BoNVoyage: Learning Better Rewards without Ranking

Anonymous authors
Paper under double-blind review

Abstract

Reward models (RMs) play a central role in reinforcement learning from human feedback (RLHF), yet they are typically trained with a Bradley–Terry (BT) objective that ranks response pairs under a fixed text distribution. This objective does not match how RMs are used in RLHF: as training progresses, the policy increasingly concentrates on responses that receive high reward under the RM itself. But what matters is not whether the RM ranks fixed pairs correctly, but whether it remains reliable on the responses that *its own signal* makes more likely *under the policy*. We develop BoNVoyage, a method for training RMs directly for this downstream role. Instead of training a pairwise ranker, BoNVoyage maximizes the likelihood that the RM-induced RLHF-optimal policy assigns to preferred responses. Estimating the gradient of this objective requires samples from the optimal policy induced by the learning RM. We obtain these samples through test-time alignment: given the base LM and the RM, we use Markov chain Monte Carlo to sample from the induced policy. To make this practical, we use the base LM as an independent proposal distribution, reuse candidate responses across optimization steps, and use contrastive divergence with short chains initialized at preferred responses. We evaluate BoNVoyage by training 4B RMs for Qwen3-1.7B-Base in mathematics and our supervised finetuned model Qwen3-1.7B-Science in science. Compared with BT baselines, BoNVoyage yields stronger downstream policies under RLHF and stronger rerankers under best-of-n sampling across MATH-500, GSM8K, SciQ, SciKnowEval, ARC-Challenge, and MMLU-Science. On verifiable math tasks, BoNVoyage also shows substantially more robustness to reward over-optimization, preserving alignment between RM scores and ground-truth correctness later into RLHF training. On publication, we will release our code, trained models, and datasets.

1 Introduction

Reward models (RMs), which automatically assign scalar scores to language model (LM) outputs estimating quality in some dimension of interest, play a central role in reinforcement learning from human feedback (RLHF; Ziegler et al., 2020; Stiennon et al., 2022). Beyond that, they underpin synthetic-data generation pipelines (Llama, 2024) and test-time alignment methods such as best-of-n (BoN; Gao et al., 2022; Stiennon et al., 2020), weighted majority voting (WMV; Li et al., 2023) and QAlign (Faria & Smith, 2025).

Although RM research has shown steady improvements in pairwise accuracy on leaderboards such as RewardBench (Lambert et al., 2024b) and RM-Bench (Liu et al., 2024b), these gains do not translate to downstream policy quality and often lead to reward hacking (Razin et al., 2025; Wen et al., 2025; Ivison et al., 2024; Chen et al., 2024). While RM benchmarks continue to evolve, for instance by targeting downstream correlation with BoN (Malik et al., 2025; Frick et al., 2024; Zhou et al., 2025) or downstream DPO (Rafailov et al., 2024) performance (Frick et al., 2024), they do not solve the fundamental problem of measuring RM utility in RLHF.

Due in part to these limitations, many recent large-scale efforts have opted not to use RMs during RL, relying instead on *verifiable* signals available in restricted domains such as math and code generation (DeepSeek-AI, 2025; Lambert et al., 2024a; Shao et al., 2024). Extended training under such crisp feedback has led to the advent of *reasoning models* that emit long thought traces before

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

producing a final answer, yielding large performance gains. However, most real-world tasks lack any verifiable ground truth, and in specialized expert domains, like medicine and law, judging correctness beyond MQA often requires expert knowledge, and obtaining new expert labels is progressively more expensive as we go up the capability ladder. Thus, robust reward modeling remains a valuable research direction.

An RM assigns a single scalar score to any complete prompt–response pair. Training is accomplished contrastively on human-preference pairs (Stiennon et al., 2022; Ziegler et al., 2020) or synthetically generated ones (Bai et al., 2022b). RMs are typically learned using maximum-likelihood estimation (MLE) under a Bradley-Terry model (BT; (Bradley & Terry, 1952)). In practice, this approach requires prompts each paired with two outputs, one marked as preferred (preference data).

We argue that the central issue is how RMs are trained. The standard Bradley-Terry objective trains the RM to rank response pairs correctly under a fixed text distribution. However, this does not match the data distribution in RLHF: the policy, rather than producing responses covering a wide range of rewards, will instead concentrate more and more on high-reward responses over the course of training. As a result, what matters is not whether the RM is well-calibrated on average, but whether it is reliable on the responses that *its own signal* makes more likely to be selected by the resulting policy (Zhang et al., 2026; Ziegler et al., 2020). One solution is iteratively retraining the RM on on-policy data (Ziegler et al., 2020; Bai et al., 2022a; Stiennon et al., 2022; Llama, 2024; 2023), but this is costly to scale with human-preference feedback.

Guided by this perspective, we develop BoNVoyage, a method to train RMs directly for their downstream role. Instead of training a pairwise ranker, **we maximize the likelihood that the RM-induced RLHF-optimal policy assigns to the preferred responses** with respect to the RM parameters. Our key insight is that we can estimate the gradient of this objective using samples from the RLHF-optimal policy (before it exists) through **test-time alignment** given the *base* LM and the learning RM, a problem already studied by multiple works (Faria & Smith, 2025; Huang et al., 2025; Lew et al., 2023; Loula et al., 2025). Following Faria & Smith (2025), we use Markov chain Monte Carlo (MCMC) for test-time alignment. To make MCMC practical, we use the base LM as an independent proposal distribution. Candidate responses can therefore be drawn once before training and reused across optimization steps. Since running the chain to convergence at every update would be expensive, we employ contrastive divergence (Hinton, 2002), initializing short chains at preferred responses and yielding a tractable approximation to the MLE gradient.

In our experiments (§4), we study three key research questions. First, do RMs trained with BoNVoyage lead to better downstream policies after RL than BT-trained RMs? Second, how well do these RMs support test-time scaling through BoN sampling? Third, does BoNVoyage produce RMs that are more robust to reward over-optimization during RL? To answer these questions, we train 4B RMs with BoNVoyage for Qwen3-1.7B-Base (Yang et al., 2025) in two domains: (i) *mathematics*, where correctness is verifiable, and (ii) *science*, where it is not. Across all these settings, BoNVoyage consistently outperforms BT, yielding stronger final policies in both domains across MATH-500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), SciQ (Johannes Welbl, 2017), SciKnowEval Feng et al. (2024), ARC-Challenge (Clark et al., 2018), and MMLU-Science (Gema et al., 2025), while also remaining competitive with RL using verifiable rewards (RLVR) in the setting where such a comparison is possible. Furthermore, in mathematics, we directly measure reward over-optimization for both BT baselines and show that BoNVoyage is substantially more robust to it. Figure 1 summarizes this behavior.

2 Background

Reward modeling Given a dataset $\{(x^{(k)}, y_+^{(k)}, y_-^{(k)})\}_{k=1}^{M}$ where each tuple contains a prompt $x^{(k)}$ and a preferred/dispreferred response pair, RMs are typically trained using the model of Bradley & Terry (1952) (BT). Under this model, the probability that y_+ is preferred over y_- for prompt x is

$$p(y_+ \succ y_- \mid x; \phi) = \sigma\Big(r_\phi(y_+, x) - r_\phi(y_-, x)\Big), \tag{1}$$

where $r_\phi(y, x)$ denotes the scalar reward assigned by the model to response y under prompt x. The RM parameters ϕ are trained by maximizing the log likelihood, under this model, of the collection of observed preference pairs.

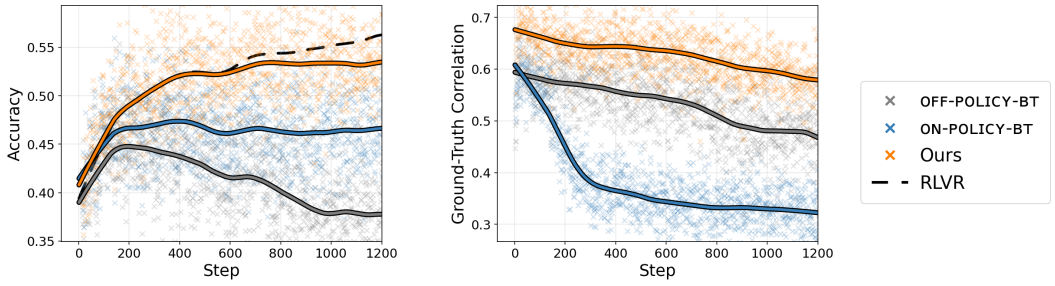


Figure 1: RL training on math data (DeepScaleR). We compare QWEN3-1.7B-BASE (Yang et al., 2025) optimized with three 4B RMs: ×**ON-POLICY-BT** , an on-policy trained BT RM; ×**OFF-POLICY-BT** , an off-policy BT RM given by SKYWORK-REWARD-V2-QWEN3-4B (Liu et al., 2026); and ×**OURS**, an RM trained with BONVOYAGE. All learned RMs are initialized from QWEN3-4B-BASE. **Left:** training accuracy as a function of RL step. The dashed curve corresponds to **RLVR**, and serves as a reference upper bound on the training accuracy attainable when the reward is perfectly faithful. **Right:** Pearson correlation between RM scores assigned to policy outputs at each RL step and their ground-truth correctness. We do not plot RLVR on the right because its reward is, by construction, perfectly correlated with correctness.

