Semi-Supervised Reward Modeling via Iterative Self-Training

Yifei He*[1], Haoxiang Wang*[1], Ziyan Jiang[2], Alexandros Papangelis[2], Han Zhao[1,2]
[1]University of Illinois Urbana-Champaign [2]Amazon
{yifeihe3,hwang264,hanzhao}@illinois.edu
{ziyjiang,papangea}@amazon.com

Abstract

Reward models (RM) capture the values and preferences of humans and play a central role in Reinforcement Learning with Human Feedback (RLHF) to align pretrained large language models (LLMs). Traditionally, training these models relies on extensive human-annotated preference data, which poses significant challenges in terms of scalability and cost. To overcome these limitations, we propose Semi-Supervised Reward Modeling (SSRM), an approach that enhances RM training using unlabeled data. Given an unlabeled dataset, SSRM involves three key iterative steps: pseudo-labeling unlabeled examples, selecting high-confidence examples through a confidence threshold, and supervised finetuning on the refined dataset. Across extensive experiments on various model configurations, we demonstrate that SSRM significantly improves reward models without incurring additional labeling costs. Notably, SSRM can achieve performance comparable to models trained entirely on labeled data of equivalent volumes. Overall, SSRM substantially reduces the dependency on large volumes of human-annotated data, thereby decreasing the overall cost and time involved in training effective reward models.[1]

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Bai et al., 2022b) is central to the advancement of large language models (LLMs), including GPT-4 (Achiam et al., 2023) and Claude (Bai et al., 2022a). In RLHF, a reward model is trained to capture common human values and preferences by learning from pairwise preference data. Subsequently, language models are refined based on the reward signal to ensure that their actions are aligned with human preferences. This enables language models to perform complex tasks with intricate and challenging objectives, including mathematical reasoning (Wei et al., 2022; Lewkowycz et al., 2022), code generation (Chen et al., 2021; Li et al., 2022), summarization (Ouyang et al., 2022), among others.

While the resulting aligned language models produced by RLHF receive considerable attention, the importance of reward models is often overlooked. The accuracy of reward models in capturing human preferences is crucial for the effectiveness of RLHF. Moreover, beyond the application in RLHF, the ability to annotate preferences for prompt-response data enables reward models to serve as a valuable tool for a broader range of alignment approaches, including Rejection Sampling Finetuning (RSF) (Dong et al., 2023; Gulcehre et al., 2023; Yuan et al., 2024b) and Direct Preference Optimization (DPO) (Rafailov et al., 2024).

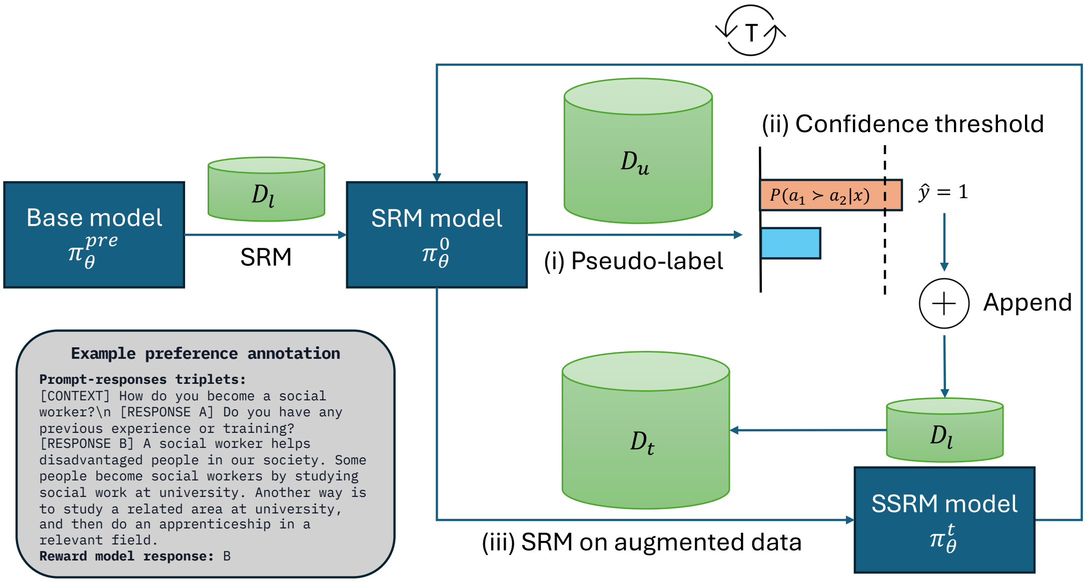
However, training reward models necessitates a significant volume of human-annotated preference data. Furthermore, in real-world applications, it is typical to encounter examples that deviate from the predefined training domains, and it is impractical to gather annotated preference data for every conceivable domain, presenting a significant barrier to the deployment of reward models. This challenge underscores the need to explore methods for enhancing reward models without relying extensively on large datasets of human-annotated preferences.

