GRAM-R²: Self-Training Generative Foundation Reward Models for Reward Reasoning

Chenglong Wang[1*], Yongyu Mu[1*†], Hang Zhou[1], Yifu Huo[1], Ziming Zhu[1], Jiali Zeng[2], Murun Yang[1], Bei Li[1], Xiaoyang Hao[1], Chunliang Zhang[1],Fandong Meng[2], Jingbo Zhu[1], Tong Xiao[1‡]

[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]Pattern Recognition Center, WeChat AI, Tencent Inc., China
clwang1119@gmail.com, xiaotong@mail.neu.edu.cn

Abstract

Major progress in reward modeling over recent years has been driven by a paradigm shift from task-specific designs to generalist reward models. Despite this trend, developing effective reward models remains a fundamental challenge: the heavy reliance on large-scale labeled preference data. Pre-training on abundant unlabeled data offers a promising direction, but existing approaches fall short in instilling explicit reasoning capabilities into reward models. To bridge this gap, we propose a self-training approach that can leverage unlabeled data to scale up reward reasoning in reward models. Based on this approach, we develop GRAM-R², a generative reward model trained to produce not only preference labels but also accompanying reward rationales. GRAM-R² can serve as a foundation model for reward reasoning and can be applied to a wide range of tasks with minimal or no additional fine-tuning. It can support downstream applications such as policy optimization and task-specific reward tuning. Experiments on response ranking, task adaptation, and reinforcement learning from human feedback demonstrate that GRAM-R² consistently delivers strong performance, outperforming several strong discriminative and generative baselines.

Code — https://github.com/NiuTrans/GRAM/tree/main/extensions/GRAM-RR

Models — https://huggingface.co/collections/wangclnlp/gram-rr

Extended version — https://arxiv.org/abs/2509.02492

Introduction

Reward models are a cornerstone of aligning large language models (LLMs) with human preferences during post-training. Typically, a reward model is trained to encode these preferences, and the LLM is subsequently fine-tuned to maximize the reward signal it provides. This paradigm is first exemplified by reinforcement learning from human feedback (RLHF) (Stiennon et al. 2020). More recently, the use of reward models has expanded beyond training into inference, where they are used to re-rank candidate responses. This approach has emerged as a strategy in studies on inference-time scaling laws (Wu et al. 2024; Li et al. 2025).

The dominant approach to developing reward models is to collect a dataset of training examples demonstrating correct behavior for desired human preferences in a specific task, train a model to imitate these behaviors, and then test its performance to align LLMs with independent and identically distributed examples. While this approach has proven successful for aligning LLMs in narrow contexts (Stiennon et al. 2020; Xu et al. 2024), its application is limited to these tasks. As the field progresses towards artificial general intelligence (AGI), a paradigm shift is necessary: moving towards generalist reward models that can generalize across a wide range of tasks to facilitate the broader alignment of AI systems with human preferences.

Labeling multi-task, large-scale preference data offers a strategy to enhance generalist performance (Cui et al. 2023; Wang et al. 2024c,b). However, from a multi-task learning perspective, each labeled example is drawn from a task-specific distribution, and current reward models typically require hundreds or thousands of labeled examples to learn functions that generalize well across tasks (Zhang and Yang 2021). This reliance on labeled data poses a significant bottleneck, making it challenging to scale reward model training to the level of LLM training.

A promising direction is to pre-train on unlabeled data before fine-tuning on a smaller labeled set. This two-stage paradigm first equips the model with implicit knowledge of human preferences from unlabeled data, such as input-response pairs, and then fine-tunes it using labeled data. Since the pre-training stage does not depend on large-scale labeled datasets, it is highly scalable. Under this paradigm, foundation reward models such as GRAM (Wang et al. 2025b) and POLAR (Dou et al. 2025) have emerged. However, while these foundation models effectively learn *what* humans prefer, they do not capture the explicit reasoning behind *why* those preferences are held during the pre-training process. This limitation prevents them from leveraging the strong reasoning capabilities inherent to the LLM backbone.

[*]Authors contributed equally.
[†]Work was done when Yongyu Mu was interning at Pattern Recognition Center, WeChat AI, Tencent Inc.
[‡]Corresponding author.

More importantly, another line of work has demonstrated that incorporating explicit reasoning (referred to as *reward reasoning*) into reward models can substantially improve model performance (Chen et al. 2025; Guo et al. 2025).

In this paper, we connect these two lines of work and extend the pre-training stage to incorporate reward reasoning explicitly. Our goal is to endow foundation reward models with the capability to perform reward reasoning across a wide range of downstream tasks, either without fine-tuning or with only minimal task-specific supervision. To train this model, we propose a self-training approach designed to elicit reward reasoning using labeled data that lacks rationales (referred to as *rationale-free labeled data*) and vast amounts of unlabeled data. This approach can circumvent the need for expensive rationale-based annotations, thus ensuring the scalability required for building foundation reward models. Specifically, we first train a preference-proving model conditioned on an input, a response pair, and a preference label, which generates a proof explaining why the labeled preference holds. For rationale-free labeled data, we use this preference-proving model to synthesize rationales for each example. For unlabeled data, we allow the reward model to enhance its reward reasoning capability through a self-training loop iteratively: 1) the reward model predicts preference labels for unlabeled data; 2) the preference-proving model generates corresponding rationales; and 3) the reward model is updated using the synthesized data. Notably, our self-training process allows the reward model to scale up its reward reasoning by leveraging vast unlabeled data.

We introduce the resulting model as a **G**enerative foundation **R**ew**A**rd **M**odel for **R**eward **R**easoning (GRAM-R^2). It can be directly applied to downstream tasks such as response ranking or further fine-tuned with a small amount of task-specific data. In our experiments, we evaluate GRAM-R^2 under three settings: response ranking, task adaptation, and RLHF. Across all test cases, GRAM-R^2 consistently exhibits a strong reward reasoning capability with little or no additional fine-tuning, and significantly outperforms both discriminative and generative baselines. For instance, when using LLaMA-3.1-8B-Instruct as the backbone, GRAM-R^2 achieves gains of 10.1 and 6.9 points in average accuracy on RM-Bench over vanilla discriminative and generative reward models, respectively. These results demonstrate that strong reasoning capabilities can be elicited from rationale-free labeled and unlabeled data.