Preference data Some documented efforts from frontier labs collect preference annotations from humans (Ziegler et al., 2020; Stiennon et al., 2022; Bai et al., 2022a; Llama, 2023; 2024). However, high-quality human preference annotation is expensive and limited by time, labor, and annotator expertise, so open-source efforts mainly use prompts from diverse sources and candidate responses from multiple models, with supervision derived from a mix of human judgments, LM-as-a-judge evaluation, and other automatic metrics (Lambert et al., 2024a; Liu et al., 2024a; Dong et al., 2024; Cui et al., 2024; Olmo et al., 2025; Liu et al., 2026).

Learning with RMs Given a prompt distribution $\mathcal{X}$, a base language model $p_{\mathrm{LM}}(y \mid x)$, and a reward model $r_\phi(y, x)$, reinforcement learning with human feedback (RLHF) (Stiennon et al., 2022; Fernandes et al., 2023; Kaufmann et al., 2023; Peters & Schaal, 2007; Peng et al., 2019; Korbak et al., 2022b;a; Go et al., 2023) seeks a policy $q_\theta(y \mid x)$ that maximizes

$$\mathbb{E}_{x\sim\mathcal{X}}\left[\mathbb{E}_{y\sim q_\theta(y|x)}\left[r_\phi(y,x)\right] - \beta D_{\mathrm{KL}}\left(q_\theta(y \mid x) \,\|\, p_{\mathrm{LM}}(y \mid x)\right)\right]. \tag{2}$$

The KL term regularizes the policy towards the base model $p_{\mathrm{LM}}(y \mid x)$, preventing the optimization from drifting from the original model distribution, and the hyperparameter β controls its strength.

Distribution shift When training an RM with BT, we teach it to rank responses drawn from some fixed response distribution we will call p_0. p_0 may be a mixture of responses from many LMs, as in the case of **offline preferences**, or from a specific LM we ultimately want to optimize with RL, our *base* LM $p_{\mathrm{LM}}(y \mid x)$, as in the case of **online preferences** (Ziegler et al., 2020; Bai et al., 2022a). In either case, this fixed distribution p_0 is not the one on which the RM will be queried during RL. Once RL begins, the policy changes, and with it the distribution of responses presented to the RM.

Over time, optimization shifts the policy toward responses that receive high reward. As a result, the RM will be asked to score responses that can differ substantially from those on which it was trained. This is problematic, because RM reliability under p_0 does not guarantee RM reliability later in training. More importantly, this shift in distribution is not arbitrary, but *adversarial* (Bukharin et al., 2025; Hardt et al., 2015; Brückner & Scheffer, 2011). Any flaw in the RM defines a direction of improvement for RL, whether or not that direction aligns with task performance (Razin et al., 2025; Wen et al., 2025; Ivison et al., 2024; Chen et al., 2024), as shown in Figure 1. This suggests that, if our goal is to learn RMs for RL, then we should train them not merely to rank responses correctly under p_0, but to remain a reliable signal for improvement on the distribution induced by RL itself.

3 LEARNING REWARD MODELS FOR REINFORCEMENT LEARNING

In this section, we develop a method for learning RMs that is aware of the response distribution induced during RL, so that once trained, the RM remains reliable on the **resulting** on-policy responses rather than only on the initial fixed distribution p_0.

A natural way to formalize this is through the optimal policy induced by a fixed RM $r_\phi(y, x)$ under the RLHF objective in Eq. equation 2. As shown in (Korbak et al., 2022a; Rafailov et al., 2024), that policy can be written as:

$$\pi_\phi(y \mid x) \triangleq \frac{1}{Z_\phi(x)} p_{\mathrm{LM}}(y \mid x) \exp\left(\frac{r_\phi(y, x)}{\beta}\right), \tag{3}$$

where $Z_\phi(x) = \sum_{y \in \mathcal{Y}} p_{\mathrm{LM}}(y \mid x) \exp(r_\phi(y, x)/\beta)$ is the partition function. This expression makes explicit how the RM shapes the downstream final policy. Although both computing $Z_\phi(x)$ and sampling exactly from $\pi_\phi(y \mid x)$ are intractable, this is the distribution that RLHF targets under the fixed RM. Our goal will therefore be to learn the RM such that the optimal policy $\pi_\phi(y \mid x)$ gives high probability to preferred responses, i.e., maximum likelihood estimation (MLE).

In the following sections, we show how to optimize this objective approximately. First, in §3.1, we show how MLE on preferred responses gives rise to a training objective for learning RMs that are better suited for downstream RL. Next, in §3.2, we show how methods from the test-time alignment literature (Faria & Smith, 2025) can be used to approximately sample from $\pi_\phi(y \mid x)$ and thereby estimate the likelihood gradient. We then show how contrastive divergence (Hinton, 2002) gives a practical truncated approximation for learning in this setting.

3.1 LEARNING THE ALIGNED DISTRIBUTION FROM PREFERRED DEMONSTRATIONS

Given a dataset of preferred demonstrations $\mathcal{D} = \{(y_+^{(k)}, x^{(k)})\}_{k=1}^M$, we want to learn the parameters ϕ of the RM $r_\phi(y, x)$, by maximizing the log likelihood of the aligned distribution $\pi_\phi(y \mid x)$ introduced in Eq. (3). Importantly, this objective is defined directly from preferred demonstrations alone, and therefore does not require the preference pairs assumed by BT style training. The gradient of the log likelihood objective with respect to ϕ takes the form:

$$\nabla_\phi \log \pi_\phi(y \mid x) = \tfrac{1}{\beta} \nabla_\phi r_\phi(y, x) - \tfrac{1}{\beta} \mathbb{E}_{y' \sim \pi_\phi(y' \mid x)} [\nabla_\phi r_\phi(y', x)]\,. \tag{4}$$

The positive term pushes the RM to increase the score of the observed demonstration (y_+, x), and the negative term subtracts the average gradient over responses drawn from the current aligned distribution $\pi_\phi(y \mid x)$. Intuitively, the RM is therefore trained to assign higher reward to preferred demonstrations than to responses that the current RM prioritizes. If the current RM assigns an excessive probability mass to undesirable samples, these samples will contribute more strongly to the negative term and will therefore be explicitly penalized.

Estimating the expectation in Eq. (4) using Monte Carlo requires sampling from $\pi_\phi(y \mid x)$, with respect to the current version of the RM. One possible approach would be to repeatedly take the current RM, train a policy to optimize the objective in Eq. 2, and then use that policy to form a Monte Carlo estimate of the expectation in Eq. 4. However, this would require running an expensive RL training loop for every RM gradient step.

We avoid an RL inner loop, instead turning to methods from the test-time alignment literature (Faria & Smith, 2025) to sample from the aligned distribution π_ϕ using only the base LM $p_{\mathrm{LM}}(y \mid x)$ and the RM $r_\phi(y, x)$.

3.2 GRADIENT ESTIMATION VIA TEST-TIME ALIGNMENT

To sample from the distribution $\pi_\phi(y \mid x)$, we will construct a Markov chain $(y^0, y^1, \ldots, y^T)$ that has $\pi_\phi(y \mid x)$ as its equilibrium distribution. The chain will start from the prompt and expert demonstration $(y_+, x) \sim \mathcal{D}$. On the tth iteration, it independently draws a new y from $p_{\mathrm{LM}}(y \mid x)$, and this hypothesis is accepted with an acceptance probability :

$$\alpha_\beta(y, y^t) = \min\left\{1, \exp\left(\tfrac{r(x,y) - r(x,y^t)}{\beta}\right)\right\}. \tag{5}$$

If the candidate y is accepted, the next state in the chain becomes $y^{t+1} = y$; if rejected, the chain stays at $y^{t+1} = y^t$. Repeating this process yields samples whose distribution approaches $\pi_\phi(y \mid x)$. This MCMC procedure has a natural connection to BoN sampling. In the limit $\beta \to 0$, the chain increasingly favors the highest-reward responses among the candidate set formed by the T independent samples from $p_{\text{LM}}(y \mid x)$ together with the initial response y_+. Although we do not use the algorithm in this hard-selection limit, we view it as a soft version of BoN sampling, we therefore name our method BONVOYAGE.

This construction can be viewed as the classical independent Metropolis (Metropolis et al., 1953) algorithm applied to the aligned distribution. It is an adaptation of QALIGN (Faria & Smith, 2025; Faria et al., 2024), which also targets $\pi_\phi(y \mid x)$ but uses a dependent proposal distribution based on resampling a random suffix using the base LM. Additionally, starting from the chain at the expert demonstration y_+ when building an approximation for Eq.equation 4 instantiates the contrastive divergence approximation (CD; Hinton, 2002). It has been shown to work well when the Markov chain is run for only a few steps, before reaching convergence. In our context, it will bias the estimation towards samples from the base LM that have higher or similar reward to y_+. In Appendix A, we provide empirical evidence supporting our test-time alignment method, particularly the use of CD to accelerate MCMC convergence.