To enhance the efficiency of data utilization in reward model training, we propose Semi-Supervised Reward Modeling (SSRM), which efficiently utilizes unlabelled data to improve reward models. Our methodology stems from seminal works in the field of boosting (Schapire, 1990; Freund, 1995) and semi-supervised learning (Seeger, 2000; Grandvalet and Bengio, 2004), which aims at converting weak models to strong models with minimal requirement of labeled data. Given a pre-

*Equal contribution.

[1] Our code is available at https://github.com/RLHFlow/RLHF-Reward-Modeling/tree/main/pair-pm.

Findings of the Association for Computational Linguistics: EACL 2024, pages 7365–7377
November 12-16, 2024 ©2024 Association for Computational Linguistics



Example preference annotation

Prompt-responses triplets:
[CONTEXT] How do you become a social worker?\n [RESPONSE A] Do you have any previous experience or training? [RESPONSE B] A social worker helps disadvantaged people in our society. Some people become social workers by studying social work at university. Another way is to study a related area at university, and then do an apprenticeship in a relevant field.
Reward model response: B

Figure 1: Semi-Supervised Reward Modeling (SSRM) enhances the ability of a language model to predict preferences using both labeled and unlabeled data. Given a pretrained model π_θ^{pre}, a small labeled dataset D_l and a large unlabeled dataset D_u, we first perform supervised reward modeling (SRM) on D_l to obtain the SRM model π_θ^0. Then, at each step t, we perform three steps: (i) Pseudo-labeling: assign pseudo-labels to examples in D_u. (ii) Confidence thresholding: given a prompt x and two responses a_1, a_2, if the prediction confidence exceeds a preset threshold, append it to the labeled dataset to obtain D_t. (iii) SRM on augmented data: finetune the model on D_t.

trained language model and an unlabeled dataset with prompt-response pairs, SSRM iteratively executes the following steps: (i) Pseudo-labeling: assign pseudo-labels to the unlabeled examples based on their predicted preferences, (ii) Confidence thresholding: employ confidence threshold to selectively retain examples where the model exhibits high certainty in its predictions, (iii) Supervised reward-modeling (SRM): finetune on the filtered subset of data to enhance the reward model.

Through extensive experiments on various language models with parameter counts ranging from 0.4B to 8B, we demonstrate that SSRM effectively improves the performance of reward models without additional labeling costs. Notably, our findings reveal that reward models trained using SSRM exhibit performance closely approaching models trained entirely through traditional supervised methods on equivalent volumes of data. This underscores the efficacy of SSRM in utilizing unlabeled data to mirror the learning gains typically achieved only with labeled datasets.

Algorithm 1 Semi-Supervised Reward Modeling

```
Input: a pretrained model $\pi_\theta^{\text{pre}}$, a labeled dataset $D_l = \{(x^{(i)}, a_1^{(i)}, a_2^{(i)}, y^{(i)})\}_{i=1}^m$, an unlabeled dataset $D_u = \{(x^{(i)}, a_1^{(i)}, a_2^{(i)})\}_{i=1}^n$, confidence threshold $s$, iteration number $T$
Output: improved reward model $\pi_\theta^T$
// Initial SRM
$\pi_\theta^0 \leftarrow \text{Update}(\pi_\theta^{\text{pre}}, \ell_{\text{SRM}}, D_l)$
for $t = 1, 2, \cdots, T-1$ do
    Initialize $D_t = D_l$
    for $i = 1, 2 \cdots, n$ do
        // Pseudo label
        $\hat{y}^{(i)} = \text{argmax}_y \, \pi_\theta^t \left(y | x^{(i)}, a_1^{(i)}, a_2^{(i)}\right)$
        // Confidence threshold
        if $\pi_\theta^t \left(\hat{y}^{(i)} | \mathbb{T}\left(x^{(i)}, a_1^{(i)}, a_2^{(i)}\right)\right) \geq s$ then
            $D_t \leftarrow D_t \cup \{(x^{(i)}, a_1^{(i)}, a_2^{(i)}, \hat{y}^{(i)})\}$
        end if
    end for
    $\pi_\theta^{t+1} \leftarrow \text{Update}(\pi_\theta^t, \ell_{\text{SRM}}, D_t)$
end for
```

