=1}^{T-1} \tau^{(t)} I_k^{(t)}\right)} \exp(\tau^{(T)} I_i^{(T)})}{\sum_k \frac{w_k^{(0)} \exp\left(\sum_{t=1}^{T-1} \tau^{(t)} I_k^{(t)}\right)}{\sum_{k \in [K]} w_k^{(0)} \exp\left(\sum_{t=1}^{T-1} \tau^{(t)} I_k^{(t)}\right)} \exp(\tau^{(T)} I_k^{(T)})} \\ &= \frac{w_i^{(0)} \exp\left(\sum_{t=1}^{T-1} \tau^{(t)} I_i^{(t)}\right) \exp(\tau^{(T)} I_i^{(T)})}{\sum_k w_k^{(0)} \exp\left(\sum_{t=1}^{T-1} \tau^{(t)} I_k^{(t)}\right) \exp(\tau^{(T)} I_k^{(T)})} \\ &= \frac{w_i^{(0)} \exp\left(\sum_{t=1}^T \tau^{(t)} I_i^{(t)}\right)}{\sum_k w_k^{(0)} \exp\left(\sum_{t=1}^T \tau^{(t)} I_k^{(t)}\right)} \end{aligned}$$

□

Theorem 5.2. *Suppose Assumptions 5.1-5.4 hold and that the step-size sequence of τ converges $|\sum_{t=1}^T \tau^{(t)} - \ell| < \epsilon$ for every arbitrarily small positive number ϵ . Then, for every pair of objectives $i, j \in [K]$, their weight ratio is uniformly bounded: $w_i^{(T)}/w_j^{(T)} = \mathcal{O}(1)$, $\forall T \in \mathbb{N}$.*

Proof.

$$\begin{aligned} \frac{w_i^{(T)}}{w_j^{(T)}} &= \frac{w_i^{(0)} \exp\left(\sum_{t=1}^T \tau^{(t)} I_i^{(t)}\right)}{w_j^{(0)} \exp\left(\sum_{t=1}^T \tau^{(t)} I_j^{(t)}\right)} \\ &= \frac{w_i^{(0)}}{w_j^{(0)}} \exp\left(\sum_{t=1}^T \tau^{(t)} (I_i^{(t)} - I_j^{(t)})\right) \\ &\leq \frac{w_i^{(0)}}{w_j^{(0)}} \exp\left(\sum_{t=1}^T \tau^{(t)} 2KC^2\right) \\ &\leq \frac{w_i^{(0)}}{w_j^{(0)}} \exp(2KC^2\ell) = \mathcal{O}(1) \end{aligned}$$

□

Remark. The ratio $w_i^{(T)}/w_j^{(T)}$ measures the cumulative advantage that objective i has gained over objective j throughout training, determined by both their initial weights $w^{(0)}$ and the accumulated influence across iterations $I^{(t)}$. Intuitively, an objective that is more influential on the gradient update will naturally gain a higher weight relative to a less influential objective. Keeping this ratio uniformly bounded has two merits. First, it offers numerical stability. A bounded ratio prevents weight collapse or explosion from the exponential update, ensuring that the updated weights w

Method →	PAMA	Accuracy-focused	Balanced	Efficiency-focused	Gradient-based(Ours)
Training RL ↓	Accuracy ↑ / Response Length ↓ / Clarity ↑				
GRPO	0.360 / 1934 / 0.387	0.836 / 831 / 0.948	0.830 / 861 / 0.960	0.835 / 842 / 0.950	0.836 / 650 / 0.980
REINFORCE	0.341 / 1929 / 0.369	0.764 / 1375 / 0.830	0.760 / 1336 / 0.850	0.755 / 1361 / 0.825	0.802 / 1202 / 0.868
RLOO	0.355 / 1938 / 0.385	0.815 / 1100 / 0.905	0.820 / 847 / 0.937	0.830 / 824 / 0.940	0.820 / 701 / 0.980

Table 2: Pareto fronts obtained from gradient-based weight optimization compared to PAMA and fixed-weight baselines. $\uparrow$ and $\downarrow$ indicate optimization direction. Each value represents the average performance across the entire Pareto front under a given training setup. **Clearly, the gradient-based weighting approach yields superior Pareto fronts compared to all baselines across both GRPO and REINFORCE training setups**, and achieves improved performance on two key objectives (conciseness and clarity) under RLOO training.

remain properly constrained and ultimately converge to a valid multi-objective equilibrium. Second, it has practical guidance for hyperparameter selection. Initializing w uniformly and employing a learning rate scheduler in which $\eta^{(t)}$ converges, so that each step updates the weights within safe and predictable bounds.

5.2 Experiment Setup

As suggested by Theorem 5.2, we initialize the reward weights using a balanced (uniform) setting and use a polynomial learning rate scheduler whose learning rate sequence converges, instead of the constant scheduler used in §4. To maintain the linearity between the overall reward and the per-objective rewards, we do not add the token-level KL divergence penalty from the reference model in each reward. Furthermore, standard approaches for preference finetuning LLMs with REINFORCE, such as REINFORCE++ (Hu et al., 2025), typically employ clipping mechanisms that render the objective non-differentiable and break the linearity assumptions made in Eq. 2. Therefore, in our experiments, we set both the clip range ϵ (Schulman et al., 2017) and dual clip constant c (Ye et al., 2020) high ($\epsilon = c = 100$) to practically disable clipping.⁵ To improve computational efficiency, we compute gradients only from middle layers, as these layers typically encode the most abstract and generalizable knowledge patterns (Grosse et al., 2023). All other experiment settings follow §4.1.

5.3 Results

We compare our gradient-based weight optimization approach against baselines trained with differ-

⁵We also experimented with standard clipping settings ($\epsilon = 0.2, c = 3$) and observed that disabling clipping did lead to improved training results.

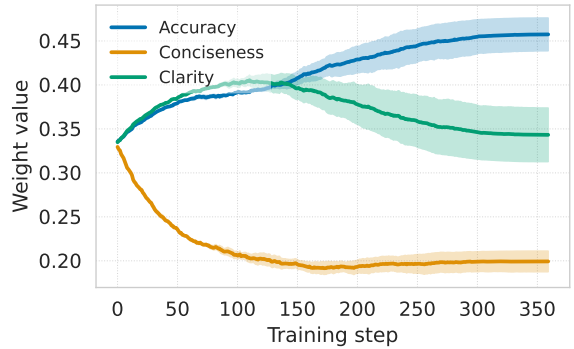
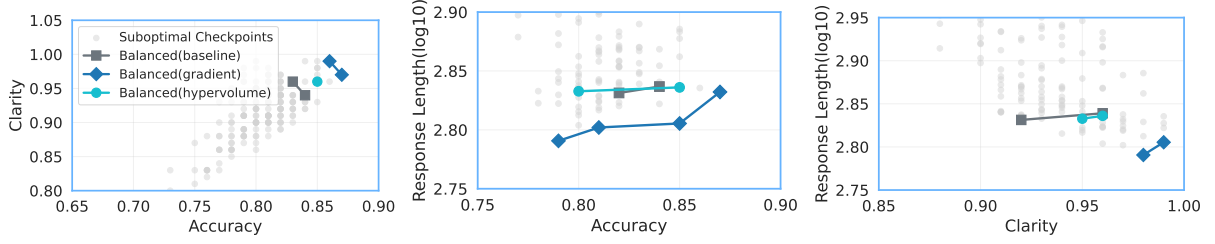