While one could use more sophisticated and unbiased samplers like QALIGN, in practical use cases we want to reduce the number of steps. Our approach allows *pre-generating* a pool of samples from the base LM $p_{\text{LM}}(y \mid x)$ ahead of training the RM. Additionally, to improve estimates of the gradient, we will follow conventions when training language models and perform the gradient estimation across batches of prompts and then average before any gradient step.

To summarize the BONVOYAGE approach, to estimate one gradient step, we sample a batch of B preferred demonstrations $\{(y_+^{(k)}, x^{(k)})\}_{k=1}^{B} \subset \mathcal{D}$. We start a MCMC chain on each y_+ and run for some number of steps T to obtain the sample y'. We then take that sample and maximize the surrogate loss $\mathcal{L}(\phi, y_+, y', x) = r_\phi(y_+, x)/\beta - r_\phi(y', x)/\beta$. See Algorithm 1 for a more detailed description.

4 EXPERIMENTS

Our experiments are intended to answer three questions. First, do RMs trained with BONVOYAGE provide a better learning signal for RL than RMs trained with BT? We consider both verifiable and mostly non-verifiable domains, and evaluate the resulting RMs through the policies obtained by RL. Second, are these RMs also effective when used for inference scaling through BoN inference? Third, in the verifiable setting, are the RMs more robust to reward over-optimization during RL training?

The section is organized as follows. In §4.1, we describe the experimental setup. In §4.2, we present the main benchmark results, answering questions 1 and 2. In §4.3, we examine reward over-optimization in the verifiable setting (question 3). We include additional scaling experiments in § 4.4, discuss computational cost in § 4.5, and cover limitations and broader societal impact in Appendix I.

4.1 EXPERIMENTAL SETUP

Models and tasks We use BONVOYAGE to learn RMs targeting the base model QWEN3-1.7B-BASE (Yang et al., 2025). We focus on two domains: (i) *mathematics*, using verifiable data, and (ii) *science*, using mostly non-verifiable data. In the science domain, we first perform supervised finetuning (SFT) of QWEN3-1.7B-BASE on DOLCI-INSTRUCT-SFT (Olmo et al., 2025); more details on the SFT stage are provided in Appendix C. We then train an RM for the resulting model, which we denote QWEN3-1.7B-SCIENCE. All the RMs that we train are based on QWEN3-4B-BASE (Yang et al., 2025), to which we add a linear reward head on top of the hidden representation of the final token. For mathematics, we use 25k prompts drawn from DeepScaleR (Luo et al., 2025). For science, we use 20k prompts again from DOLCI-INSTRUCT-SFT, together with an additional 10k prompts from the SciQ (Johannes Welbl, 2017) training dataset, so that the RM is exposed to multiple-choice questions similar to those encountered at evaluation.

Preference collection Across both domains, we construct on-policy preference data using the same procedure. For each prompt x, we sample a set of 32 candidate responses from the target base model.

Preferred responses are then identified either using an automatic verifier, as in mathematics, or using a judge model (QWEN3-8B(Yang et al., 2025)) that compares each candidate against a reference response; the prompt provided for the judge model can be found in Appendix F. The resulting 32 candidates form the static sample pool used by BONVOYAGE, while the winning responses (y_+, x) are used as the preferred demonstrations.

Baselines Our primary baseline is an ON-POLICY BT RM trained on the same candidate pools as BONVOYAGE. For each prompt, we form BT preference pairs by randomly sampling one preferred response and one dispreferred response from the collected candidates. We additionally compare against a strong off-the-shelf OFF-POLICY RM, as measured by RewardBench (Lambert et al., 2024b) and RewardBench 2 (Malik et al., 2025), SKYWORK-REWARD-V2-QWEN3-4B (Liu et al., 2026), which is based on a QWEN3-4B backbone. This model is trained on roughly 20M offline preference pairs, which is 3 orders of magnitude more preference data, spanning multiple domains and base language models. Finally, in mathematics, we also include a Verifiable Rewards (VR) setting, in which the ground-truth correctness of a response is used directly as the reward signal.

Evaluation We evaluate RMs primarily through their downstream utility for RL. Concretely, we use each RM to optimize a target policy, starting from QWEN3-1.7B-BASE in mathematics and from QWEN3-1.7B-SCIENCE in science, and then measure the quality of the resulting policy on domain-specific benchmark suites.

In addition, we evaluate RMs for inference scaling using BoN sampling (Gao et al., 2022; Stiennon et al., 2020). In this setting, we repeatedly sample the base model n times, score each candidate response with the RM, and select the response with the highest score. We report results for powers of two up to $n = 512$. BoN has been shown in prior work to correlate with downstream RL performance (Frick et al., 2024), and is widely used in practice.

For mathematics, we report results on **MATH-500** (Hendrycks et al., 2021) and **GSM8K** (Cobbe et al., 2021). For science, we evaluate on **SciQ** (Johannes Welbl, 2017), **SciKnowEval** (Feng et al., 2024), **MMLU-Sci**, the science-related subsets of MMLU-Redux (Gema et al., 2025), namely `anatomy`, `astronomy`, `college_chemistry`, `college_physics`, `virology`, and `conceptual_physics`, as well as **ARC-Easy** and **ARC-C** (Clark et al., 2018).

RL training setup For RL, we adopt the DAPO (Yu et al., 2025) training recipe, implemented in VERL (Sheng et al., 2024) and adapted to the RM setting. Since different RMs produce outputs at different numerical scales, we normalize rewards using the mean and standard deviation computed from the first batch of RL training; additional details, including hyperparameter choices, are provided in Appendix D.

We perform RL on the same prompt dataset used to train the RMs. We exclude easy prompts for the target model, retaining only those with accuracy below 0.7. In all experiments, the policy is constrained to generate at most 2048 tokens, which is significantly lower than the typical reasoning model's sequence length. This limit is substantially shorter than the context lengths often used by reasoning models and may cap performance by restricting reasoning-chain length. We enforce it to reduce RL compute costs and keep responses within the length regime on which the RMs were trained. All runs are trained for 1200 RL steps. We save checkpoints every 200 steps and select the checkpoint with the highest validation accuracy. For mathematics, validation is performed on a held-out subset of DeepScaleR, while for science, we use the validation split of SciKnowEval.

RM training setup For RM training, we use DeepSpeed (Rasley et al., 2020) to implement the distributed training pipeline. We train with a batch size of 128 and use $T = 32$ steps of MCMC with the Metropolis algorithm with $\beta = 1.0$. Thus, each gradient step involves 4096 RM evaluations, from which we select 128 negative responses and form 128 training pairs for back-propagation. Additional details on optimization hyperparameters are provided in Appendix E.

4.2 BENCHMARK RESULTS

Reinforcement learning Table 1 and Table 2 summarize the main benchmark results in mathematics and science, respectively. Across both domains, BONVOYAGE consistently yields RMs that provide a more effective learning signal during RL, leading to final policies that outperform those obtained

Table 1: Math benchmark results (% accuracy, ↑).We compare QWEN3-1.7B-BASE optimized with three 4B RMs: ON-POLICY-BT, a BT RM trained on on-policy preferences, OFF-POLICY-BT, the off-policy RM SKYWORK-REWARD-V2-QWEN3-4B; and our RM trained with BONVOYAGE. We also report RLVR, which uses ground-truth correctness as the reward signal and therefore serves as a *reference upper bound* on what can be achieved with a perfectly faithful reward.

Model	MATH-500	GSM8K
QWEN3-1.7B-BASE	25.2 ± 1.0	41.9 ± 0.6
+ RL w/ ON-POLICY-BT	63.4 ± 1.8	82.2 ± 0.8
+ RL w/ OFF-POLICY-BT	63.0 ± 1.8	83.7 ± 0.8
+ RL w/ BONVOYAGE	$\mathbf{67.7} \pm 1.8$	$\mathbf{84.7} \pm 0.8$
QWEN3-1.7B-BASE + RLVR	67.9 ± 1.7	85.3 ± 0.8

Table 2: Science benchmarks results (% accuracy, ↑). Starting from QWEN3-1.7B-BASE, we first obtain QWEN3-1.7B-SCIENCE via SFT, and then evaluate further RL training with RMs learned using ON-POLICY-BT, OFF-POLICY-BT and BONVOYAGE.