Related Work

Reward Modeling. Reward models, typically trained on human preference data, are central to RLHF and other alignment strategies like DPO and rejection sampling (Lee, Auli, and Ranzato 2021; Rafailov et al. 2023; Chu et al. 2023; Wang et al. 2024a; Zhou et al. 2024; Wang et al. 2025c). Recent works on improving reward models could be classified into three groups. The first group focused on large-scale, high-quality training data, developing either task-specific datasets (Stiennon et al. 2020; Xu et al. 2024) or more general preference datasets (Cui et al. 2023). The second group explored stronger reward modeling approaches (Coste et al. 2024; Min et al. 2024). Notably, researchers have shown that integrating explicit reasoning into reward models is crucial for improving alignment performance (Chen et al. 2025; Guo et al. 2025). Although reward modeling through these approaches effectively captures human preferences, they often rely heavily on complex reinforcement learning and labeled data. To alleviate this, a third line of work has emerged that leverages unlabeled data to pre-train foundation reward models, such as GRAM (Wang et al. 2025b) and POLAR (Dou et al. 2025). However, these approaches overlook the development of reward reasoning capabilities, thereby limiting the model to exploit the reasoning potential of the LLM backbone fully.

Self-Training. Self-training (Scudder 1965; Han, Luo, and Wang 2019; Xie et al. 2020; Wang et al. 2021) is a classic semi-supervised learning framework. The basic idea is to employ model predictions on unlabeled data to generate pseudo-labels. These pseudo-labeled examples are then used to augment the original training set, enabling the model to improve its performance by leveraging large-scale unlabeled corpora without requiring additional human annotation. Such a guiding principle has shown empirical success in diverse domains such as computer vision (Yalniz et al. 2019; Zoph et al. 2020), natural language processing (Yeo et al. 2024; Zhang et al. 2024a; Luo et al. 2025), and lifelong learning (Lee et al. 2019). Here, we extend this idea to training reward models and show that self-training with large-scale unlabeled data can effectively scale up reward reasoning in reward models. To our knowledge, this is the first work to apply self-training to reward model training.

Preliminaries

Reward Model Training

In LLMs literature, a reward model is typically written as a function $r_\phi(x, y)$, where ϕ is the set of model parameters, x is the input, and y is the response. Throughout this work, an *input* can be an arbitrary token sequence fed into an LLM, such as "*What is the capital of France?*", and a *response* is the token sequence produced by the LLM as a result of that input, such as "*Paris*". To date, mainstream reward model architectures can be broadly categorized into two types: *discriminative* and *generative*.

Discriminative Reward Models. Discriminative reward models compute scores directly as scalar outputs from a classification architecture. Such a model typically consists of a Transformer decoder without a Softmax layer. The concatenated input–response $[x, y]$ is passed via a pre-trained LLM, and the final-layer hidden representations are used to compute a scalar score. This model can be trained through a Bradley-Terry loss function (Bradley and Terry 1952). While this loss function considers pairwise ranking between responses, the trained reward model is used as a scoring function that assigns a numerical reward $r_\phi(x, y)$ to each response y, along with its corresponding input x.

Generative Reward Models. While discriminative reward models have demonstrated success, this scoring approach fails to fully leverage the text generation capabilities that LLMs are fundamentally designed for (Zhang et al.

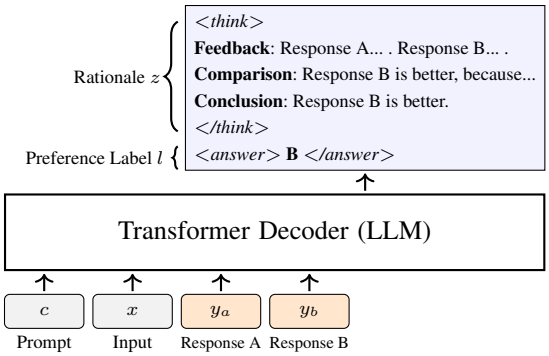


Figure 1: Architecture of the Generative Reward Model. The generative reward model utilizes a pre-trained LLM to predict a preference label from a given prompt directly. Optionally, it can incorporate reward reasoning before generating the final preference label prediction.

2024b). To address this limitation, recent studies have increasingly focused on developing generative reward models (Liang et al. 2025). These models produce reward signals via natural language generation. Specifically, they use an LLM to generate preference-related tokens, given a natural language prompt c and a tuple (x, y_a, y_b). The prompt describes the task in natural language, and the model predicts a label token w that aligns with the human preference l, where $l = \text{A}$ denotes preference for y_a, and $l = \text{B}$ indicates preference for y_b. The model can be trained by

$$\mathcal{L}_{\text{g}} = -\mathbb{E}_{(c,x,y_a,y_b,l)\sim D_r}\big[\log \pi_\phi(w = l|s)\big] \quad (1)$$

where s denotes the string $[c, x, y_a, y_b]$, and $\pi_\phi(\cdot)$ denotes the probability of token prediction by the LLM.

Recent studies have shown that framing reward prediction as a reasoning task can further leverage the powerful reasoning capabilities of LLMs to improve reward modeling performance (Chen et al. 2025; Guo et al. 2025). In these works, the model is trained to generate explicit reward reasoning (*e.g.*, analyzing and evaluating the responses individually) before predicting the final preference label as shown in Figure 1. Let z denote this rationale, a natural language justification for the preference label. The model first generates z conditioned on the input string s, and then predicts the preference label based on both the context and the generated rationale. In this process, it can be trained to generate both the rationale and the final label via the following objective:

$$\mathcal{L}_{\text{g}} = -\mathbb{E}_{(c,x,y_a,y_b,l,z)\sim D_p} \big[\log \pi_\phi(z|s) + \log \pi_\phi(w = l|s, z)\big] \quad (2)$$

where $\mathcal{D}_p$ is a set of annotated data containing both a preference label l and a corresponding labeled rationale z. Note that although incorporating reward reasoning significantly improves the performance of reward models, it presents a non-trivial challenge: it requires costly human annotations that include not only preference labels but also their corresponding detailed rationales.