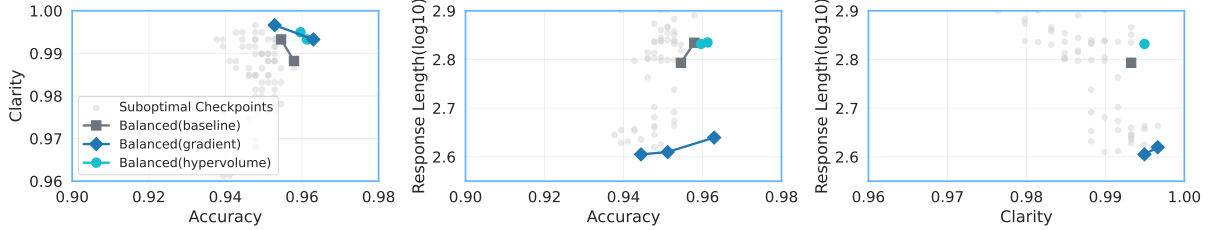
Figure 3: Reward weight evolution over training.

ent weight configurations in Table 2. Our method consistently outperforms all baselines in multi-objective alignment across different online RL algorithms. The corresponding Pareto fronts visualized in Figure 10 (Appendix A.5) further support these findings, demonstrating that our method generates superior Pareto fronts that dominate all baseline approaches under both GRPO and REINFORCE training.

Learning differs for objectives. We analyze the evolution of objective weights during training to better understand the relative importance of each objective. As shown in Figure 3, the weight for conciseness rapidly converges to approximately 0.2, with the lost weight mostly shifted to accuracy. Consequently, the accuracy weight exhibits continuous growth, which aligns with our intuition that accuracy is a more challenging objective requiring continual learning. In contrast, conciseness improvements can be achieved rapidly (see Figure 8 in Appendix A.2) and thus require less learning effort. The higher weights for accuracy and clarity compared to conciseness also support our preliminary findings that accuracy and clarity objectives are highly intertwined in the optimization process, effectively playing similar roles in



(a) Pareto fronts of Qwen3-8B trained with GRPO on MATH500 (Lightman et al., 2023) problems.



(b) Pareto fronts of Qwen3-8B trained with GRPO on MATH (Hendrycks et al., 2021) algebra problems.

Figure 4: Pareto fronts under different datasets.

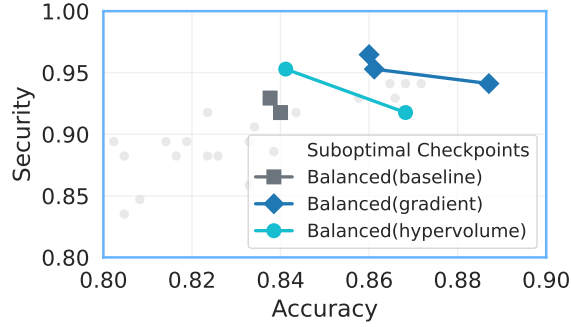


Figure 5: Pareto fronts of Qwen3-8B trained with SafeSQL (Yao et al., 2025) problems.

model updates as evidenced by their consistently low KL-divergence trajectories shown in Figure 7, and driving LLM learning towards the same direction. Conversely, conciseness works orthogonally to these objectives, applying a different influence on model training. We provide further analysis in Appendix A.2.

6 Generalizability and Convergence Rate Tests

To demonstrate the training efficiency of dynamic reward weighting, we compute the average number of training steps required to achieve the current Pareto fronts. As illustrated in Table 3, while the hypervolume-guided method that proactively pushes Pareto fronts has only marginal efficiency gains compared to baselines, the gradient-based method consistently has a higher convergence rate, reducing the required steps by 6.1 on average

Method →	Hypervolume-guided		Gradient-based	
Online RL ↓	Baseline	Ours (Δ)	Baseline	Ours (Δ)
GRPO	66.5	64.0 (-2.5)	71.5	62.6 (-8.9)
REINFORCE	63.8	65.5 (+1.7)	68.5	67.2 (-1.3)
RLOO	61.1	62.5 (+1.4)	68.7	60.5 (-8.2)

Table 3: Average number of training steps required to reach Pareto fronts for Hypervolume-Guided and Optimization methods versus their respective baselines.

across RL algorithms.

We further validate the generalizability of our two methods by extending our experiments to two additional datasets spanning mathematical and coding problems (Figure 4 and Figure 5) and three model families (Figure 6). For fair comparison, we use the same constant learning rate across both methods and the baseline. We adopt identical objective sets for the mathematical datasets. For coding tasks, we use SafeSQL, a benchmark designed to evaluate SQL injection vulnerability. Each task requires generating code that constructs SQL queries which both retrieve correct results from a given database and are robust to injection attacks. Accordingly, SafeSQL involves two objectives: *accuracy* and *security*. We report the SafeSQL results in Figure 5. **Similar to our main experiment results, our two methods achieve superior trade-offs compared to the baseline, with the gradient-based weighting showing the best overall performance.**