Model	SciQ	SciKnowEval	ARC-Easy	ARC-C	MMLU-Sci
QWEN3-1.7B-BASE	28.6 ± 0.4	27.2 ± 0.4	29.0 ± 0.3	25.1 ± 0.4	20.0 ± 0.6
QWEN3-1.7B-SCIENCE	82.5 ± 0.9	67.3 ± 1.2	79.5 ± 0.6	65.7 ± 1.0	46.6 ± 1.5
+ RL w/ ON-POLICY-BT	91.1 ± 0.7	72.2 ± 1.2	91.5 ± 0.4	78.7 ± 0.9	57.4 ± 1.6
+ RL w/ OFF-POLICY-BT	90.4 ± 0.8	73.1 ± 1.2	90.0 ± 0.5	76.7 ± 1.0	57.4 ± 1.7
+ RL w/ BONVOYAGE	$\mathbf{92.3} \pm 0.7$	$\mathbf{74.1} \pm 1.2$	$\mathbf{93.1} \pm 0.4$	$\mathbf{81.3} \pm 0.9$	$\mathbf{59.3} \pm 1.7$

with the BT baselines. In mathematics, BONVOYAGE is also competitive with RL using verifiable rewards (RLVR): it trails RLVR by just 0.2 and 0.5 percentage points on MATH-500 and GSM8K, respectively, differences that lie within the reported error bars. These results are particularly striking given that RLVR uses a perfectly faithful reward signal, aligned by construction with the target utility, whereas both BONVOYAGE and BT rely on a learned proxy.

The science results show a similar pattern. SFT already yields a large gain over the base model, but RL with learned RMs provides a further substantial improvement across all benchmarks, and BONVOYAGE consistently achieves the best performance. This suggests that the SFT data still contains useful learning signal beyond the point at which validation loss stops improving with SFT.

Furthermore, we note that BONVOYAGE improves over ON-POLICY-BT on every benchmark by 0.8–2.6 points absolute. These results show that the advantage of BONVOYAGE is not limited to the verifiable mathematics setting, but extends to the more realistic non-verifiable data, where we cannot do RLVR.

Inference scaling The BoN scaling results in Figure 2 show that BONVOYAGE also yields the strongest RM for test-time scaling. In both domains, the RM learned with BONVOYAGE consistently outperforms the corresponding BT baselines as the number of samples increases. At the same time, the BT RMs remain relatively strong rerankers, especially OFF-POLICY-BT. We also observe that performance under ON-POLICY-BT begins to plateau once the number of samples exceeds roughly $n = 128$, while the RM learned with BONVOYAGE continues to improve throughout the full range of budgets we consider. This suggests that BoN's outputs remain relatively close to the base model's distribution $p_{\text{LM}}(y \mid x)$, so both BT RMs remain fairly reliable in this regime, without the stronger over-optimization effects observed during RL.

4.3 REWARD OVER-OPTIMIZATION RESULTS

Figure 1 shows training accuracy on the left and the correlation between reward and ground-truth correctness on the right. We do not plot the correlation for RLVR because, by construction, it is equal to 1 throughout training. Across the three RMs, we observe that the correlation between reward and ground-truth correctness decreases during RL training. We attribute these decreases in part to policy's distribution shift during training, as discussed in § 2. This degradation is sharper for

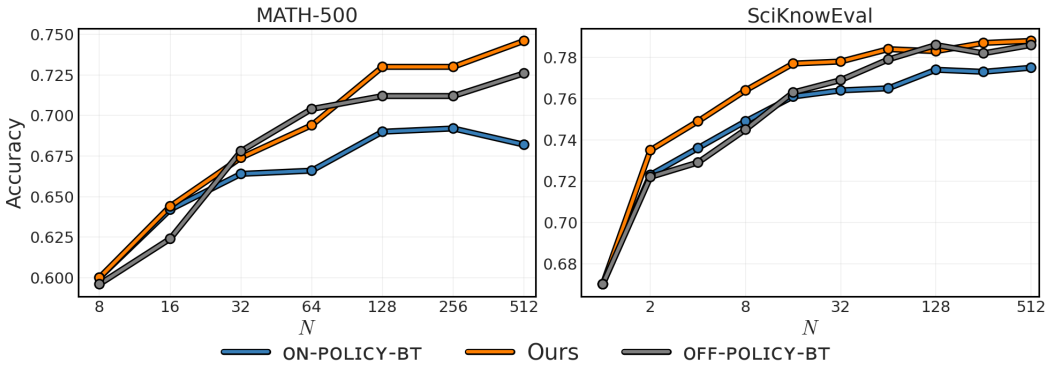


Figure 2: BoN scaling results. We evaluate RMs as rerankers for inference scaling by sampling N candidate responses from the base LM, scoring them with the RM, and selecting the highest-scoring response. All learned RMs are initialized from QWEN3-4B-BASE. **Left:** mathematics results on MATH-500, using QWEN3-1.7B-BASE as the base LM. We compare ●ON-POLICY-BT, an on-policy BT RM, ●OFF-POLICY-BT, the off-policy BT RM SKYWORK-REWARD-V2-QWEN3-4B, and ●OURS, an RM trained with BONVOYAGE. **Right:** science results on SCIKNOWEVAL, using QWEN3-1.7B-SCIENCE as the base LM. We compare ●ON-POLICY-BTand ●OURS.

ON-POLICY-BT in absolute terms. At the beginning of training, the correlation is approximately 0.60 for ON-POLICY-BT and OFF-POLICY-BT while BONVOYAGE is 0.68. By step 1000, these values decrease to roughly 0.33, 0.48, and 0.60, respectively. Despite this degradation, ON-POLICY-BT maintains a higher accuracy throughout training, and reaches a higher maximum accuracy than OFF-POLICY-BT. Furthermore, we observe clear signs of over-optimization on OFF-POLICY-BT, with a significant drop in training accuracy after step 200. By contrast, BONVOYAGE's training accuracy continues to go up until step 700 and then plateaus instead of displaying any significant evidence of over-optimization. Finally, RLVR achieves the highest final training accuracy, since its reward is perfectly aligned with correctness. However, these gains do not fully translate into benchmark performance; the best-performing checkpoint picked using validation accuracy was at step 800 just like BONVOYAGE.

Overall, Figure 1 provides empirical evidence for the mathematical motivation we offered in §3: as RL training progresses, the faithfulness of the RM signal can deteriorate even while the reward continues to increase. Because we selected the final checkpoint based on validation accuracy, the BT RMs' results were not impacted by this phenomenon. The BT RMs' checkpoints that achieved the best results were collected early into training at step 200, while BONVOYAGE and RLVR were at step 800. This indicates that BONVOYAGE, just like RLVR, is better able to leverage additional compute to obtain a superior performance.

4.4 SENSITIVITY TO RM SIZE

To study the effect of RM size, we retrain the RMs for both domains using QWEN3-1.7B-BASE as the RM backbone, and then repeat the same RL setup used in our main experiments. This allows us to compare how all methods benefit from increasing RM size. Figure 3 shows that increasing the RM size helps all RM learning methods across both benchmarks. In particular, the 1.7B version of BONVOYAGE **already outperforms** both 4B BT baselines. On the more challenging benchmark, MATH-500, BONVOYAGE gains a further 2.0 accuracy points when scaled to 4B. OFF-POLICY-BT exhibits a similar gain, which is consistent with it being trained on roughly 20M preference pairs, whereas ON-POLICY-BT improves by only 1.0 point.

4.5 COMPUTATIONAL COST

BONVOYAGE is more expensive to train than BT, since each gradient step requires many more forward passes through the RM. In our setting, this amounts to roughly 32× more forward passes. However, RM inference is dominated by the prefill stage and is therefore substantially more compute-

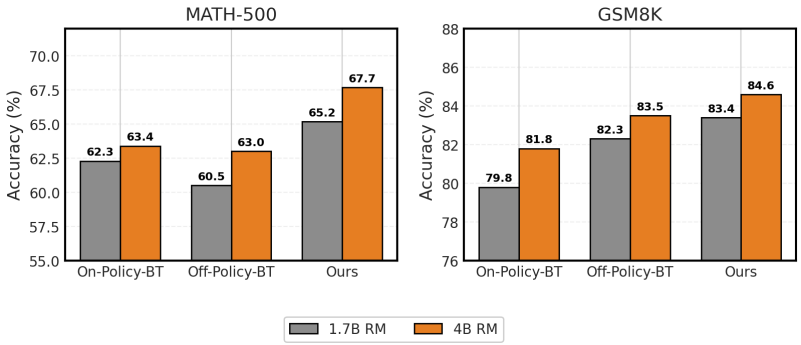


Figure 3: Mathematics benchmark results comparing RL training with RMs of different sizes. We compare ON-POLICY-BT, OFF-POLICY-BT, and BONVOYAGE using RMs based on ●QWEN3-4B-BASE and ●QWEN3-4B-BASE. **Left:** evaluation on MATH-500. **Right:** evaluation on GSM8K. Bars report benchmark accuracy (%), higher is better.

bound than autoregressive text generation. As a result, the increase in wall-clock time is much smaller than the increase in the number of forward passes. For example, on the science setup with 4 NVIDIA H200 GPUs, training the on-policy BT RM takes 62 minutes, whereas training the BONVOYAGE RM takes 293 minutes, corresponding to about a $4.7\times$ increase in wall-clock time.

However, note that BT can continue cycling through the data and forming fresh pairs, we could have trained it for longer (matching wall-clock time), but in practice we stopped when validation performance ceased to improve. More broadly, the off-the-shelf RM we compare against, SKYWORK-REWARD-V2-QWEN3-4B, was trained on roughly three orders of magnitude more data, a substantially larger training-compute budget than BONVOYAGE.