Applying Reward Models

Three applications of foundation reward models can be considered in LLMs. A straightforward application is response ranking, where several responses are given, and we score and rank these responses. This approach is widely used in reranking settings, such as best-of-n sampling, where the highest-scoring response among n candidates is selected based on reward scores (Lee, Auli, and Ranzato 2021; Fernandes et al. 2022; Gao, Schulman, and Hilton 2023).

A second application of reward models is to provide learning signals for fine-tuning LLMs toward human preferences in RLHF, typically through algorithms such as PPO (Ouyang et al. 2022; Wang et al. 2022).

A third application is that when task-specific human preference data is available, the reward model can be further fine-tuned to better align with that particular task (Wang et al. 2025a; Dou et al. 2025). The adapted reward model can then be used in downstream applications such as RLHF or response ranking.

Our Method

In this section, we present a **G**enerative foundation **R**ew**A**rd **M**odel for **R**eward **R**easoning (GRAM-R^2). An overview of the GRAM-R^2 training process is shown in Figure 2. As illustrated, we first train a preference-proving model and then utilize it to perform iterative self-training to pre-train GRAM-R^2, enabling it to scale up its reward reasoning using vast rationale-free labeled data and unlabeled data.

Preference-Proving Model Training

While a considerable amount of labeled preference data exists, it often lacks the very rationales needed to train generative reward models in the art of reward reasoning. To unlock the full potential of this data, we propose a preference-proving model that can automatically generate textual proofs for the provided preference labels.

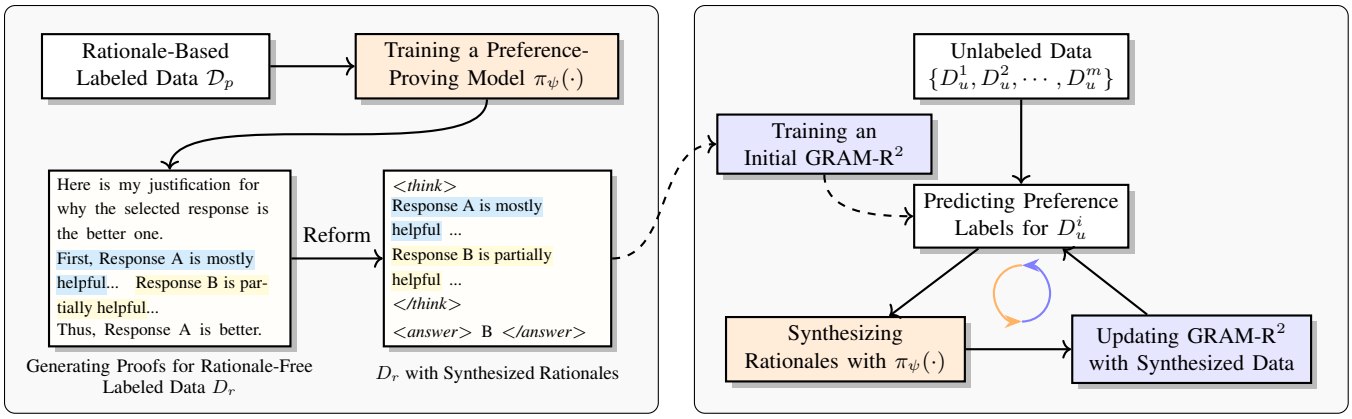
Task Definition. Given an example (s, l) from a rationale-free labeled dataset $\mathcal{D}_r$, the objective of the preference-proving model is to generate a textual proof $\hat{z}$ that justifies the preference label l. We define the preference-proving model as a conditional LLM:

$$\pi_\psi : (s, l) \mapsto \hat{z} \quad (3)$$

where ψ denotes the model parameters. To train the model, we minimize the negative log-likelihood of generating the ground-truth rationale:

$$\mathcal{L}_{\text{p}} = -\mathbb{E}_{(c,x,y_a,y_b,l,z)\sim \mathcal{D}_p}\left[\log \pi_\psi(\hat{z} \mid s, l)\right] \quad (4)$$

In our implementation, we design a reversible transformation rule that converts a rationale z into a structured, proof-like format $\hat{z}$ and vice versa. Notably, training the preference-proving model requires significantly less annotated data than training the reward model itself, as it only involves teaching the model to explain existing preference judgments rather than learning the preferences from scratch.



(a) Training with Rationale-Free Labeled Data (b) Training with Unlabeled Data

Figure 2: An overview of the self-training approach for GRAM-R^2. The process begins by training a preference-proving model on a small, rationale-based seed dataset of approximately 40.5K examples. This model is then used to synthesize rationales for a larger, rationale-free labeled dataset of 1M examples, which in turn is used to train the initial GRAM-R^2 model. Subsequently, GRAM-R^2 undergoes three iterations of self-training, using a new batch of 0.5M unlabeled examples in each iteration.

Preference Proof Selection. To enhance the quality and reliability of the generated proofs, we do not rely on a single output from the preference-proving model. Instead, for each input tuple (s, l), we employ a sampling-and-filtering strategy. First, we generate k candidate proofs $\{\hat{z}^1, \hat{z}^2, \cdots, \hat{z}^k\}$ by sampling from the model π_ψ with a non-zero temperature. We then re-rank the sampled proofs using a probabilistic scoring function. For each candidate $\hat{z}^i$, we compute

$$\text{Score}(s, l, \hat{z}^i) = -\frac{\log \pi_\psi(\hat{z}^i | s, l)}{\log \pi_\psi(\hat{z}^i)} \quad (5)$$

This scoring function produces values in the range $(-\infty, 0]$, with higher scores indicating higher-quality proofs. The basic intuition behind this design is to favour proofs that are highly *specific* to the given context (s, l). It accomplishes this by rewarding proofs that are probable given the context but improbable in isolation, thereby penalizing generic or templated statements that lack contextual relevance.