Notably, several panels show the Pareto front

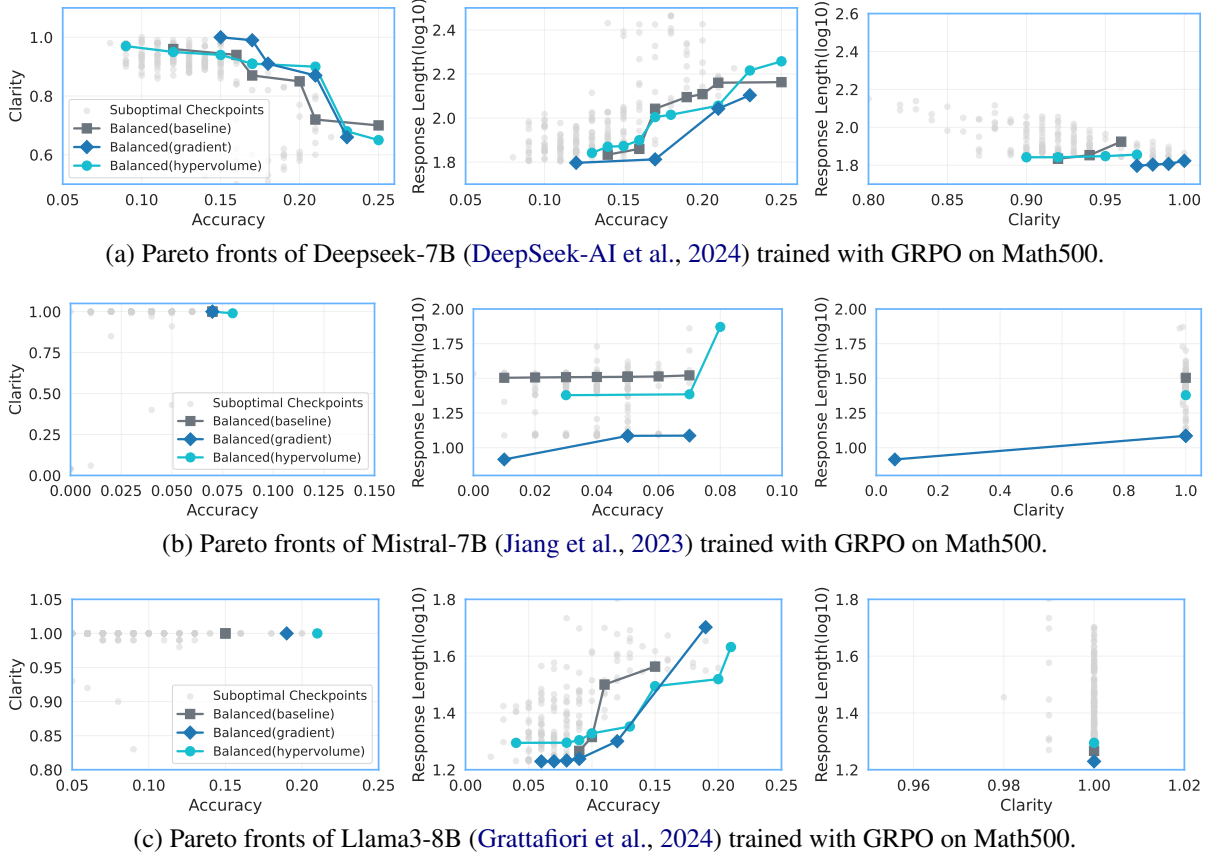


Figure 6: Pareto fronts under different models.

collapsing to a single point (e.g., accuracy–clarity plots for Mistral-7B in Figure 6b and Llama3-8B in Figure 6c). This reflects a ceiling effect where clarity quickly saturates near its maximum and exhibits little variation across runs. Consequently, Pareto-optimal checkpoints become indistinguishable along the clarity dimension and overlap in two-dimensional projections. The meaningful trade-offs are instead captured by other objective pairs such as accuracy versus response length.

7 Conclusion and Future Work

We found that different objectives in multi-objective LLM alignment require varying learning efforts and therefore proposed dynamic reward weighting. We introduced two dynamic reward weighting approaches of increasing sophistication: hypervolume-guided weight adaptation and gradient-based weight optimization. Our experiments across multiple online RL algorithms, datasets, and model families demonstrated that both methods consistently outperform fixed-weight linear scalarization baselines, achieving superior Pareto optimal solutions while improving training efficiency.

Future Work. While our dynamic reward weighting proves effective across various models, its success depends critically on a model’s inherent capacity to simultaneously improve multiple objectives. For instance, when we evaluated Mistral-8B-Instruct (Mistral AI Team, 2024) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) under identical multi-objective settings, we consistently observe that reducing response length leads to accuracy degradation, revealing a fundamental conflict. In such cases, our methods yield limited gains. We think this limitation likely stems from either performance saturation due to pretraining or insufficient model capacity to effectively learn certain objectives. Both issues are difficult to resolve through post-training alone. Therefore, these findings suggest two promising research directions: (1) developing pretraining strategies that equip models to learn and balance diverse objectives more effectively, and (2) designing post-training approaches for determining which objectives should be prioritized when irreducible conflicts arise.

Acknowledgement

We thank the anonymous reviewers and the action editor, Jesse Thomason, for their valuable feedback. This work was partially supported by NSF IIS-2119531, IIS-2137396, IIS-2142827, IIS-2234058, and Coefficient Giving. We also appreciate the support from Amazon, the Foundation Models and Applications Lab of Lucy Institute, and ND-IBM Tech Ethics Lab.

References

- Pranjal Aggarwal and Sean Welleck. 2025. [L1: Controlling how long a reasoning model thinks with reinforcement learning](#).
- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. [Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms](#).
- Amir Beck and Marc Teboulle. 2003. [Mirror descent and nonlinear projected subgradient methods for convex optimization](#). *Operations Research Letters*, 31(3):167–175.
- S.P. Boyd and L. Vandenberghe. 2004. [Convex Optimization](#). Number pt. 1 in *Berichte über verteilte messysteme*. Cambridge University Press.
- Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. [Acereason-nemotron: Advancing math and code reasoning through reinforcement learning](#).
- DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. [Deepseek llm: Scaling open-source language models with longtermism](#).
- Jean-Antoine Désidéri. 2012. [Multiple-gradient descent algorithm \(mgda\) for multiobjective optimization](#). *Comptes Rendus Mathématique*, 350(5):313–318.
- Michael TM Emmerich and André H Deutz. 2018. [A tutorial on multiobjective optimization: fundamentals and evolutionary methods](#). *Natural computing*, 17:585–609.
- Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2024. [Doge: domain reweighting with generalization estimation](#). In *Proceedings of the 41st International Conference on Machine Learning*, ICML’24. JMLR.org.
- Peter J. Fleming, Robin C. Purshouse, and Robert J. Lygoe. 2005. [Many-objective optimization: An engineering design perspective](#). In *Evolutionary Multi-Criterion Optimization*, pages 14–32, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Carlos M. Fonseca and Peter J. Fleming. 1996. [On the performance assessment and comparison of stochastic multiobjective optimizers](#). In *Parallel Problem Solving from Nature — PPSN IV*, pages 584–593, Berlin, Heidelberg. Springer Berlin Heidelberg.
- C.M. Fonseca, L. Paquete, and M. Lopez-Ibanez. 2006. [An improved dimension-sweep algorithm for the hypervolume indicator](#). In *2006 IEEE International Conference on Evolutionary Computation*, pages 1157–1163.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil

Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Gefert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Manan Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal

Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collet, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie DelPierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papanikos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal,

Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabza, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshchev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth

Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojuan Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. [The llama 3 herd of models](#).

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiuotė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. 2023. [Studying large language model generalization with influence functions](#).