5 RELATED WORK

Reward modeling Several works revisit the standard BT objective for reward modeling (Dorka, 2024; Lou et al., 2025; Zhang et al., 2025b; Sun et al., 2025; Tang et al., 2024; Zhao et al., 2023). These works share our motivation of improving RMs for their downstream use in RLHF, but they do not explicitly model the role of the RM as defining the RLHF-optimal policy. They study reward modeling as a preference-learning problem, where the RM is trained to rank responses drawn from a fixed data distribution.

Large-scale RM training provides another route to improving downstream performance. Skywork-Reward-V2 (Liu et al., 2026) constructs a 20M-pair synthetic preference dataset through an human-AI synergistic two-stage pipeline and reports strong RM benchmark, BoN, and RLHF performance, while also finding that several BT-loss modifications do not consistently improve over standard BT for their data and GEMMA-2-27B variants. In this work we compare directly against the RMs they have released. Like our work, Hong et al. (2025) and Bukharin et al. (2025) address reward hacking directly. Hong et al. (2025) co-optimizes the RM and policy using verifiable rewards, training the RM while the policy is optimized against it. Bukharin et al. (2025) instead use RL against a target RM to source adversarial examples for improving the RM. Both approaches require RL training to learn the RM, making them significantly more expensive than our approach.

6 CONCLUSION

We proposed BONVOYAGE, a method to train RMs directly for their downstream role. Our empirical results show that BONVOYAGE consistently outperforms BT, including a strong off-the-shelf SKYWORK-REWARD-V2-QWEN3-4B baseline, yielding stronger final policies in both domains across MATH-500, GSM8K, SciQ, SciKnowEval, ARC-Challenge, and MMLU-Science. We also show that BONVOYAGE yields stronger rerankers under BoN sampling than BT-trained RMs. Furthermore, in mathematics, we directly measure reward over-optimization for both BT baselines and show that BONVOYAGE is substantially more robust to it, preserving alignment between RM scores and ground-truth correctness later into RL training.

AI USE STATEMENT

We used language models as part of our experimental pipeline to assist with data filtering and preference annotation. In §4.1, we used QWEN3-8B as a judge to identify the preferred responses in the science-domain training data by comparing model-generated candidates with ground truths.

REFERENCES

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D. Chang, and Prithviraj Ammanabrolu. Critique-out-loud reward models, 2024. URL https://arxiv.org/abs/2408.11791.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022a. URL https://arxiv.org/abs/2204.05862.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022b. URL https://arxiv.org/abs/2212.08073.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/2334029.

Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pp. 547–555, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450308137. doi: 10.1145/2020408.2020495. URL https://doi.org/10.1145/2020408.2020495.

Alexander Bukharin, Haifeng Qian, Shengyang Sun, Adithya Renduchintala, Soumye Singhal, Zhilin Wang, Oleksii Kuchaiev, Olivier Delalleau, and Tuo Zhao. Adversarial training of reward models, 2025. URL https://arxiv.org/abs/2504.06141.

Yanjun Chen, Dawei Zhu, Yirong Sun, Xinghao Chen, Wei Zhang, and Xiaoyu Shen. The accuracy paradox in rlhf: When better reward models don't yield better language models, 2024. URL https://arxiv.org/abs/2410.06554.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled ai feedback, 2024. URL https://arxiv.org/abs/2310.01377.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf, 2024. URL https://arxiv.org/abs/2405.07863.

Nicolai Dorka. Quantile regression for distributional reward models in rlhf, 2024. URL https://arxiv.org/abs/2409.10164.

Gonçalo Faria and Noah A. Smith. Sample, don't search: Rethinking test-time alignment for language models, 2025. URL https://arxiv.org/abs/2504.03790.

Gonçalo R. A. Faria, Sweta Agrawal, António Farinhas, Ricardo Rei, José G. C. de Souza, and André F. T. Martins. Quest: Quality-aware metropolis-hastings sampling for machine translation, 2024. URL https://arxiv.org/abs/2406.00049.

Kehua Feng, Keyan Ding, Weijie Wang, Xiang Zhuang, Zeyuan Wang, Ming Qin, Yu Zhao, Jianhua Yao, Qiang Zhang, and Huajun Chen. Sciknoweval: Evaluating multi-level scientific knowledge of large language models, 2024.

Patrick Fernandes, Aman Madaan, Emmy Liu, António Farinhas, Pedro Henrique Martins, Amanda Bertsch, José G. C. de Souza, Shuyan Zhou, Tongshuang Wu, Graham Neubig, and André F. T. Martins. Bridging the Gap: A Survey on Integrating (Human) Feedback for Natural Language Generation. *Transactions of the Association for Computational Linguistics*, 11:1643–1668, 12 2023. ISSN 2307-387X. doi: 10.1162/tacl_a_00626. URL https://doi.org/10.1162/tacl_a_00626.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios N. Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. How to evaluate reward models for rlhf, 2024. URL https://arxiv.org/abs/2410.14872.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization, 2022. URL https://arxiv.org/abs/2210.10760.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. Are we done with mmlu?, 2025. URL https://arxiv.org/abs/2406.04127.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization, 2023.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains, 2025. URL https://arxiv.org/abs/2507.17746.

Moritz Hardt, Nimrod Megiddo, Christos H. Papadimitriou, and Mary Wootters. Strategic classification. *CoRR*, abs/1506.06980, 2015. URL http://arxiv.org/abs/1506.06980.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970. URL https://api.semanticscholar.org/CorpusID:21204149.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018. URL https://doi.org/10.1162/089976602760128018.

Haitao Hong, Yuchen Yan, Xingyu Wu, Guiyang Hou, Wenqi Zhang, Weiming Lu, Yongliang Shen, and Jun Xiao. Cooper: Co-optimizing policy and reward models in reinforcement learning for large language models, 2025. URL https://arxiv.org/abs/2508.05613.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, 1982. ISSN 00278424. URL http://www.jstor.org/stable/12175.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J. Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment, 2025. URL https://arxiv.org/abs/2503.21878.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback, 2024. URL https://arxiv.org/abs/2406.09279.

Dulhan Jayalath, Shashwat Goel, Thomas Foster, Parag Jain, Suchin Gururangan, Cheng Zhang, Anirudh Goyal, and Alan Schelten. Compute as teacher: Turning inference compute into reference-free supervision, 2026. URL https://arxiv.org/abs/2509.14234.

Matt Gardner Johannes Welbl, Nelson F. Liu. Crowdsourcing multiple choice science questions. 2017.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think, 2025. URL https://arxiv.org/abs/2510.14901.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting, 2022a.

Tomasz Korbak, Ethan Perez, and Christopher L Buckley. Rl with kl penalties is better viewed as bayesian inference, 2022b.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2024a. URL https://arxiv.org/abs/2411.15124.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024b. URL https://arxiv.org/abs/2403.13787.

Yann LeCun, Sumit Chopra, Raia Hadsell, Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. 2006. URL https://api.semanticscholar.org/CorpusID:8531544.

Alexander K. Lew, Tan Zhi-Xuan, Gabriel Grand, and Vikash K. Mansinghka. Sequential monte carlo steering of large language models using probabilistic programs, 2023. URL https://arxiv.org/abs/2306.03081.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5315–5333, Toronto, Canada, jul 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.291. URL https://aclanthology.org/2023.acl-long.291/.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms, 2024a. URL https://arxiv.org/abs/2410.18451.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-reward-v2: Scaling preference data curation via human-ai synergy, 2026. URL https://arxiv.org/abs/2507.01352.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. Rm-bench: Benchmarking reward models of language models with subtlety and style, 2024b. URL https://arxiv.org/abs/2410.16184.

Team Llama. Llama 2: Open foundation and fine-tuned chat models, 2023.

Team Llama. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Xingzhou Lou, Dong Yan, Wei Shen, Yuzi Yan, Jian Xie, and Junge Zhang. Uncertainty-aware reward model: Teaching reward models to know what is unknown, 2025. URL https://arxiv.org/abs/2410.00847.

João Loula, Benjamin LeBrun, Li Du, Ben Lipkin, Clemente Pasti, Gabriel Grand, Tianyu Liu, Yahya Emara, Marjorie Freedman, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Alexander K. Lew, Tim Vieira, and Timothy J. O'Donnell. Syntactic and semantic control of large language models via sequential monte carlo, 2025. URL https://arxiv.org/abs/2504.13139.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2, 2025. Notion Blog.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. Rewardbench 2: Advancing reward model evaluation, 2025. URL https://arxiv.org/abs/2506.01937.

Nicholas C. Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, and A. H. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953. URL https://api.semanticscholar.org/CorpusID:1046577.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. CGMH: constrained sentence generation by metropolis-hastings sampling. *CoRR*, abs/1811.10996, 2018. URL http://arxiv.org/abs/1811.10996.

Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In Pat Langley (ed.), *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pp. 663–670. Morgan Kaufmann, 2000.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 745–750, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273590. URL https://doi.org/10.1145/1273496.1273590.