Self-Training with Unlabeled Data

The preference-proving model allows us to synthesize rationales for existing labeled data, thereby creating a dataset suitable for training reasoning reward models. However, the performance of this approach is ultimately constrained by the scarcity of the initial labeled preference data. To overcome this bottleneck and further enhance the model's reward reasoning capabilities, we introduce a self-training approach that leverages abundant unlabeled data.

Iterative Self-Training. Starting with an initial generative reward model trained on labeled data with synthesized rationales, we iteratively enhance it using batches of unlabeled data $\{D_u^1, D_u^2, \cdots, D_u^m\}$. In the i-th iteration, the model is first used to generate preference labels (*i.e.*, preference predictions) for the unlabeled data in batch D_u^i. These pseudo-labeled samples are then fed into the preference-proving model, which synthesizes corresponding rationales. The resulting rationale-based data is merged with the existing synthesized data, and the reward model is retrained on this combined set. This updated model is then used in the next iteration to further improve reward reasoning capabilities.

Preference Label Denoising. A principal challenge in self-training is the propagation of errors from noisy pseudo-labels, which can degrade model performance over successive iterations (Xie et al. 2020; Das and Sanghavi 2023). To mitigate this risk, we implement a multi-pronged denoising strategy that filters both unreliable preference labels and low-quality rationales. First, to enhance label stability, we aggregate predictions from multiple inference runs and apply a majority vote strategy. Second, we enforce a confidence threshold, discarding any pseudo-label whose softmax probability falls below a predefined value. Finally, we validate the rationales themselves through rule-based checks to remove malformed or irrelevant examples. Specifically, we discard examples that contain excessively long rationales, omit rationales to the predicted preference label, or fail to adhere to the structural constraints specified in the prompt.

It is worth noting that a key design choice in our self-training pipeline is the use of a dedicated preference-proving model to generate rationales, rather than relying on those produced internally by the reward model itself. This decision is motivated by the pursuit of high-quality, reliable proofs. While the reward model is trained to perform both reasoning and prediction, the preference-proving model specializes in a single task: generating compelling and coherent proofs.

Experiments

Experimental Setups

Model Backbones. For our main experiments, we initialized the preference-proving model with Qwen3-14B (Yang et al. 2025). For the GRAM-R^2 model itself, we developed and evaluated two separate versions based on the

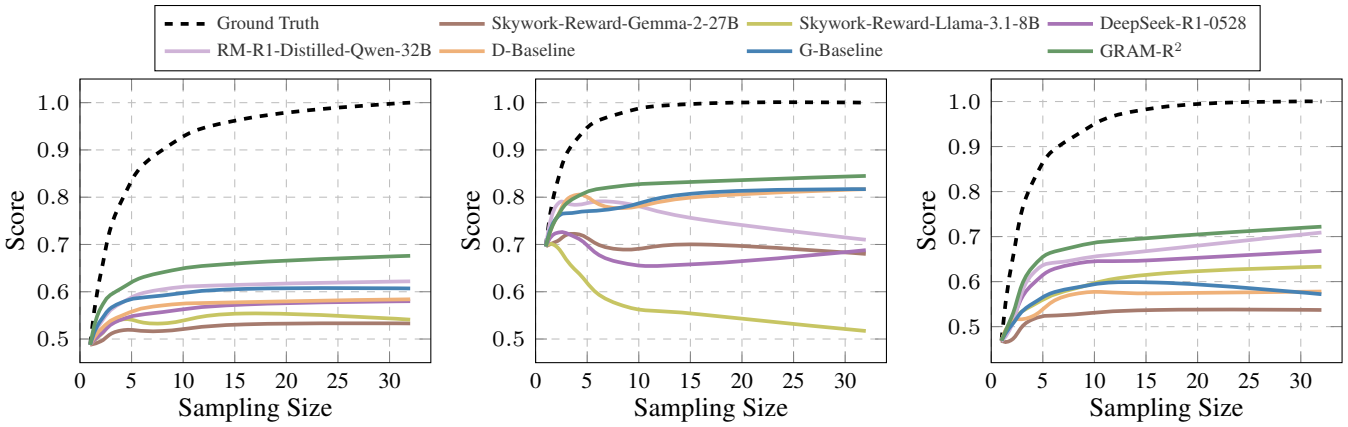
Model	Params.	RM-Bench			JudgeBench		
		Chat	Math	Overall	Knowl.	Reason.	Overall
LLM-as-a-Judge							
GPT-4o♯†	-	67.2	67.5	72.5	50.6	54.1	59.8
Claude-3.5-Sonnet♯†	-	62.5	62.6	61.0	62.3	66.3	64.8
DeepSeek-R1-0528†	671B	76.7	74.3	72.8	59.1	82.7	78.8
Open-Source Reward Models							
Llama-3.1-Nemotron-70B-Reward‡	70B	70.7	64.3	70.7	62.3	72.5	67.2
Skywork-Reward-Gemma-2-27B‡	27B	71.8	59.2	70.5	59.7	66.3	65.0
Skywork-Reward-Llama-3.1-8B‡	8B	69.5	60.6	70.1	59.1	64.3	62.5
Nemotron-Super‡	49B	73.7	91.4	82.7	71.4	73.5	77.2
Nemotron-Super-Multilingual‡	49B	77.2	91.9	84.2	64.9	74.5	75.2
Reasoning Reward Models							
RM-R1-Distilled-Qwen-32B	32B	74.2	91.8	83.9	76.0	80.6	78.8
RM-R1-Distilled-Qwen-14B	14B	71.8	90.5	81.5	68.1	72.4	78.1
RRM-32B	32B	66.6	81.4	73.1	79.9	70.4	75.7
Training with Unlabeled Preference Data							
GRAM-Qwen3-14B	14B	67.4	55.2	69.9	63.0	64.3	71.4
GRAM-Qwen3-8B	8B	63.5	53.9	68.3	62.3	64.3	67.8
Training on the Same Labeled Preference Data (LLaMA-3.1-8B-Instruct)							
Discriminative RM	8B	70.2	78.3	76.0	88.2	67.1	74.4
Generative RM	8B	74.8	81.1	79.2	90.8	69.4	76.9
GRAM-R^2 (Ours)	8B	76.0	89.8	85.7	90.9	83.7	81.0
+*voting@16*	8B	**76.3**	**90.4**	**86.1**	**91.2**	**84.3**	**81.6**
Training on the Same Labeled Preference Data (LLaMA-3.2-3B-Instruct)							
Discriminative RM	3B	70.5	70.6	75.6	86.0	70.9	73.4
Generative RM	3B	72.3	72.1	77.1	90.4	74.3	76.6
GRAM-R^2 (Ours)	3B	74.4	88.8	83.8	93.0	78.1	80.3
+*voting@16*	3B	**74.8**	**89.4**	**84.6**	**93.5**	**78.6**	**80.8**