Shangding Gu, Bilgehan Sel, Yuhao Ding, Lu Wang, Qingwei Lin, Alois Knoll, and Ming

- Jin. 2024. [Safe and balanced: A framework for constrained multi-objective reinforcement learning](#).
- Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. [Controllable preference optimization: Toward controllable multi-objective alignment](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1437–1454, Miami, Florida, USA. Association for Computational Linguistics.
- Conor F Hayes, Roxana Rădulescu, Eugenio Bar-giacchi, Johan Källström, Matthew Macfar-lane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. 2022. [A practical guide to multi-objective reinforcement learning and planning](#). *Autonomous Agents and Multi-Agent Systems*, 36(1):26.
- Qiang He and Setareh Maghsudi. 2025. [Pareto multi-objective alignment for language models](#).
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the math dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.
- Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Zilin Zhu, Weixun Wang, Songlin Jiang, Hao-ran Wang, Hao Chen, Bin Chen, Weikai Fang, Xianyu, Yu Cao, and Haotian Xu. 2025. [Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models](#).
- Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, Jianguo Li, and Junbo Zhao. 2025. [Reinforcement learning with rubric anchors](#).
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#).
- Pang Wei Koh and Percy Liang. 2017. [Understanding black-box predictions via influence functions](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.
- Harshat Kumar, Alec Koppel, and Alejandro Ribeiro. 2023. [On the sample complexity of actor-critic method for reinforcement learning with function approximation](#). *Mach. Learn.*, 112(7):2433–2467.
- Chengao Li, Hanyu Zhang, Yunkun Xu, Hongyan Xue, Xiang Ao, and Qing He. 2025a. [Gradient-adaptive policy optimization: Towards multi-objective alignment of large language models](#).
- Zhuo Li, Guodong Du, Weiyang Guo, Yigeng Zhou, Xiucheng Li, Wenya Wang, Fangming Liu, Yequan Wang, Deheng Ye, Min Zhang, and Jing Li. 2025b. [Multi-objective large language model alignment with hierarchical experts](#).
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. [Let’s verify step by step](#).
- Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025. [Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models](#).
- Suyun Liu and Luis Nunes Vicente. 2024. [The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning](#). *Annals of Operations Research*, 339(3):1119–1148.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. [Learning word vectors for sentiment analysis](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon,

- USA. Association for Computational Linguistics.
- Mistral AI Team. 2024. [Ministral-8b-instruct-2410](#).
- Subhojyoti Mukherjee, Anusha Lalitha, Sailik Sengupta, Aniket Deshmukh, and Branislav Kveton. 2024. [Multi-objective alignment of large language models through hypervolume maximization](#).
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#).
- Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2023. [Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 71095–71134. Curran Associates, Inc.
- Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. [A survey of multi-objective sequential decision-making](#). *J. Artif. Int. Res.*, 48(1):67–113.
- Roxana Rădulescu, Patrick Mannion, Yijie Zhang, Diederik M. Roijers, and Ann Nowé. 2020. [A utility-based analysis of equilibria in multi-objective normal-form games](#). *The Knowledge Engineering Review*, 35:e32.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#).
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#).
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. [Hybridflow: A flexible and efficient rlhf framework](#). In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys ’25, page 1279–1297. ACM.
- Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. [Policy gradient methods for reinforcement learning with function approximation](#). In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. 2025. [Kimi k1.5: Scaling reinforcement learning with llms](#).
- Kaiwen Wang, Rahul Kidambi, Ryan Sullivan, Alekh Agarwal, Christoph Dann, Andrea Michi, Marco Gelmi, Yunxuan Li, Raghav Gupta, Kumar Avinava Dubey, Alexandre Rame, Johan Ferret, Geoffrey Cideron, Le Hou, Hongkun Yu, Amr Ahmed, Aranyak Mehta, Leonard Hussenot, Olivier Bachem, and Edouard Leurent. 2024. [Conditional language policy: A general framework for steer-](#)

- able multi-objective finetuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2153–2186, Miami, Florida, USA. Association for Computational Linguistics.
- Tianze Wang, Dongnan Gui, Yifan Hu, Shuhang Lin, and Linjun Zhang. 2025. [Mpo: An efficient post-processing framework for mixing diverse preference alignment](#).
- Yue Wang, Wei Chen, Yuting Liu, Zhi-Ming Ma, and Tie-Yan Liu. 2017. [Finite sample analysis of the gtd policy evaluation algorithms in markov setting](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Susan Wei and Marc Niethammer. 2022. [The fairness-accuracy pareto front](#). *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 15(3):287–302.
- Yue Frank Wu, Weitong ZHANG, Pan Xu, and Quanquan Gu. 2020. [A finite-time analysis of two time-scale actor-critic methods](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 17617–17628. Curran Associates, Inc.
- Zequi Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. [Fine-grained human feedback gives better rewards for language model training](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 59008–59033. Curran Associates, Inc.
- Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. [Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning](#).
- Nuoya Xiong and Aarti Singh. 2025. [Projection optimization: A general framework for multi-objective and multi-group rlhf](#).
- Mingjing Xu, Peizhong Ju, Jia Liu, and Haibo Yang. 2025. [Psmgd: Periodic stochastic multi-gradient descent for fast multi-objective optimization](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(20):21770–21778.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Kekun Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. [Qwen3 technical report](#).
- Kailai Yang, Zhiwei Liu, Qianqian Xie, Jimin Huang, Tianlin Zhang, and Sophia Ananiadou. 2024a. [Metaaligner: Towards generalizable multi-objective alignment of language models](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 34453–34486. Curran Associates, Inc.
- Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024b. [Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment](#). In *International Conference on Machine Learning*, pages 56276–56297. PMLR.
- Feng Yao, Zilong Wang, Liyuan Liu, Junxia Cui, Li Zhong, Xiaohan Fu, Haohui Mai, Vish Krishnan, Jianfeng Gao, and Jingbo Shang. 2025. [Training language models to generate quality code with program analysis feedback](#).
- Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, Qiaobo Chen, Yinyuting Yin, Hao Zhang, Tengfei Shi, Liang Wang, Qiang Fu, Wei Yang, and Lanxiao Huang. 2020. [Mastering complex control in moba games with deep reinforcement learning](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6672–6679.
- Jixiao Zhang and Chunsheng Zuo. 2025. [Grpo-lead: A difficulty-aware reinforcement learning](#)

approach for concise mathematical reasoning in language models.

Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Haojun Chen, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. 2024. [Panacea: Pareto alignment via preference adaptation for llms](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 7552–75558. Curran Associates, Inc.

Shiji Zhou, Wenpeng Zhang, Jiyan Jiang, Wenliang Zhong, Jinjie GU, and Wenwu Zhu. 2022. [On the convergence of stochastic multi-objective gradient manipulation and beyond](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 38103–38115. Curran Associates, Inc.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. 2024. [Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10586–10613, Bangkok, Thailand. Association for Computational Linguistics.