Du Phan, Matthew D. Hoffman, David Dohan, Sholto Douglas, Tuan Anh Le, Aaron Parisi, Pavel Sountsov, Charles Sutton, Sharad Vikram, and Rif A. Saurous. Training chain-of-thought via latent-variable inference, 2023. URL https://arxiv.org/abs/2312.02179.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 3505–3506, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3406703. URL https://doi.org/10.1145/3394486.3406703.

Noam Razin, Zixuan Wang, Hubert Strauss, Stanley Wei, Jason D. Lee, and Sanjeev Arora. What makes a reward model a good teacher? an optimization perspective, 2025. URL https://arxiv.org/abs/2503.15477.

MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Bing Liu, Yunzhong He, and Afra Feyza Akyürek. Online rubrics elicitation from pairwise comparisons, 2025. URL https://arxiv.org/abs/2510.07284.

Ronald Rosenfeld, Stanley F. Chen, and Xiaojin Zhu. Whole-sentence exponential language models: a vehicle for linguistic-statistical integration. *Computer Speech & Language*, 15(1): 55–73, 2001. ISSN 0885-2308. doi: https://doi.org/10.1006/csla.2000.0159. URL https://www.sciencedirect.com/science/article/pii/S0885230800901591.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G. Finlayson, David Sontag, Tyler Murray, Sewon Min, Pradeep Dasigi, Luca Soldaini, Faeze Brahman, Wen tau Yih, Tongshuang Wu, Luke Zettlemoyer, Yoon Kim, Hannaneh Hajishirzi, and Pang Wei Koh. Dr tulu: Reinforcement learning with evolving rubrics for deep research, 2025. URL https://arxiv.org/abs/2511.19399.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *NeurIPS*, 2020.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022.

Hao Sun, Yunyi Shen, and Jean-Francois Ton. Rethinking bradley-terry models in preference-based reward modeling: Foundations, theory, and alternatives, 2025. URL https://arxiv.org/abs/2411.04991.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment, 2024. URL https://arxiv.org/abs/2402.05749.

Xueru Wen, Jie Lou, Yaojie Lu, Hongyu Lin, Xing Yu, Xinyu Lu, Ben He, Xianpei Han, Debing Zhang, and Le Sun. Rethinking reward model evaluation: Are we barking up the wrong tree?, 2025. URL https://arxiv.org/abs/2410.05584.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning, 2025. URL https://arxiv.org/abs/2505.10320.

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, Keqin 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. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Junkai Zhang, Zihao Wang, Lin Gui, Swarnashree Mysore Sathyendra, Jaehwan Jeong, Victor Veitch, Wei Wang, Yunzhong He, Bing Liu, and Lifeng Jin. Chasing the tail: Effective rubric-based reward modeling for large language model post-training, 2026. URL https://arxiv.org/abs/2509.21500.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction, 2025a. URL https://arxiv.org/abs/2408.15240.

Maosen Zhang, Nan Jiang, Lei Li, and Yexiang Xue. Language generation via combinatorial constraint satisfaction: A tree search enhanced Monte-Carlo approach. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1286–1298, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.115. URL https://aclanthology.org/2020.findings-emnlp.115.

Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. Beyond bradley-terry models: A general preference model for language model alignment, 2025b. URL https://arxiv.org/abs/2410.02197.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. Slic-hf: Sequence likelihood calibration with human feedback, 2023. URL https://arxiv.org/abs/2305.10425.

Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. Rmb: Comprehensively benchmarking reward models in llm alignment, 2025. URL https://arxiv.org/abs/2410.09893.

Brian D. Ziebart, Andrew Maas, J. Andrew (drew) Bagnell, and Anind Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of 23rd AAAI Conference on Artificial Intelligence (AAAI '08)*, pp. 1433 – 1438, July 2008.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL https://arxiv.org/abs/1909.08593.

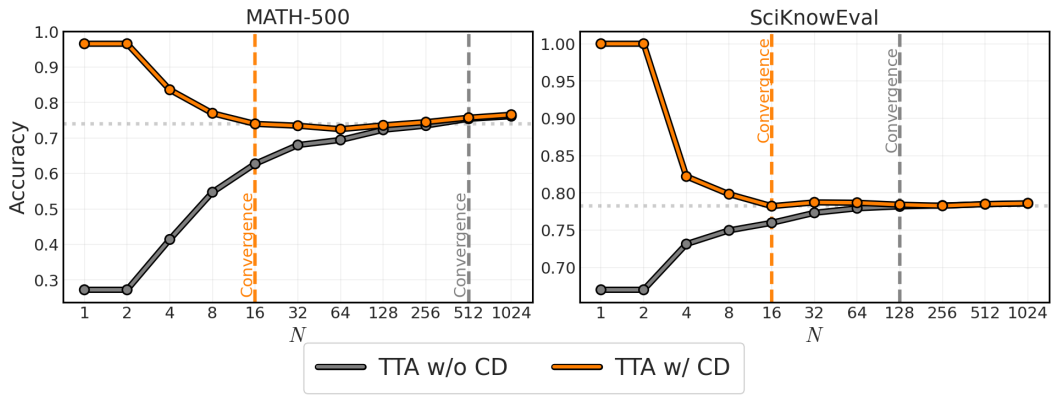


Figure 4: Test-time alignment (TTA) with RMs trained only using BoNVoyage. **Left:** MATH-500. **Right:** SciKnowEval (Qwen3-1.7B-Science). Curves show mean benchmark accuracy (%) versus number of MCMC steps T (log2-scale); higher is better. Dashed vertical lines indicate convergence point of each method.

A Test-time Alignment Experiments

In this section, we evaluate the effectiveness of BoNVoyage for estimating the gradient of the maximum-likelihood objective under the RLHF-optimal distribution in Eq. equation 3. Our gradient estimator relies on test-time alignment (TTA): using the base LM and RM, we approximately sample from this RLHF-optimal distribution. Thus, the quality of the gradient estimate depends directly on how well TTA approximates samples from this distribution.

We study how quickly our TTA procedure converges, with a particular focus on the role of contrastive divergence (CD) in accelerating convergence. Starting from the BoN candidate sets generated in § 4.2, we perform TTA with respect to RMs trained using BoNVoyage. We use our Metropolis-style variant of QALIGN as the underlying sampling procedure; see Algorithm 1. We compare two variants: TTA without CD and TTA with CD.

We define convergence as the point at which the benchmark accuracy of the TTA procedure reaches its terminating accuracy and does not change significantly with additional MCMC steps. Figure 4 shows the resulting accuracy as a function of the number of MCMC steps. Across both MATH-500 and SciKnowEval, CD substantially reduces the number of steps required for convergence. On MATH-500, TTA with CD converges at approximately $T = 16$ steps, while TTA without CD requires around $T = 512$ steps. Similarly, on SciKnowEval, CD converges at approximately $T = 16$ steps, compared with around $T = 128$ steps without CD. Note that these results come from an already strong RMs trained with BoNVoyage. Earlier in training, when the RM is closer to random initialization, the TTA chain may require more than $T = 16$ steps to obtain samples from the RLHF-optimal distribution.

These results show that CD makes TTA substantially more compute-efficient while reaching comparable final accuracy. Since our gradient estimator depends on samples produced by TTA, faster convergence also improves the practicality of using BoNVoyage for gradient estimation: it allows us to obtain useful approximate samples from the RLHF-optimal distribution.

B More Related Work

Reward modeling Several works revisit the standard BT objective for reward modeling (Dorka, 2024; Lou et al., 2025; Zhang et al., 2025b; Sun et al., 2025; Tang et al., 2024; Zhao et al., 2023). These works share our motivation of improving RMs for their downstream use in RLHF, but they do not explicitly model the role of the RM as defining the RLHF-optimal policy. They study reward modeling as a preference-learning problem, where the RM is trained to rank responses drawn from a fixed data distribution.

Large-scale RM training provides another route to improving downstream performance. Skywork-Reward-V2 (Liu et al., 2026) constructs a 20M pairs synthetic preference dataset through an human-AI synergistic two-stage pipeline and reports strong RM benchmark, BoN, and RLHF performance, while also finding that several BT-loss modifications do not consistently improve over standard BT for their data and GEMMA-2-27B variants. In this work we compare directly against the RMs they have released. Like our work, Hong et al. (2025) and Bukharin et al. (2025) address reward hacking directly. Hong et al. (2025) co-optimizes the RM and policy using verifiable rewards, training the RM while the policy is optimized against it. Bukharin et al. (2025) instead use RL against a target RM to source adversarial examples for improving the RM. Both approaches require RL training to learn the RM, making them significantly more expensive than our approach.

Generative RMs and Rubrics Work on reward modeling has moved beyond scalar RMs toward generative RMs, which produce textual critiques, reasoning, or explanations before assigning a final score (Zhang et al., 2025a; Ankner et al., 2024; Whitehouse et al., 2025). These methods change the form of the RM, but the underlying training problem typically remains a Bradley–Terry-style pairwise classification task over a fixed distribution of text. We therefore view generative RMs as orthogonal to our work. A separate line of work uses rubrics to enable reward signals in non-verifiable problems. Gunjal et al. (2025) propose using rubrics as rewards for RL, generating instance-wise rubrics from reference answers with frontier models. Other works explore online rubrics by contrasting pairwise or multiple model responses in closed-book settings (Rezaei et al., 2025; Jayalath et al., 2026). Shao et al. (2025) propose evolving rubrics, using responses from the learning policy together with additional information from web search to adapt the rubrics to better distinguish high-quality responses during training.