Table 1: Accuracies (%) on RM-Bench and JudgeBench. The best result in each group is in **bold**. Results marked with ♯ on RM-Bench are from Chen et al. (2025), those with † on JudgeBench are from Liu et al. (2025), and those with ‡ for both RM-Bench and JudgeBench are from Wang et al. (2025d). The other baseline results are either reproduced from their original papers or obtained by evaluating their publicly available models or API access. We use a dotted line to distinguish between the discriminative and generative reward models. Results for the "Code" and "Safety" dimensions of RM-Bench and the "Math" and "Code" dimensions of JudgeBench can be found in our full arXiv version.

LLaMA-3.1-8B-Instruct and LLaMA-3.2-3B-Instruct models (Dubey et al. 2024).

Training Datasets. Our preference-proving model was trained on the HelpSteer3 dataset (Wang et al. 2025d), which comprises 40.5K labeled preference examples. Each example was enriched with human-written feedback and a comparative analysis, and we treated this combination as the rationale. For the initial training of GRAM-R^2, we curated a 1M-sample rationale-free dataset by amalgamating data from various open sources: MultiPref (Miranda et al. 2024), CodeUltraFeedback (Weyssow et al. 2024), Unified-Feedback, Prometheus2-Preference (Kim et al. 2024), PKU-SafeRLHF (Ji et al. 2023), and Skywork-Reward-Preference-80K-v0.2 (Liu et al. 2024a). The unlabeled data for self-training was sourced from the StackExchange dataset. To enhance the model's reasoning capabilities after pre-training, we performed fine-tuning on the human-annotated rationale-based HelpSteer3 dataset.

Baselines. Our primary baselines included strong open-source reward reasoning models, such as RRM (Guo et al. 2025) and RM-R1 (Chen et al. 2025). We also compared GRAM-R^2 with several strong baselines: *LLM-as-a-Judge*, where we prompted LLMs like GPT-4o and DeepSeek-V3 to generate preferences; *open-source reward models*, open-source discriminative and generative reward models, including Nemotron-Super-GenRM (Wang et al. 2025d), and others; and *training on the same labeled preference data*, denoting the standard reward models trained on discriminative and generative frameworks using our labeled preference data, respectively (denoted as Discriminative RM and Generative RM). Furthermore, we compared GRAM-R^2 with several approaches designed to utilize the unlabeled data to



(a) Average Score Across Benchmarks (b) MBPP Plus Best-of-n Score (c) MMLU-Pro Best-of-n Score

Figure 3: Best-of-n sampling performance curves for GRAM-R^2 and strong baseline models on the PPE benchmark. "D-Baseline" and "G-Baseline" refer to discriminative and generative reward models, respectively, trained on the same labeled preference data. "Ground Truth" represents an oracle reward model that selects responses based on gold-truth answers. All results are reported using the LLaMA-3.1-8B-Instruct backbone.

enhance reward models. These include GRAM, which pretrains a generative reward model on a response generation task (Wang et al. 2025b). Note that the POLAR model is excluded from this comparison (Dou et al. 2025), as it requires reference responses not available in these benchmarks.

Pair-wise Response Ranking

Task Setups. Pairwise response ranking is the most commonly used evaluation protocol for reward models. Given an input x^t and two candidate responses, y_a^t and y_b^t, the task is to predict the preferred response. Evaluation is conducted on a test set $D_{\text{pair}}^t = (x^t, y_a^t, y_b^t, l^t)$, where l^t denotes the ground-truth preference label. Model performance is measured by the accuracy of its predictions against these labels. For this task, we evaluate GRAM-R^2 on two widely adopted benchmarks: RM-Bench (Liu et al. 2024b), which assesses the model's ability to detect subtle stylistic preferences, and JudgeBench (Tan et al. 2024), which is designed to evaluate generative reward models across diverse tasks.

Results. We evaluated the reward reasoning capabilities of GRAM-R^2 using the pairwise response ranking task. Table 1 reports the performance of GRAM-R^2 and various baselines on RM-Bench and JudgeBench. Firstly, a key finding from the results is the consistent and substantial performance improvement brought by incorporating unlabeled data through self-training. Notably, across both backbone settings, GRAM-R^2 outperforms both discriminative and generative baselines trained on the same labeled dataset, demonstrating that reward reasoning capabilities can be effectively scaled using large-scale unlabeled data. Furthermore, compared to reasoning reward models that rely on expensive rationale-based annotations or complex reinforcement learning training, GRAM-R^2 achieves stronger reward reasoning performance through a simpler and more cost-effective approach, *i.e.*, only using supervised fine-tuning with rationale-free labeled data and unlabeled data. This highlights the practicality and scalability of our approach for training generalist reward models. Additionally, our approach enables the development of compact yet competitive reward models. For instance, our GRAM-R^2 model initialized with LLaMA-3.2-3B-Instruct achieves scores of 83.8% on RM-Bench and 80.3% on JudgeBench. This performance is remarkably on par with that of the much larger RM-R1-Distilled-Qwen-32B (which scores 83.9% and 78.8%, respectively), despite our model being over 10 times smaller.