2 Semi-Supervised Reward Modeling

Semi-supervised reward modeling (SSRM) utilizes a small amount of labeled data and a large amount of unlabeled data to efficiently enhance the capabilities of reward models. In Section 2.1, we introduce reward modeling and our training recipe. In Section 2.2, we detail the self-training procedure in the context of reward modeling.

2.1 Reward Model

Reward modeling aims to encode human values by predicting the preference of a pair of responses given the same prompt. For reward models, two prominent types are widely recognized: Bradley-Terry models (Bradley and Terry, 1952) and preference models. Preference models have the flexibility to provide a more generalizable approach for capturing preferences as compared to Bradley-Terry models (Munos et al., 2023). Furthermore, they can be trained in a more computationally efficient manner. Consequently, we focus on preference models and employ them in our framework.

A preference model takes a prompt x and two responses a_1, a_2 as inputs and predicts the preference score $P(a_1 \succ a_2|x)$, indicating the preference of response a_1 over response a_2 given x. In the implementation, we adopt the methodology described by Zhao et al. (2023), which casts preference modeling as an instruction following task by leveraging the capabilities of LLMs for next-token prediction. Each preference pair takes the form (x, a_1, a_2, y), where $y \in \{A, B\}$ denotes whether the first or the second response is more preferable. The instruction template $\mathbb{T}(x, a_1, a_2)$ is formatted as `[CONTEXT]{x}[RESPONSE A]{`a_1`}[RESPONSE B]{`a_2`}`, and the target is the index for the preferred response (example shown in Figure 1). To mitigate the positional bias, i.e., the tendency for the ordering of responses to influence preference, we randomize their order during data preparation. To differentiate from supervised finetuning (SFT), which refers to finetuning for general instruction following, we term supervised finetuning on preference data as supervised reward modeling (SRM). At the end, we use SRM to train the reward model

$$\ell_{\text{SRM}}(\pi_\theta) = -\mathbb{E}_{(x,a_1,a_2,y)}[\log \pi_\theta(y|\mathbb{T}(x, a_1, a_2))].$$

During inference, we directly use the probability of decoding the correct label, i.e., $\pi_\theta(y|\mathbb{T}(x, a_1, a_2))$, as the preference score.

2.2 Iterative Self-Training

Training reward models requires pairwise data annotated with preference, consuming significant human efforts and resources. On the other hand, unlabeled data is easily accessible as language models can generate diverse responses given prompts. Therefore, to reduce the labeling cost for preference learning, we propose to utilize self-training (Grandvalet and Bengio, 2004; Lee et al., 2013). Specifically, we leverage confident predictions of a model to produce pseudo-labels for the unlabeled data and train on this augmented dataset iteratively. In the context of reward modeling, a labeled dataset takes the form $D_l = \{(x^{(i)}, a_1^{(i)}, a_2^{(i)}, y^{(i)})\}_{i=1}^m$, and an unlabeled dataset takes the form $D_u = \{(x^{(i)}, a_1^{(i)}, a_2^{(i)})\}_{i=1}^n$. Typically, the volume of unlabeled data vastly exceeds that of labeled data ($n \gg m$), providing a rich resource for augmenting the training dataset through self-training. The detailed steps of applying self-training in reward modeling are as follows.

Supervised training Initially, we train a reward model on the labeled dataset D_l using SRM mentioned in the previous section

$$\pi_\theta^0 = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{m} \log \pi_\theta \left(y^{(i)} | \mathbb{T} \left(x^{(i)}, a_1^{(i)}, a_2^{(i)} \right) \right).$$

This supervised model serves as a starting point of the self-training pipeline. The subsequent steps update upon this supervised model iteratively.