MCMC for language models Metropolis–Hastings (MH; Hastings, 1970) MCMC has been used with auto-regressive LMs to sample from text Gibbs distributions. Early work used it for constrained generation tasks (Miao et al., 2018; Zhang et al., 2020). More recently, Faria et al. (2024) uses it to sample high-quality diverse translations with a suffix-resampling proposal, which Faria & Smith (2025) later adopts for test-time alignment, and Karan & Du (2025) uses it to sample from the power distribution over reasoning traces. We use a modification of Faria & Smith (2025) during training for gradient estimation. However, for practical reasons, we use an independent proposal distribution.

Furthermore, TRICE (Phan et al., 2023) frames teaching an LM to reason before outputting a final answer as probabilistic inference under a latent-variable model. They propose a new method to solve this problem using an expectation-maximization algorithm that also employs MH with an independent proposal distribution.

Energy Based Models and Inverse RL The optimal policy induced by RLHF for a fixed RM, Eq. 3, can be viewed as a residual energy-based model (EBM) (Rosenfeld et al., 2001). Training EBMs is a long-standing problem (Hopfield, 1982; Hinton, 2002; LeCun et al., 2006), and its main obstacle is estimating the gradient of the log-partition function in Eq. 4. We directly learn the EBM formed by the optimum of the RLHF objective, and estimate its partition-function gradient with MH samples and CD (Hinton, 2002).

Furthermore, learning a reward signal from demonstrations sampled from an expert trajectory distribution is the central objective of inverse RL (IRL), including maximum-entropy IRL (Ng & Russell, 2000; Ziebart et al., 2008).

C IMPLEMENTATION DETAILS FOR SFT

In the science domain, we first perform supervised fine-tuning (SFT) of QWEN3-1.7B-BASE on DOLCI-INSTRUCT-SFT (Olmo et al., 2025). We use full-parameter fine-tuning with DeepSpeed ZeRO-2 and select the best checkpoint according to validation loss. The main SFT hyperparameters are listed in Table 3.

Table 3: SFT training hyperparameters.

Hyperparameter	Value
DeepSpeed configuration	ZeRO-2
Cutoff length	4096 tokens
batch size	32
Learning rate	2×10^{-5} for 1.7B; 1×10^{-5} for 4B
Learning rate schedule	Cosine
Warmup ratio	0.05
Training epochs	3
Precision	bfloat16
Checkpoint frequency	Every 230 steps
Best checkpoint selection	Validation loss

D Implementation Details for RL Training

For RL, we adopt the DAPO (Yu et al., 2025) training recipe, implemented in VERL (Sheng et al., 2024) and adapted to the RM setting. Since different RMs produce outputs at different numerical scales, we normalize rewards using the mean and standard deviation computed from the first batch of RL training. Table 4 lists the hyperparameters used in our experiments. We selected these settings primarily based on the RLVR baseline's validation accuracy at 500 steps in the math experiments, while also maximizing GPU memory utilization for Qwen-1.7B-Base.

Table 4: RL training hyper-parameters used for RLHF.

Hyperparameter	Value
Advantage normalization	Disabled
Algorith	GRPO
Policy loss aggregation	Token mean
KL coefficient β	10^{-4}
Learning rate	10^{-5}
Weight decay	10^{-2}
Warmup steps	10
Total steps	1200
Train batch size	128 prompts
PPO mini-batch size	32
Number of generations per prompt	8
Rollout temperature	1.0
Maximum prompt length	1024 tokens
Maximum response length	2048 tokens
Actor max tokens per GPU	36,864
Reference/rollout log-prob max tokens per GPU	221,184
Gradient clipping	1.0
PPO clipping range	$[0.20, 0.28]$
PPO clip ratio c	10.0
Entropy coefficient	0
Actor precision	bfloat16
Rollout precision	float16
Actor strategy	FSDP2

E Implementation Details for RM Training

For RM training, we use DeepSpeed (Rasley et al., 2020) to implement the distributed training pipeline. We train with a batch size of 128 and use $T = 32$ steps of MCMC with the Metropolis algorithm and $\beta = 1.0$. Thus, each gradient step involves 4096 RM evaluations, from which we select 128 negative responses and form 128 training pairs for back-propagation. We train for one epoch, save checkpoints every 50 steps, and select the checkpoint with the highest validation BoN@8 accuracy. Following Stiennon et al. (2022), we initialize the reward head weights from a normal distribution with

Science Generation Prompt

```
Take a look at the following question:
{question}

Answer with a response of your own, including the thinking process:
```

Figure 5: Prompt template used for question answering in the science experiments.

standard deviation $1/\sqrt{d_{\text{model}}+1}$. Additional details on optimization hyperparameters are provided in Table 5. We tuned the hyperparameters using validation BoN@8 accuracy and GPU utilization for the ON-POLICY-BT baseline. We then used the same hyperparameters for BONVOYAGE, after a small exploratory sweep indicated that these settings were effective for this setting as well. We selected $\beta = 1.0$ based on validation BoN@8 accuracy; however, we observed that changing β affected the tradeoff between BoN@16 and BoN@32 accuracy. This may have implications for RL, where the optimal choice of β may depend on the RL compute budget. We leave a more systematic exploration of this effect to future work.

Table 5: Reward model training hyperparameters.

Hyperparameter	**Value**
Total training epochs	1
Batch size	128
Learning rate	10^{-5}
LR Scheduler	cosine decay
Weight decay	10^{-4}
KL regularization weight β	1.0
Dropout rate	0.0
Precision	bfloat16
Optimizer ϵ	10^{-5}
Reward head initialization std.	$1/\sqrt{d_{\text{model}}+1}$
Maximum prompt length	2048 tokens
Maximum response length	2048 tokens
DeepSpeed configuration	ZeRO-3

F ON-POLICY DEMONSTRATIONS AND RESPONSE GENERATION

To construct on-policy preferred responses in non-verifiable domains, we first sample 32 responses with temperature 0.6 and maximum number of output tokens 2048, from the on-policy model for each dataset of prompts. The generation prompts for math and science are shown in Figures 6 and 5, respectively. We then compare each sampled response to the dataset reference response using a semantic-equivalence judge. We retain a sampled response as an on-policy demonstration when the judge returns `similar=true`. We use QWEN-8B as the judge; the full judge prompt is shown in Figure 7.

G DERIVATION OF THE RLHF-OPTIMAL POLICY GRADIENT

We derive the gradient of the RLHF-optimal policy in equation 3 with respect to the RM's parameters ϕ. This calculation shows that the gradient decomposes into a positive term on the observed response and a negative term given by an expectation under the induced optimal policy. This expectation is the quantity we estimate with the MCMC procedure in Algorithm 1.

Math Generation Prompt

```
Solve the following math problem step-by-step: {prompt}

Present the answer in LaTex format: \boxed{Your answer}
```

Figure 6: Prompt template for generating a step-by-step math solution with the final answer in boxed LaTeX format.

Prompt for Judging Semantic Equivalence

```
System:
You are an impartial judge. Your task: Given a question and two answers,
    determine: Are these two answers semantically similar/equivalent? Consider
    similar if they give the same factual content, reach the same conclusion with
     different reasoning, use different wording but equivalent meaning, or cover
    the same key points. Consider different if they contradict each other, give
    different conclusions, omit or add substantively different information, or
    address different aspects of the question.

Output JSON only: {"similar": true} or {"similar": false}. No other text.

User:
<question>{question}</question>

<answer1>{answer1}</answer1>

<answer2>{answer2}</answer2>

Provide your answer in a concise manner. No other text.
Output JSON only: {"similar": true} or {"similar": false}.
```

Figure 7: Prompt template for judging semantic equivalence between an on-policy response and the dataset reference response.

The gradient of the log likelihood objective with respect to ϕ takes the form :

$$\nabla_\phi \log \pi_\phi(y \mid x) = \tfrac{1}{\beta} \nabla_\phi r_\phi(y, x) - \nabla_\phi \log Z_\phi(x). \tag{6}$$

Expanding the second term, we obtain

$$\begin{aligned}
\nabla_\phi \log Z_\phi(x) &= \frac{1}{Z_\phi(x)} \nabla_\phi Z_\phi(x) && (7)\\
&= \frac{1}{Z_\phi(x)} \sum_{y'} \exp\Big(\tfrac{1}{\beta} r_\phi(y', x)\Big)\, \tfrac{1}{\beta} \nabla_\phi r_\phi(y', x)\; p_{\text{LM}}(y' \mid x) && (8)\\
&= \tfrac{1}{\beta} \sum_{y'} \pi_\phi(y' \mid x)\, \nabla_\phi r_\phi(y', x) && (9)\\
&= \tfrac{1}{\beta} \mathbb{E}_{y' \sim \pi_\phi(\cdot|x)} [\nabla_\phi r_\phi(y', x)]\,. && (10)
\end{aligned}$$

Substituting back yields

$$\nabla_\phi \log \pi_\phi(y \mid x) = \tfrac{1}{\beta} \nabla_\phi r_\phi(y, x) - \tfrac{1}{\beta} \mathbb{E}_{y' \sim \pi_\phi(\cdot|x)} [\nabla_\phi r_\phi(y', x)]\,. \tag{11}$$

H BoNVoyage Algorithm

Algorithm 1 summarizes the RM gradient estimation step used by BoNVoyage. For each preferred demonstration in a batch, we initialize a Markov chain at the preferred response, run T Metropolis steps with the independent sampled proposals, and use the final chain state as the negative response for RM training.