List-wise Response Ranking

Task Setups. In practice, multiple candidate responses are typically generated for re-ranking. Given a list-wise test set $D_{\text{list}}^t = \{(x^t, y_1^t, y_2^t, \cdots, y_n^t)\}$, where n denotes the number of candidates, the task is to either rank the responses or identify the most preferred one based on human preferences. When the objective is to select the best response, a straightforward strategy involves performing a linear search using the generative reward model. More specifically, we initialize $y_b^t = y_1^t$ as the current best response and iteratively compare it with each remaining candidate. If y_b^t is found to be less preferred during any comparison, it is replaced with the superior response. This process continues until the most preferred response is identified. To improve computational efficiency and support parallelization, we also explore optimized selection algorithms, such as the divide-and-conquer approach. Similarly, this best-response search procedure can be extended to generate a full ranking by repeatedly selecting the best response from the remaining set. Here, to evaluate list-wise ranking performance, we adopt the PPE benchmark (Frick et al. 2024), which includes human preference data from verifiable correctness-based preferences from rigorous benchmarks such as MMLU-Pro and MATH. Specifically, we used the best-of-n (BoN) sampling from PPE to evaluate the ranking quality of our GRAM-R^2 model.

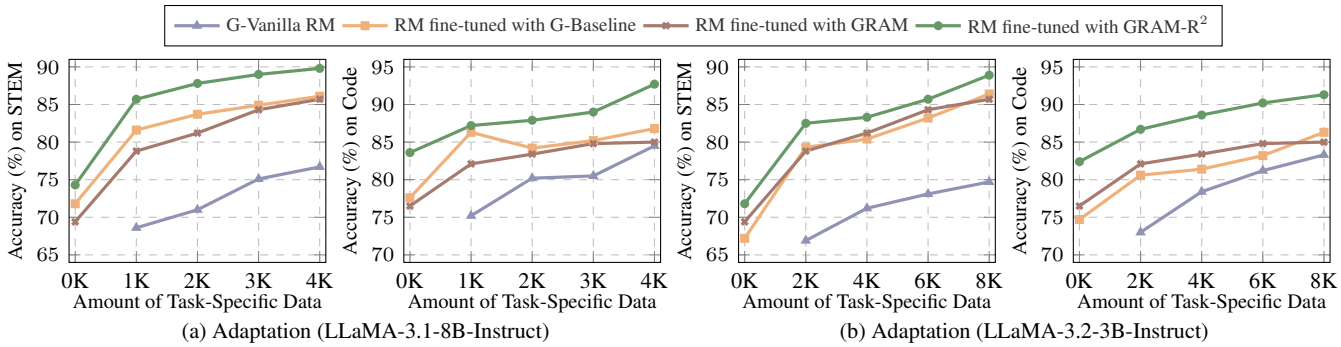


Figure 4: The performance of reward models fine-tuned with varying amounts of task-specific data (STEM and code generation).

Results of Best-of-n Sampling. Figure 3 presents the BoN sampling performance of GRAM-R^2 compared to several strong baselines. A key observation is the prevalence of reward overoptimization (Gao, Schulman, and Hilton 2023), particularly on the MBPP benchmark, where models such as Skywork-Reward-LLaMA-3.1-8B experience significant performance degradation as the number of samples increases. This degradation is primarily due to the limited generalization capabilities of these models to task-specific distributions. In contrast, GRAM-R^2 exhibits strong robustness against overoptimization and generalizes effectively across diverse tasks, owing to the incorporation of reward reasoning and self-training on large-scale data.

Reward Model Adaptation

Task Setups. We randomly sampled STEM and Code task data of varying sizes from the HelpSteer3, using subsets of {1K, 2K, 3K, 4K} for STEM and {2K, 4K, 6K, 8K} for Code. These subsets are used to fine-tune both GRAM-R^2 and its baselines (Generative RM and GRAM-Qwen3-14B). We also trained a generative reward model directly on each dataset as a baseline (*G-Vanilla RM*). All reward models were evaluated on the corresponding held-out validation sets provided by HelpSteer3 for each task.

Results. Figure 4 shows the accuracy of reward models fine-tuned on varying amounts of STEM and code data. We see that GRAM-R^2 fine-tunes more effectively into high-quality reward reasoning models compared to training a reward model directly from an LLM backbone. Notably, with 1K STEM samples, GRAM-R^2 achieves a task-specific accuracy that exceeds G-Vanilla RM by 17.1 points. GRAM-R^2 also outperforms all baselines across various data scales, showing its effectiveness as a foundation reward model.

Analysis

Scaling Training Data for Improved Performance. We explore the impact of training data size on the pre-training performance of GRAM-R^2. Specifically, we pre-train GRAM-R^2 using datasets of varying sizes: {0.5M, 1M, 1.5M, 2M, 2.5M}, each constructed by combining different amounts of rationale-free labeled data and unlabeled data. The model's performance is evaluated on RM-Bench, as shown in Figure 5. The results show that increasing the

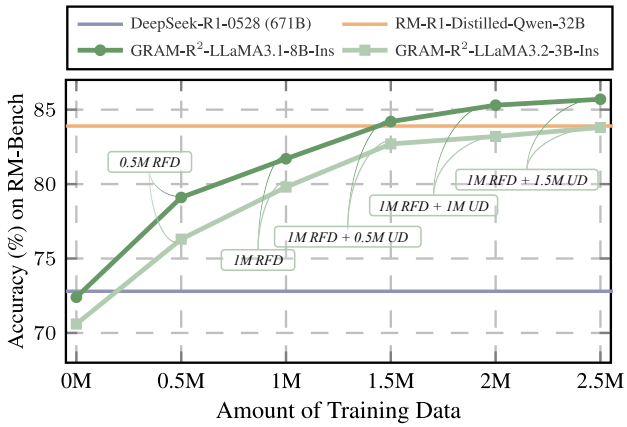


Figure 5: Performance scaling with different amounts of training data used to pre-train GRAM-R^2. "0M" denotes the setting where GRAM-R^2 is trained solely during the fine-tuning stage, without any pre-training on rationale-free labeled data or unlabeled data. *RFD*: Rationale-Free Labeled Data; *UD*: Unlabeled Data.

amount of training data generally improves the accuracy of GRAM-R^2, with the most notable gains observed when scaling from 0M to 1.5M examples. These findings highlight the importance of both unlabeled data and data scale, suggesting that using both rationale-free labeled data and unlabeled data can substantially enhance the reward reasoning capabilities in reward models.