Pseudo-labeling At each iteration t, we assign pseudo-labels to unlabeled data in D_u based on the model predictions. For each data point in D_u, we select the response with the higher preference score as the pseudo-label

$$\hat{y}^{(i)} = \underset{y}{\operatorname{argmax}}\, \pi_\theta^t \left(y | x^{(i)}, a_1^{(i)}, a_2^{(i)} \right).$$

Here, we employ hard labeling: we pseudo-label data points in a binary way, instead of a probabilistic label based on the output logits.

Confidence thresholding After pseudo-labeling, it is crucial not to directly use the entirety of pseudo-labeled data for self-training, as doing so will result in a final model with identical performance as the initial model (Chapelle et al., 2006). Thus, we only select those data where the model exhibits high confidence. In the context of reward modeling, we compute the confidence based on the preference score of the assigned pseudo-label

$$\max_{y} \pi_\theta^t \left(y | \mathbb{T} \left(x^{(i)}, a_1^{(i)}, a_2^{(i)} \right) \right).$$

For a preset confidence threshold s, we only retain the pseudo-labeled data with confidence above the threshold, which are then combined with the labeled data to form the new training set for the current iteration, denoted as D_t.

Model update Following the dataset construction, we perform another round of SRM on D_t to get the updated model.

These steps are repeated for a preset number of iterations, with the entire procedure detailed in Algorithm 1. This approach efficiently leverages unlabeled data, reducing reliance on expensive labeled datasets and iteratively enhancing the model's performance in predicting human preferences.

3 Experiments

Our experiments are designed to evaluate the scalability and efficiency of SSRM across a spectrum of model sizes and configurations. We use a confidence threshold of 0.8, and more implementation details can be found in Appendix A.

3.1 Setup

Models We utilize three models to ensure a comprehensive assessment: PairRM (Jiang et al., 2023), Gemma-2B-it (Team et al., 2024) and Llama3-8b-it (Meta, 2024), with 0.4B, 2B and 8B parameters respectively. Note that PairRM is an encoder-based model specifically designed for reward modeling, so we follow the training methodology in Jiang et al. (2023) instead of training it through the mentioned SRM approach in Section 2.1, which is only applicable for language models with generation capabilities.

Datasets To fit the specific formatting requirements of the models, we use two datasets for training. For experiments involving Gemma-2B and Llama3-8B, we follow Dong et al. (2024) to use a mixture of 8 open-source datasets for reward model training: HH-RLHF (Bai et al., 2022a), SHP (Ethayarajh et al., 2022), HelpSteer (Wang et al., 2023), PKU-SafeRLHF (Ji et al., 2024), UltraFeedback (Cui et al., 2023), UltraInteract (Yuan et al., 2024a), Distilabel-Capybara (Daniele and Suphavadeeprasit, 2023) and Distilabel-Orca (Lian et al., 2023). Collectively, these datasets provide a rich and diverse corpus of 700K prompt-response triplets. Initially, these datasets come with ground-truth labels, but in our semi-supervised learning setup, we utilize only a tiny portion of the labels for the initial stages of supervised training (SRM) as described in Section 2.2. The remainder of the data, which constitutes the majority, is used unlabeled.

For experiments involving PairRM, we use the OpenHermesPreferences dataset (Huang et al., 2024), which contains 990k prompt-response triplets. This dataset is selected due to its compatibility with the specific data formatting requirements of PairRM, ensuring optimal training and performance evaluation conditions. We use the dataset in a purely unsupervised manner.

Model	# Data for SRM
Gemma-2B-it	175K
Llama3-8B-it	43.75K
PairRM	0

Table 1: Number of labeled data used for SRM.

Data Splitting We detail our approach to partitioning the datasets into labeled and unlabeled segments for training (summarized in Table 1). Both Gemma-2B-it and Llama3-8B-it are general-purpose language models that have not been tuned on preference datasets, so we first perform supervised training on a small labeled subset of the preference dataset. Specifically, we train Gemma-2B-it on one-fourth of the dataset, and train Llama3-8B-it on one-sixteenth of the dataset. This distinction in the volume of data used for initial training reflects their inherent differences in model capabilities and processing capacity. For instance, training Llama3-8B-it with a larger share of the dataset (such as one-fourth) tends to lead to an oversaturation of its learning capacity, thereby leaving minimal room for potential gains from subsequent exposure to unlabeled data (more detailed discussion in Appendix A). This SRM process with limited labeled data aims to equip the models with a basic understanding of preference learning, ensuring that the pseudo-labeling conducted in the initial iteration of SSRM has reasonable accuracy.