Algorithm 1 BoNVoyage: RM gradient estimation step

Require: dataset $\mathcal{D}$ of preferred demonstrations, RM r_ϕ, target KL regularization β, batch size B, chain length T

Sample $\{(y_+^{(k)}, x^{(k)})\}_{k=1}^{B} \subset \mathcal{D}$
for $k = 1, \ldots, B$ **do**
  Initialize $y^{(k)} \leftarrow y_+^{(k)}$
  for $t = 1, \ldots, T$ **do**
    Sample proposal $\tilde{y} \sim p_{\text{LM}}(\cdot \mid x^{(k)})$
    Accept $\tilde{y}$ with probability given by:

$$\alpha_\beta(\tilde{y}, y^{(k)}) = \min\left\{1,\ \exp\left(\frac{r_\phi(\tilde{y}, x^{(k)}) - r_\phi(y^{(k)}, x^{(k)})}{\beta}\right)\right\}$$

    If accepted, set $y^{(k)} \leftarrow \tilde{y}$
  end for
  Set $y'^{(k)} \leftarrow y^{(k)}$
end for
Maximize

$$\mathcal{L}(\phi) = \frac{1}{B} \sum_{k=1}^{B} \left[\frac{r_\phi(y_+^{(k)}, x^{(k)})}{\beta} - \frac{r_\phi(y'^{(k)}, x^{(k)})}{\beta}\right]$$

Take one gradient step with respect to ϕ

I Limitations & Broader Impact

BoNVoyage is more expensive to train than On-Policy-BT, since each gradient step requires substantially more forward passes through the RM. In our setting, this amounts to approximately $4.7\times$ more RM training compute. We nevertheless compare against a compute-matched On-Policy-BT baseline, using fresh preference pairs on the same prompts but with different responses. In our experiments, this additional compute did not improve validation accuracy for On-Policy-BT.

Our method also relies on contrastive divergence as a truncated approximation to the maximum-likelihood gradient. While we show that this approximation works well in practice, its effectiveness may depend on factors such as model size, dataset size, and the number of steps. Our experiments use a limited set of models and scales, including two Qwen-1.7B-Base policies and 1.7B and 4B RMs, all within a single model family.

Finally, our results show that BONVOYAGE can produce RMs that are more effective than BT-trained RMs for downstream RL training. Like other post-training techniques, this capability could be used to improve systems in beneficial applications, but it could also be misused to optimize models for harmful behavior. Responsible deployment should therefore include standard mitigation measures, such as careful data curation, evaluation for harmful capabilities, and monitoring of downstream model behavior.

Which neurofilament subunit polypeptide exhibits characteristic KSP repeats often targeted for posttranslational phosphorylation modifications?

A). NFM B). Peripherin C). NFH D). NFL

Provide your reasoning about the answer and finish your answer with the letter corresponding to the correct option (e.g., A, B, C, or D).

BT **Response 1** — ON-POLICY-BT — Checkpoint step: 720

Question Analysis:

The question asks about the neurofilament subunit that exhibits characteristic KSP repeats, often targeted for posttranslational phosphorylation modifications.

Thinking Process:

1. **Understand the Question:** - Focus on neurofilament subunits, their KSP repeats, and phosphorylation targets.
2. **Recall Neurofilament Components:** - Neurofilaments are composed of different subunits (NFM, NFH, NFL). - KSP repeats are known to be part of neurofilament subunits, particularly associated with phosphorylation targets.
3. **Identify the Key Features:** - KSP repeats are characteristic of neurofilament subunits. - These repeats are often targeted for posttranslational phosphorylation.
4. **Match the Subunit to the Features:** - The subunit associated with KSP repeats and targeted phosphorylation is **NFH (Neurofilament Heavy)**. - NFH contains the KSP repeats, making it a key target for posttranslational modifications like phosphorylation.
5. **Verify Against Options:** - The correct answer corresponds to **NFH (Option C)**. - Neurofilament heavy chain contains the KSP repeats and is commonly targeted for phosphorylation modifications.

Conclusion:

The neurofilament subunit with characteristic KSP repeats, often targeted for posttranslational phosphorylation modifications, is **NFH (Option C)**.

Answer: C). NFH

Key Points: - Neurofilament subunits like NFH contain KSP repeats. - These subunits are often subject to posttranslational modifications like phosphorylation. - NFH is specifically known for these features.

Note: Neurofilament light chain (NFM), heavy chain (NFH), and other components play different roles, with NFH being the correct answer here.

Final Answer: C). NFH

Answer Letter: C

This reasoning process ensures that the correct subunit, NFH, is selected based on its characteristics (KSP repeats) and targets (phosphorylation).

Note on Common Terminology: - KSP repeats are associated with neurofilament heavy chains (NFH), confirming the correctness of the answer.

Educational Takeaway:
Understanding the specific subunit (NFH) in neurofilaments that possess KSP repeats targeted for phosphorylation modifications is key to answering this question correctly.

Final Assessment:
The correct answer is **C). NFH**.

Answer Letter: C

This structured thinking process ensures a thorough understanding of the question and accurate selection of the correct answer.

BV **Response 2** — BONVOYAGE — Checkpoint step: 720

The question asks which neurofilament subunit polypeptide exhibits characteristic KSP repeats targeted for post-translational phosphorylation modifications.

Key Points:

1. **Neurofilament Subunits**: Neurofilaments are composed of multiple subunits. The question specifies which subunit has characteristic KSP repeats targeted for phosphorylation.
2. **KSP Repeats**: These are structural repeats known to be target sites for phosphorylation.
3. **Subunit Identification**: The question gives options for the subunit. The key here is to recall which neurofilament subunit is associated with KSP repeats and phosphorylation.

Analysis of the Options:

- **NFM (Neurofilament Mild Form)**:
  NFM is one of the subunits of neurofilaments. KSP repeats are known to be present in certain subunits.
- **Peripherin**: This term sounds more like a peripheral membrane protein, not directly related to neurofilament subunits and their KSP repeats.
- **NFH (Neurofilament Heavy)**:
  NFH is a major subunit of neurofilaments. KSP repeats are characteristic of NFH, which is targeted for post-translational modifications like phosphorylation.
- **NFL (Neurofilament Light)**:
  NFL is another subunit, often involved in the structural framework of neurofilaments. While it may have structural features, its KSP repeats are not as prominently associated with phosphorylation modifications as NFH, which is the subunit associated with KSP repeats and phosphorylation targets.

Conclusion:

The question specifies that the neurofilament subunit with characteristic KSP repeats targeted for phosphorylation modifications is **NFH**.

Neurofilament heavy chain (NFH) contains the KSP repeats, which are commonly targeted for post-translational modifications like phosphorylation. This makes **C) NFH** the correct answer.

Reasoning Process: 1. **Identify Key Concepts**: Understand that neurofilament subunits have structural features (like KSP repeats) targeted for phosphorylation. 2. **Recall Neurofilament Subunits**: Neurofilaments are composed of NFM, NFH, and NFL. The question specifies KSP repeats, which are linked to NFH. 3. **Link KSP Repeats to Subunit**: Since KSP repeats are characteristic of NFH and are targeted for phosphorylation, the correct answer must be NFH. 4. **Verify Other Options**: NFM, peripherin, and NFL are associated with different structural features, not the specific KSP repeats targeted for phosphorylation. 5. **Conclude**: Based on the knowledge that NFH contains KSP repeats and is targeted for phosphorylation, the correct answer is **C) NFH**.

Answer: C) NFH

\boxed{C}

Figure 8: Example validation response during RL training in the science domain, initialized from QWEN-1.7B-SCIENCE. Both policies are evaluated on the same question. The **left panel** (response 1) shows the response from the policy trained with the ON-POLICY-BT RM, while the **right panel** (response 2) shows the response from the policy trained with the RM from BONVOYAGE. Both policies are shown at RL checkpoint step 720.

J QUALITATIVE EXAMPLE

We also examine whether the trained models differ qualitatively beyond their benchmark performance. Figure 8 shows an example validation question from the science domain, along with responses from two policies at the same RL checkpoint step. Both responses arrive at the correct answer.

The response generated by the policy trained with the ON-POLICY-BT RM is noticeably repetitive, adding redundant statements and unnecessary restatements of the same conclusion. In contrast, the response from BONVOYAGE is also relatively long, but is more information-dense, less repetitive, and ends with a clearly boxed final answer.