Conclusions

We have explored training approaches for reward models with advanced capabilities in reward reasoning. We have developed a generative reward model, called GRAM-R^2. The model undergoes initial training on labeled data with synthetic rationales, and then further improves through self-training on large-scale unlabeled data to enhance its reward reasoning capabilities. Extensive experiments demonstrate that GRAM-R^2 consistently outperforms various baselines, yielding superior performance in reward reasoning.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Nos. U24A20334 and 62276056), the Yunnan Fundamental Research Projects (No.202401BC070021), the Yunnan Science and Technology Major Project (No. 202502AD080014), the Fundamental Research Funds for the Central Universities (Nos. N25BSS054 and N25BSS094), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009). We would like to thank the anonymous reviewers and SPC for their valuable comments and suggestions that helped improve this paper.

References

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*.

Chen, X.; Li, G.; Wang, Z.; Jin, B.; Qian, C.; Wang, Y.; Wang, H.; Zhang, Y.; Zhang, D.; Zhang, T.; et al. 2025. Rm-r1: Reward modeling as reasoning. *ArXiv preprint*.

Chu, Y.; Xu, J.; Zhou, X.; Yang, Q.; Zhang, S.; Yan, Z.; Zhou, C.; and Zhou, J. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *ArXiv preprint*.

Coste, T.; Anwar, U.; Kirk, R.; and Krueger, D. 2024. Reward Model Ensembles Help Mitigate Overoptimization. In *Proc. of ICLR*.

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

Das, R.; and Sanghavi, S. 2023. Understanding Self-Distillation in the Presence of Label Noise. In *Proc. of ICML*, 7102–7140.

Dou, S.; Liu, S.; Yang, Y.; Zou, Y.; Zhou, Y.; Xing, S.; Huang, C.; Ge, Q.; Song, D.; Lv, H.; et al. 2025. Pre-Trained Policy Discriminators are General Reward Models. *ArXiv preprint*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv e-prints*, arXiv–2407.

Fernandes, P.; Farinhas, A.; Rei, R.; C. de Souza, J. G.; Ogayo, P.; Neubig, G.; and Martins, A. 2022. Quality-Aware Decoding for Neural Machine Translation. In *Proc. of NAACL*, 1396–1412.

Frick, E.; Li, T.; Chen, C.; Chiang, W.-L.; Angelopoulos, A. N.; Jiao, J.; Zhu, B.; Gonzalez, J. E.; and Stoica, I. 2024. How to evaluate reward models for rlhf. *ArXiv preprint*.

Gao, L.; Schulman, J.; and Hilton, J. 2023. Scaling Laws for Reward Model Overoptimization. In *Proc. of ICML*, 10835–10866.

Guo, J.; Chi, Z.; Dong, L.; Dong, Q.; Wu, X.; Huang, S.; and Wei, F. 2025. Reward reasoning model. *ArXiv preprint*.

Han, J.; Luo, P.; and Wang, X. 2019. Deep Self-Learning From Noisy Labels. In *Proc. of ICCV*, 5137–5146.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Chen, B.; Sun, R.; Wang, Y.; and Yang, Y. 2023. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. In *Proc. of NeurIPS*.

Kim, S.; Suk, J.; Longpre, S.; Lin, B. Y.; Shin, J.; Welleck, S.; Neubig, G.; Lee, M.; Lee, K.; and Seo, M. 2024. Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models. arXiv:2405.01535.

Lee, A.; Auli, M.; and Ranzato, M. 2021. Discriminative Reranking for Neural Machine Translation. In *Proc. of ACL*, 7250–7264.

Lee, K.; Lee, K.; Shin, J.; and Lee, H. 2019. Overcoming Catastrophic Forgetting With Unlabeled Data in the Wild. In *Proc. of ICCV*, 312–321.

Li, Z.-Z.; Zhang, D.; Zhang, M.-L.; Zhang, J.; Liu, Z.; Yao, Y.; Xu, H.; Zheng, J.; Wang, P.-J.; Chen, X.; et al. 2025. From system 1 to system 2: A survey of reasoning large language models. *ArXiv preprint*.

Liang, X.; Zhang, H.; Li, J.; Chen, K.; Zhu, Q.; and Zhang, M. 2025. Generative Reward Modeling via Synthetic Criteria Preference Learning. In *Proc. of ACL*, 26755–26769.

Liu, C. Y.; Zeng, L.; Liu, J.; Yan, R.; He, J.; Wang, C.; Yan, S.; Liu, Y.; and Zhou, Y. 2024a. Skywork-Reward: Bag of Tricks for Reward Modeling in LLMs. *ArXiv preprint*.

Liu, C. Y.; Zeng, L.; Xiao, Y.; He, J.; Liu, J.; Wang, C.; Yan, R.; Shen, W.; Zhang, F.; Xu, J.; et al. 2025. Skywork-Reward-V2: Scaling Preference Data Curation via Human-AI Synergy. *ArXiv preprint*.

Liu, Y.; Yao, Z.; Min, R.; Cao, Y.; Hou, L.; and Li, J. 2024b. Rm-bench: Benchmarking reward models of language models with subtlety and style. *ArXiv preprint*.

Luo, N.; Gema, A. P.; He, X.; Van Krieken, E.; Lesci, P.; and Minervini, P. 2025. Self-Training Large Language Models for Tool-Use Without Demonstrations. *ArXiv preprint*.