A Appendix

A.1 Proof for Gradient-Based Weight Optimization

We first write optimizing reward weights $\mathbf{w} \in \mathbb{R}^K$ along the training of the policy model θ as a bilevel-optimization problem:

$$\mathbf{w} \in \arg \min_{\mathbf{w} \in \mathbb{R}^K} \sum_{i \in [K]} J_i(\theta^*(\mathbf{w})), \quad s.t. \quad \theta^*(\mathbf{w}) \in \arg \min_{\theta} \sum_{i \in [K]} w_i J_i(\theta).$$

The lower-level objective $\theta(\mathbf{w})$ is updated using the weights $\mathbf{w}$, and the higher-level objective $\mathbf{w}$ is found based on the loss function of the updated model. With a greedy approximation of $\min_{\mathbf{w}} \sum_i J_i(\theta^{(t)}(\mathbf{w}))$ (for simplicity, we denote $\theta^{(t)}(\mathbf{w}) = \theta^{(t)}$ hereafter), we search for the optimal reward weights $\mathbf{w}^{(t)}$ at step t to minimize the average loss over K objectives at step $t + 1$:

$$\arg \min_{\mathbf{w} \in \mathbb{R}^K} \bar{J}(\theta^{(t+1)}) = \arg \min_{\mathbf{w} \in \mathbb{R}^K} \sum_{i \in [K]} J_i(\theta^{(t+1)}) = \arg \min_{\mathbf{w} \in \mathbb{R}^K} \sum_{i \in [K]} [J_i(\theta^{(t+1)}) - J_i(\theta^{(t)})] \quad (4)$$

From the first-order Taylor approximation, we have

$$\begin{aligned} J_i(\theta^{(t+1)}) &= J_i(\theta^{(t)}) + \nabla J_i(\theta^{(t)}) \cdot (\theta^{(t+1)} - \theta^{(t)}) + \mathcal{O}(\|\theta^{(t+1)} - \theta^{(t)}\|) \\ &= J_i(\theta^{(t)}) + \nabla J_i(\theta^{(t)}) \cdot \left[-\eta^{(t)} \sum_{i \in [K]} w_i^{(t)} \nabla J_i(\theta^{(t)}) \right] + \mathcal{O}(\|\mathbf{w}\|), \end{aligned}$$

where $\mathcal{O}(\|\theta^{(t+1)} - \theta^{(t)}\|) = \mathcal{O}(\|\mathbf{w}\|)$ as the high-order remainder from Tylor approximation. Therefore, we can write Eq. 4 as:

$$\begin{aligned} \arg \min_{\mathbf{w} \in \mathbb{R}^K} \bar{J}(\theta^{(t+1)}) &= \arg \min_{\mathbf{w} \in \mathbb{R}^K} \sum_{i \in [K]} \nabla J_i(\theta^{(t)}) \cdot \left[-\eta^{(t)} \sum_{k \in [K]} w_k^{(t)} \nabla J_k(\theta^{(t)}) \right] + \mathcal{O}(\|\mathbf{w}\|) \\ &= \arg \min_{\mathbf{w} \in \mathbb{R}^K} -\eta^{(t)} \sum_{i \in [K]} \nabla J_i(\theta^{(t)}) \cdot \left[\sum_{k \in [K]} w_k^{(t)} \nabla J_k(\theta^{(t)}) \right] + \mathcal{O}(\|\mathbf{w}\|) \\ &= \arg \min_{\mathbf{w} \in \mathbb{R}^K} -\eta^{(t)} \sum_{i \in [K]} w_i^{(t)} \left[\nabla J_i(\theta^{(t)}) \cdot \sum_{k \in [K]} \nabla J_k(\theta^{(t)}) \right] + \mathcal{O}(\|\mathbf{w}\|) \\ &= \arg \min_{\mathbf{w} \in \mathbb{R}^K} -\eta^{(t)} \sum_{i \in [K]} w_i^{(t)} I_i^{(t)} + \mathcal{O}(\|\mathbf{w}\|), \quad \text{where } I_i^{(t)} \triangleq \langle \nabla J_i(\theta^{(t)}), \sum_{k \in [K]} \nabla J_k(\theta^{(t)}) \rangle \\ &= \arg \min_{\mathbf{w} \in \mathbb{R}^K} -\eta^{(t)} \langle \mathbf{w}, \mathbf{I}^{(t)} \rangle + \mathcal{O}(\|\mathbf{w}\|), \quad \text{where } \mathbf{I}^{(t)} = [I_1^{(t)}, I_2^{(t)}, \dots, I_K^{(t)}]. \end{aligned}$$

Follow common practice in mirror descent (Beck and Teboulle, 2003), we estimate $\mathcal{O}(\|\mathbf{w}\|)$ by introducing a regularization term with factor μ via Bregman divergence $D_h(\mathbf{w} \|\mathbf{w}^{(t-1)}) = h(\mathbf{w}) - h(\mathbf{w}^{(t-1)}) - \langle \nabla h(\mathbf{w}^{(t-1)}), \mathbf{w} - \mathbf{w}^{(t-1)} \rangle$, where we employ the entropy function $h(\mathbf{w}) = \sum_i w_i \ln w_i$. So the above equation becomes:

$$\begin{aligned} \mathbf{w}^{(t)} &:= \arg \min_{\mathbf{w} \in \mathbb{R}^K} -\eta^{(t)} \langle \mathbf{w}, \mathbf{I}^{(t)} \rangle + \mu \left(h(\mathbf{w}) - h(\mathbf{w}^{(t-1)}) - \langle \nabla h(\mathbf{w}^{(t-1)}), \mathbf{w} - \mathbf{w}^{(t-1)} \rangle \right) \quad (5) \\ &= \arg \min_{\mathbf{w} \in \mathbb{R}^K} -\eta^{(t)} \langle \mathbf{w}, \mathbf{I}^{(t)} \rangle + \mu \left(h(\mathbf{w}) - \langle \nabla h(\mathbf{w}^{(t-1)}), \mathbf{w} \rangle \right). \end{aligned}$$

We then take the derivative of Eq. 5:

$$\begin{aligned} \nabla \left(-\eta^{(t)} \langle \mathbf{w}, \mathbf{I}^{(t)} \rangle + \mu \left(h(\mathbf{w}) - \langle \nabla h(\mathbf{w}^{(t-1)}), \mathbf{w} \rangle \right) \right) &= 0 \\ -\eta^{(t)} \mathbf{I}^{(t)} + \mu [\ln w + 1]_i - \mu [\ln w^{(t-1)} + 1]_i &= 0 \\ \Rightarrow \ln w^{(t)} &= \ln w^{(t-1)} + \frac{\eta^{(t)} \mathbf{I}^{(t)}}{\mu}. \end{aligned}$$