Conversely, PairRM operates differently. As an encoder-based model explicitly developed for reward modeling, PairRM is already equipped with advanced capabilities for preference prediction. Therefore, we exempt it from additional supervised training on preference data, skipping the SRM step described in Section 2.1. Instead, PairRM is employed directly in our SSRM setup for pseudo-labeling, leveraging its innate abilities to process and evaluate preference data effectively.

Evaluation The efficacy of our reward models is assessed using RewardBench (Lambert et al., 2024), a comprehensive evaluation benchmark for reward modeling. This benchmark consists of 2,985 prompt-chosen-rejected triplets, which encompass a range of critical evaluation criteria such

	Chat	Chat Hard	Safety	Reasoning	Average	# Data (Pseudo-labeled portion)
Gemma-2B-it	52.51	**50.00**	42.36	48.50	48.34	0 (0)
Partial SRM	90.78	35.96	31.38	51.61	52.44	175K (0)
SSRM [t=1]	**95.25**	37.06	47.70	41.37	55.34	310K (43.5%)
SSRM [t=2]	94.41	37.06	49.39	**68.10**	62.24	406.3K (56.9%)
SSRM [t=3]	94.41	35.65	**53.79**	66.33	**62.54**	402.7K (56.5%)
Full SRM	94.97	37.50	61.76	68.81	65.76	770K (0)

Table 2: RewardBench evaluation for Gemma-2B models. We start with SRM on one-fourth of the dataset. The overall performance substantially improves through SSRM, where the drop in Chat Hard results from its conflict with Chat. Notably, the SSRM performance approaches that of the model trained in a purely supervised manner.

as instruction following, reasoning, and safety. Specifically, it incorporates common LLM evaluation benchmarks formatted for reward model assessment, including MT-Bench (Zheng et al., 2024), AlpacaEval (Li et al., 2023) and HumanEval (Chen et al., 2021). The primary metric of evaluation is the accuracy of predicting the chosen response. Performance on RewardBench serves as a strong and direct indicator of the reward model's capability to align policy language models effectively.

3.2 Benchmark Evaluation

Gemma-2B We report the RewardBench evaluation on Gemma-2B in Table 2. Initially, the out-of-the-box performance of Gemma-2B-it, detailed in the first row of the table, serves as a baseline that reveals the model's rudimentary capabilities as a reward model. At this stage, its performance across all four categories closely resembles random guessing, indicating significant room for improvement. The second row represents the model trained with only a quarter of the labeled data, which shows marginal improvements. This underscores the challenges in significantly enhancing the performance of preference learning with limited labeled data.

As detailed in the subsequent rows, we apply the SSRM process described in Algorithm 1 for three iterations. Throughout these iterations, we observe a consistent improvement in performance, with the most significant gains occurring between the first and second iterations. This notable enhancement highlights the benefits of the iterative self-training approach. Note that the performance plateaus after the second iteration, as indicated by the similar metrics in the $t = 2$ and $t = 3$ rows. This suggests that once the learning capacity of the model is saturated, additional iterations may not yield further substantial gains.

We note a decrease in performance within the Chat Hard category, which can be attributed to the inherent biases in different chat categories. As observed in Dong et al. (2024); Wang et al. (2024), the Chat category is verbosity biased and the Chat Hard category is simplicity biased, each favoring responses of different lengths. These biases establish a competitive dynamic where improvements in one category can inadvertently lead to declines in the other. This relationship accounts for the observed drop in Chat Hard performance, but we see significant gains in the Chat category. The conflict of these biases is particularly pronounced in smaller models, where the limited model capacity must balance the conflicting tasks, exacerbating trade-offs. Conversely, in larger models like Llama3-8B, this conflict tends to be less obvious.

The final row reports the oracle performance of the full SRM Gemma-2B-it on the complete dataset with ground-truth labels. The model with SSRM achieves performance metrics closely approaching those of the full SRM model, despite using only one-fourth of the labeled data. This efficiency is significant, highlighting SSRM's effectiveness in leveraging pseudo-labels to enhance model performance without extensive reliance on labeled data.

We also report the number of training data used in each SSRM iteration in the last column. The proportion of pseudo-labels shows an increasing trend, illustrating the model's growing confidence in its predictions, which allows it to self-train on an expanding pool of data. A more detailed discussion on prediction confidence is presented in Section 3.4. We report the number of data used for subsequent models in Appendix A due to space limit.