Min, D. J.; Perez-Rosas, V.; Resnicow, K.; and Mihalcea, R. 2024. Dynamic Reward Adjustment in Multi-Reward Reinforcement Learning for Counselor Reflection Generation. In *Proc. of COLING*, 5437–5449.

Miranda, L. J. V.; Wang, Y.; Elazar, Y.; Kumar, S.; Pyatkin, V.; Brahman, F.; Smith, N. A.; Hajishirzi, H.; and Dasigi, P. 2024. Hybrid Preferences: Learning to Route Instances for Human vs. AI Feedback. *ArXiv preprint*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In *Proc. of NeurIPS*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Proc. of NeurIPS*.

Scudder, H. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 363–371.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D. M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. In *Proc. of NeurIPS*.

Tan, S.; Zhuang, S.; Montgomery, K.; Tang, W. Y.; Cuadron, A.; Wang, C.; Popa, R. A.; and Stoica, I. 2024. JudgeBench: A Benchmark for Evaluating LLM-Based Judges.

Wang, B.; Lin, R.; Lu, K.; Yu, L.; Zhang, Z.; Huang, F.; Zheng, C.; Dang, K.; Fan, Y.; Ren, X.; et al. 2025a. WorldPM: Scaling Human Preference Modeling. *ArXiv preprint*.

Wang, C.; Gan, Y.; Huo, Y.; Mu, Y.; He, Q.; Yang, M.; Li, B.; Xiao, T.; Zhang, C.; Liu, T.; et al. 2025b. GRAM: A Generative Foundation Reward Model for Reward Generalization. *ArXiv preprint*.

Wang, C.; Lu, Y.; Mu, Y.; Hu, Y.; Xiao, T.; and Zhu, J. 2022. Improved Knowledge Distillation for Pre-trained Language Models via Knowledge Selection. In *Proc. of EMNLP Findings*, 6232–6244.

Wang, Q.; Ding, K.; Gao, H.; Wang, H.; and Xu, R. 2025c. Error Comparison Optimization for Large Language Models on Aspect-Based Sentiment Analysis. In *Proc. of ACL*, 18630–18646.

Wang, Q.; Ding, K.; Luo, X.; and Xu, R. 2024a. Improving In-Context Learning via Sequentially Selection and Preference Alignment for Few-Shot Aspect-Based Sentiment Analysis. In *Proc. of SIGIR*, 2462–2466.

Wang, Q.; Wen, Z.; Zhao, Q.; Yang, M.; and Xu, R. 2021. Progressive Self-Training with Discriminator for Aspect Term Extraction. In *Proc. of EMNLP*, 257–268.

Wang, Z.; Dong, Y.; Delalleau, O.; Zeng, J.; Shen, G.; Egert, D.; Zhang, J.; Sreedhar, M. N.; and Kuchaiev, O. 2024b. HelpSteer 2: Open-source dataset for training top-performing reward models. In *Proc. of NeurIPS*.

Wang, Z.; Dong, Y.; Zeng, J.; Adams, V.; Sreedhar, M. N.; Egert, D.; Delalleau, O.; Scowcroft, J.; Kant, N.; Swope, A.; and Kuchaiev, O. 2024c. HelpSteer: Multi-attribute Helpfulness Dataset for SteerLM. In *Proc. of NAACL*, 3371–3384.

Wang, Z.; Zeng, J.; Delalleau, O.; Shin, H.-C.; Soares, F.; Bukharin, A.; Evans, E.; Dong, Y.; and Kuchaiev, O. 2025d. HelpSteer3-Preference: Open Human-Annotated Preference Data across Diverse Tasks and Languages.

Weyssow, M.; Kamanda, A.; Zhou, X.; and Sahraoui, H. 2024. Codeultrafeedback: An llm-as-a-judge dataset for aligning large language models to coding preferences. *ArXiv preprint*.

Wu, Y.; Sun, Z.; Li, S.; Welleck, S.; and Yang, Y. 2024. An empirical analysis of compute-optimal inference for problem-solving with language models.

Xie, Q.; Luong, M.; Hovy, E. H.; and Le, Q. V. 2020. Self-Training With Noisy Student Improves ImageNet Classification. In *Proc. of CVPR*, 10684–10695.

Xu, H.; Sharaf, A.; Chen, Y.; Tan, W.; Shen, L.; Durme, B. V.; Murray, K.; and Kim, Y. J. 2024. Contrastive Preference Optimization: Pushing the Boundaries of LLM Performance in Machine Translation. In *Proc. of ICML*.

Yalniz, I. Z.; Jégou, H.; Chen, K.; Paluri, M.; and Mahajan, D. 2019. Billion-scale semi-supervised learning for image classification. *ArXiv preprint*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *ArXiv preprint*.

Yeo, W. J.; Ferdinan, T.; Kazienko, P.; Satapathy, R.; and Cambria, E. 2024. Self-training large language models through knowledge detection. *ArXiv preprint*.

Zhang, D.; Zhoubian, S.; Hu, Z.; Yue, Y.; Dong, Y.; and Tang, J. 2024a. ReST-MCTS*: LLM Self-Training via Process Reward Guided Tree Search. In *Proc. of NeurIPS*.

Zhang, L.; Hosseini, A.; Bansal, H.; Kazemi, M.; Kumar, A.; and Agarwal, R. 2024b. Generative verifiers: Reward modeling as next-token prediction. *ArXiv preprint*.

Zhang, Y.; and Yang, Q. 2021. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 5586–5609.

Zhou, H.; Wang, C.; Hu, Y.; Xiao, T.; Zhang, C.; and Zhu, J. 2024. Prior constraints-based reward model training for aligning large language models. In *Proc. of CCL*, 555–570.

Zoph, B.; Ghiasi, G.; Lin, T.; Cui, Y.; Liu, H.; Cubuk, E. D.; and Le, Q. 2020. Rethinking Pre-training and Self-training. In *Proc. of NeurIPS*.