Therefore, we can get the following update rule for reward weights with normalization:

$$\mathbf{w}^{(t)} = \frac{\mathbf{w}^{(t)}}{\sum_k w_k^{(t)}}, \quad \text{where } \mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} \odot \exp\left(\frac{\eta^{(t)} \mathbf{I}^{(t)}}{\mu}\right), \mathbf{I}_i^{(t)} = \langle \nabla J_i(\theta^{(t)}), \sum_{k \in [K]} \nabla J_k(\theta^{(t)}) \rangle. \quad (6)$$

A.2 Preliminary Findings

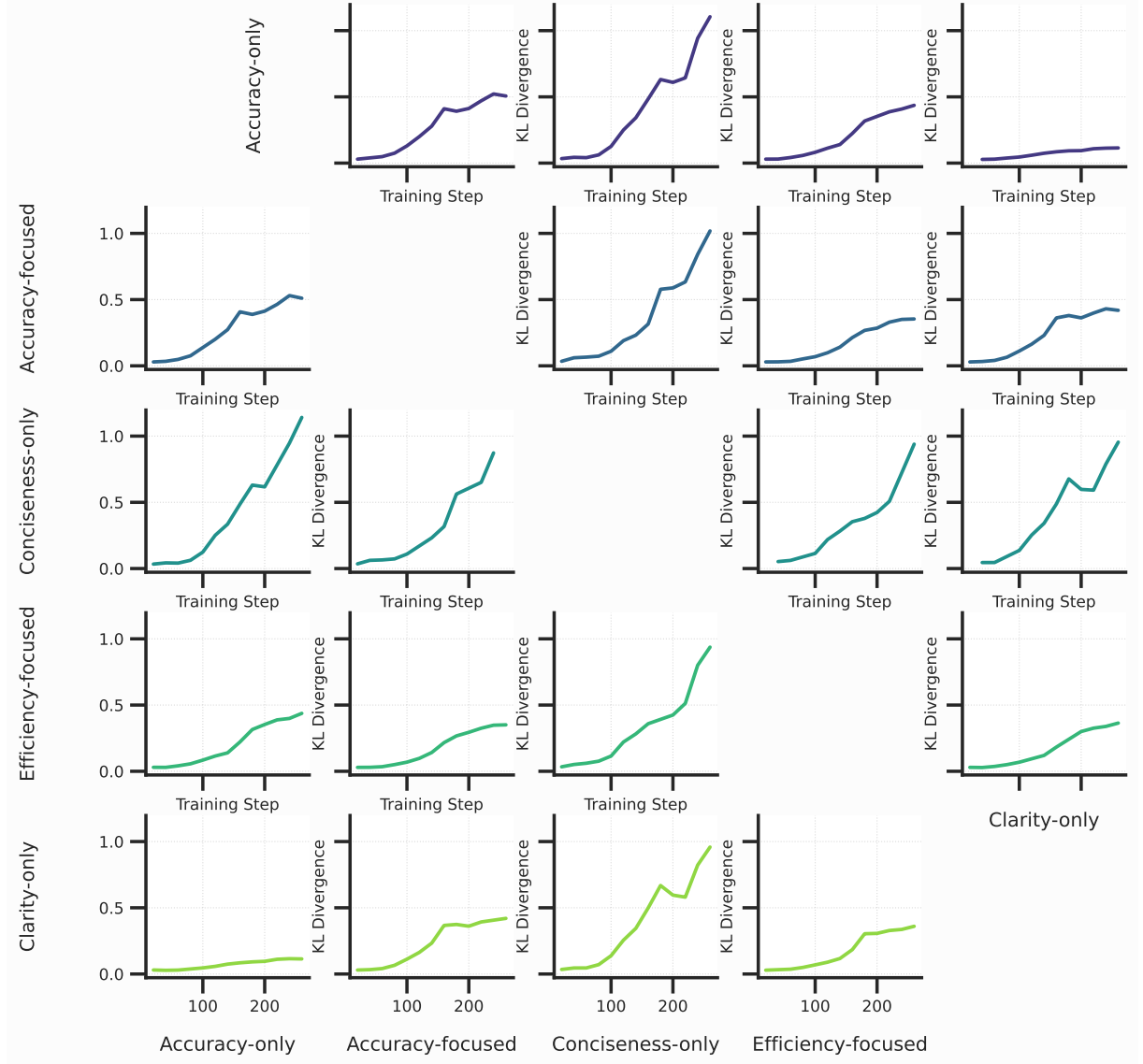


Figure 7: Pairwise KL divergence trajectories between models trained under different weight configurations. Rows represent the model’s own weight configurations and columns the configuration it is compared against; within each panel the KL-divergence (y-axis) is plotted over training steps (x-axis).

Accuracy and clarity objectives exhibit synergistic effects while conciseness drives learning orthogonally. To understand the synergistic or antagonistic effect between different objectives in model training, we analyze pairwise KL divergence trajectories between models trained with different weight configurations: accuracy-focused (0.5, 0.25, 0.25), efficiency-focused (0.25, 0.375, 0.375), accuracy-only (1, 0, 0), conciseness-only (0, 1, 0), and clarity-only (0, 0, 1). We conduct this study using the Qwen3-8B model trained on the Math500 dataset via GRPO. As shown in Figure 7, the KL divergence between the accuracy- and clarity-only configurations remains consistently low throughout training, indicating these objectives exhibit strong synergistic effects and induce similar parameter updates. In contrast, models

trained with conciseness-only weights diverge significantly from all accuracy- and clarity-only(focused) models, as evidenced by steadily increasing KL divergence over time. This suggests that conciseness applies an orthogonal influence on model optimization, steering parameter updates in a fundamentally different direction compared to accuracy and clarity.

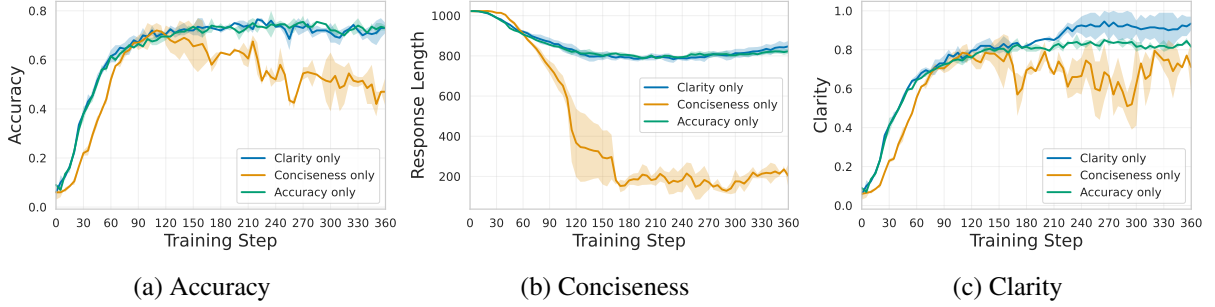


Figure 8: Validation performance of individual objectives during training. **Each objective reaches saturation at different training stages.** For example, under the conciseness-only weight configuration, the model achieves optimal response brevity at approximately step 165 (b), whereas the clarity-only configuration reaches the best clarity after step 240 (c).

Objectives exhibit varying convergence rates. When training individual objectives, we observe that different objectives reach saturation at different training stages, as illustrated in Figure 8.

A.3 hypervolume-Guided Reward Adaptation Details

We compute three types of rewards: (1) accuracy reward through exact matching against the provided ground truth, (2) conciseness reward by comparing current response length to the global average response length from previous rollouts, and (3) clarity reward by checking whether responses contain explicit reasoning steps (e.g., “first”, “second”, “third”) according to our pre-defined rules.

When training the REINFORCE algorithm with a constant learning rate, we observed significant variation in convergence rates across runs using identical hyperparameters, particularly for the conciseness and clarity objectives. To ensure robust and fair comparisons, we repeated runs three times for all baselines and our proposed approach, reporting results from the run that achieved the fastest convergence in each case. This phenomenon was not observed in other algorithms or training settings, and we leave its investigation to future works.

We also observed that zero reward would result in increased response length and degraded performance in both accuracy and formatting, which is likely due to the entropy penalty dominating the gradient updates. To mitigate this issue, we build a minimum threshold of $r_{\text{pareto}} = 0.5$ in Eq. 1 even when the current checkpoint is Pareto dominated (i.e., $\Delta\text{HV} = 0$).

A.4 Hyperparameters

We use verl (Sheng et al., 2025) for RL training. We provide the hyperparameters used for GRPO training on Math500 in Table 4 and Table 5.

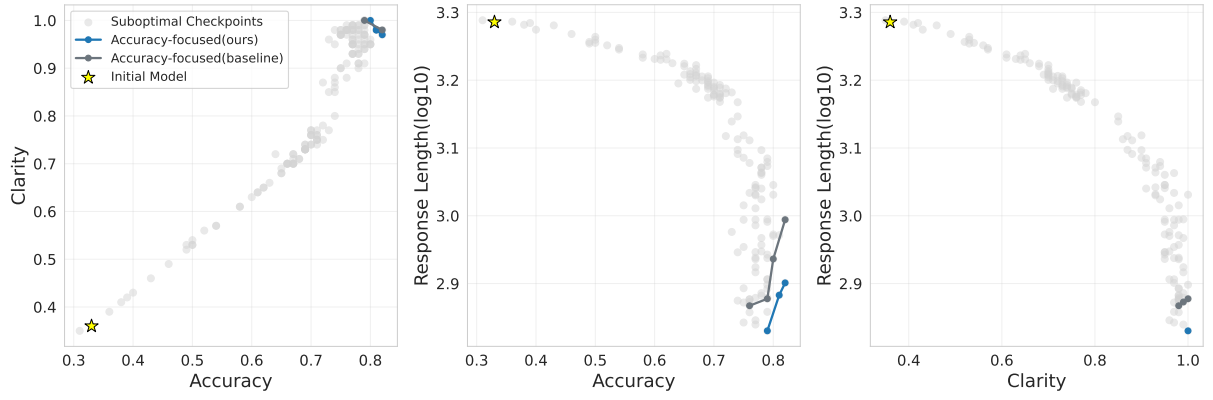
A.5 Pareto Fronts Visualization

Name	Search Bounds
r_{pareto}	$\left\{1 + \frac{1}{1 + \exp(-\Delta\text{HV})}, \frac{2}{1 + \exp(-\Delta\text{HV})}, \right.$ $\left.0.5 + 1.5\tanh(\Delta\text{HV}), 0.5 + 1.5 \tanh(3\Delta\text{HV})\right\}$
Learning rate	$1e^{-6}$
LR scheduler	constant
Batch size B	64
Mini-batch size	32
Epoch	90
Max response length	2048
Rollout size G	8
KL coefficient	0.001
Clip range ϵ	0.2
Dual clip constant c	3
Avg. training time	14 hrs
GPU used	8 Nvidia H200 (143 GB)

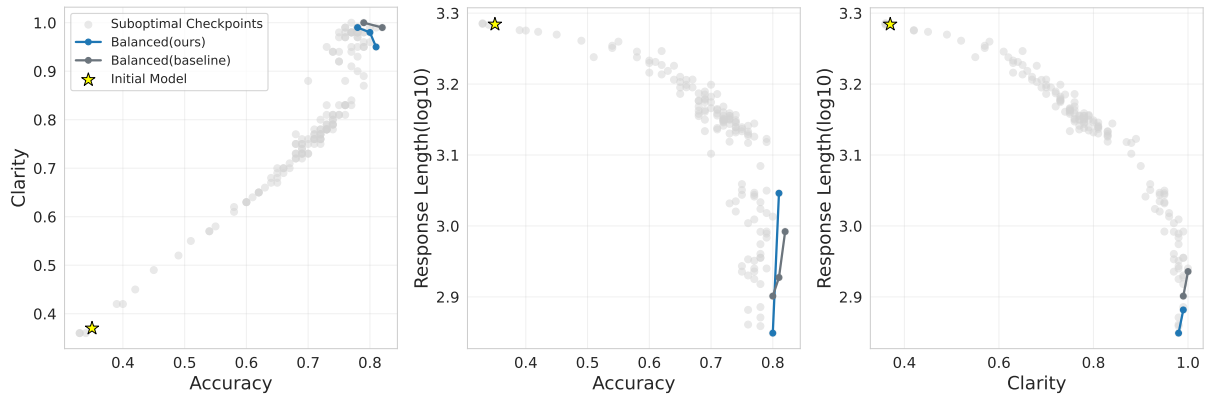
Table 4: Hyperparameters and other reproducibility information for hypervolume-based weight adaptation.

Name	Search Bounds
Learning rate η	$\{1e^{-4}, 1e^{-5}, \mathbf{1e^{-6}}\}$
LR scheduler	$\{\text{constant, cosine-annealing, exponential decay, } \mathbf{\text{polynomial}}\}$
Poly scheduler power	$\{1.01, \mathbf{1.03}\}$
Regularization factor μ	$\{1e^{-4}, \mathbf{1e^{-5}}, 1e^{-6}\}$
Clip range ϵ	$\{0.2, \mathbf{100}\}$
Dual clip constant c	$\{3, \mathbf{100}\}$
Batch size B	64
Mini-batch size	32
Epoch	90
Max response length	2048
Rollout size G	8
Max gradient norm	1.0
KL coefficient	0.001
Target layer IDs	$[10, 11, 12, \dots, 25]$
Avg. training time	24 hrs
GPU used	8 Nvidia H200 (143 GB)

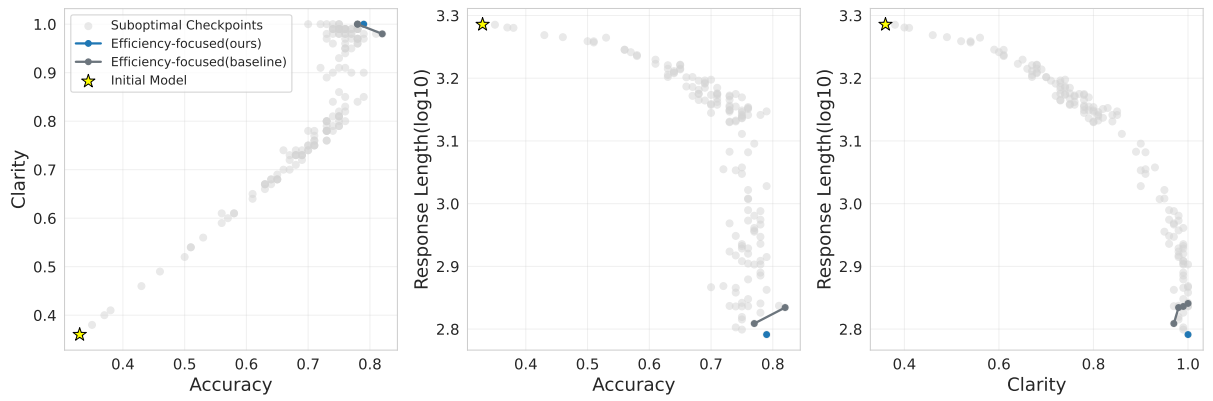
Table 5: Hyperparameters and other reproducibility information for gradient-based weight optimization.



(a) Accuracy-focused.

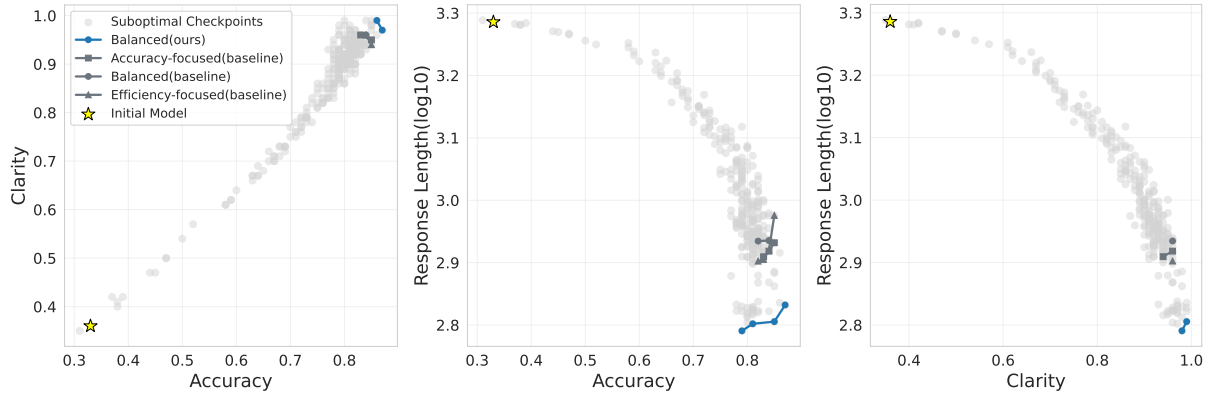


(b) Balanced.

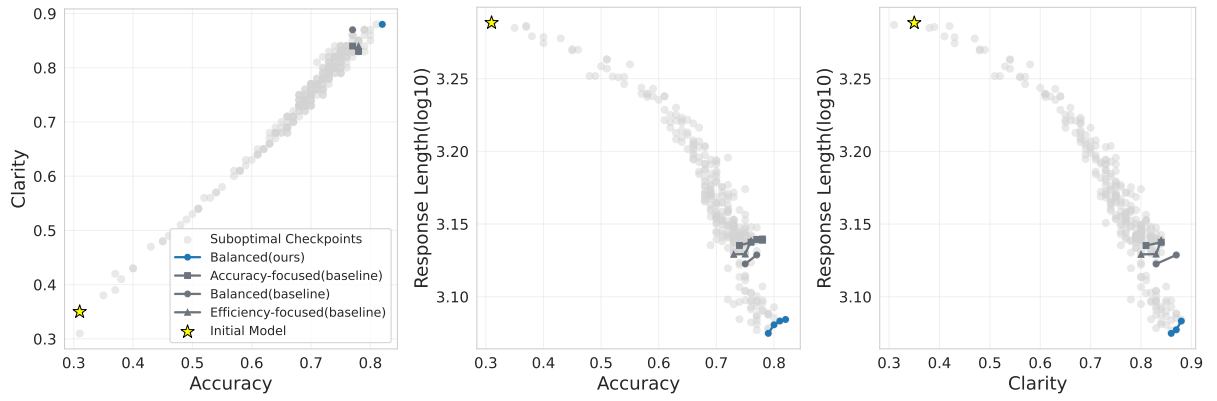


(c) Efficiency-focused.

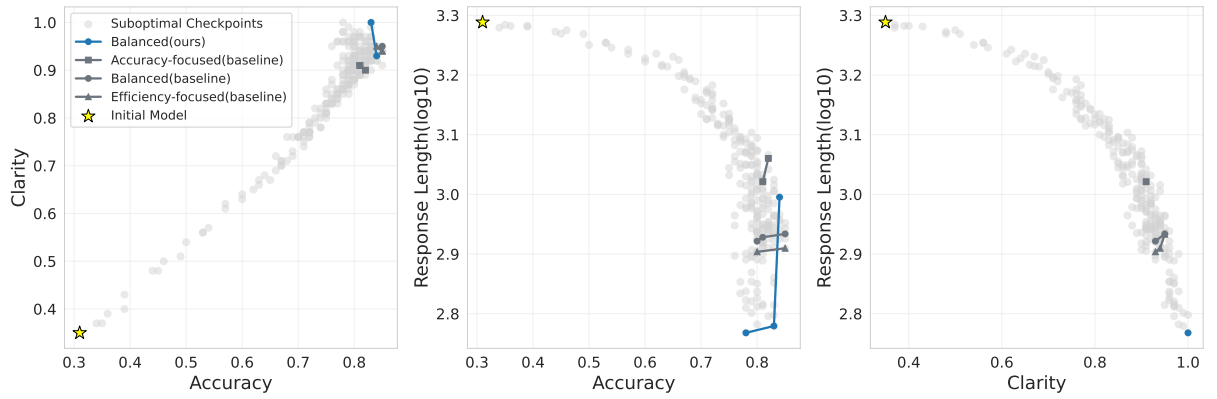
Figure 9: Pareto fronts obtained from our hypervolume-guided weight adaptation against baselines under REINFORCE training.



(a) GRPO.



(b) REINFORCE.



(c) RLOO.

Figure 10: Pareto fronts obtained from our gradient-based weight optimization against baselines under different RL training algorithms.