Llama3-8B We report the results on Llama3-8B in Table 3. With only one-sixteenth of the full dataset, the partial SRM performance already gets a noticeable boost. This substantial early gain, compared to the marginal improvements seen with Gemma-2B under similar conditions, underscores

	Chat	Chat hard	Safety	Reasoning	Average
Llama3-8B-it	44.69	53.29	59.15	50.82	51.99
Partial SRM	96.09	40.79	62.37	76.11	68.83
SSRM [t=1]	97.77	52.85	74.04	91.25	78.98
SSRM [t=2]	96.65	**64.04**	83.42	87.03	82.78
SSRM [t=3]	**98.32**	59.21	**85.68**	**93.57**	**84.19**
Full SRM	98.60	65.35	88.81	92.07	86.21

Table 3: RewardBench evaluation for Llama3-8B models. We start with SRM on one-sixteenth of the dataset. SSRM significantly improves the performance in all categories. Different from Gemma-2B, both Chat and Chat hard improve as a result of larger model capacity.

	Chat	Chat hard	Safety	Reasoning	Average
PairRM	90.22	**53.29**	39.80	48.80	58.03
SSRM [t=1]	**93.02**	43.20	84.10	54.44	68.69
SSRM [t=2]	92.45	38.38	86.51	**58.56**	68.98
SSRM [t=3]	91.51	42.98	**88.91**	57.09	**70.12**

Table 4: RewardBench evaluation for PairRM models. No SRM step is performed as PairRM is a reward model. SSRM consistently enhances the model, showing its effectiveness for small encoder-based models.

Policy model	MT-bench score
Unaligned Gemma-2B	1.41
DPO aligned by Gemma-2B-it [Partial SRM]	1.76
DPO aligned by Gemma-2B-it [SSRM]	**2.29**

Table 5: MT-Bench evaluation for the pretrained Gemma-2B aligned by different reward models. SSRM significantly enhances the model.

the greater potential of models with larger parameter counts to leverage limited data effectively.

After applying SSRM, the performance of the model continues to improve. The improvement is particularly significant at the first iteration of SSRM, which indicates substantial gains from incorporating the self-training approach. As SSRM progresses through additional iterations, we observe further performance improvements, with notable advancements in categories like Safety and Reasoning, highlighting the model's improved reliability in sensitive scenarios. Similar to the observation in Gemma-2B experiments, the performance with SSRM approaches that of the full SRM model, with only one-sixteenth of the labeled data.

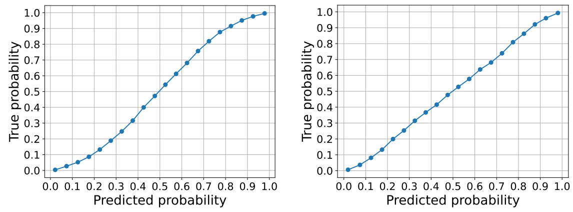
PairRM We report the results on PairRM in Table 4. Initially, since PairRM is specifically trained for preference learning, it demonstrates a strong overall capability as a reward model. The performance is noticeably better compared with Gemma-2B-it and Llama3-8B-it, despite these models possessing significantly more parameters. This difference emphasizes the distinct requirements and capabilities needed for preference learning as opposed to general language modeling tasks. However, its performance in Safety and Reasoning is considerably lower, indicating limitations in handling nuanced content out of the box.

SSRM in subsequent rows reveals a clear trend of performance enhancement across iterations. The performance improvement is especially noticeable in the Safety category, where the original PairRM underperforms. This improvement likely indicates an initial deficiency in safety-related data during the original PairRM training, a gap which our semi-supervised approach begins to fill by leveraging unsupervised data. Like the observations with Llama3-8B, the most significant gains are observed at the first iteration of SSRM, highlighting the immediate impact of the semi-supervised learning process. Similar to our previous finding, the performance gain plateaus with more iterations.

Overall, the empirical results across models of varying sizes confirm the versatility and efficiency of SSRM in improving reward model performance. By effectively employing unlabeled data, SSRM not only enhances model capabilities but also presents a cost-effective training alternative to traditional fully supervised methods. These advantages are especially important in scenarios where acquiring extensive labeled datasets is impractical due to resource constraints.

3.3 Evaluation of the aligned LM

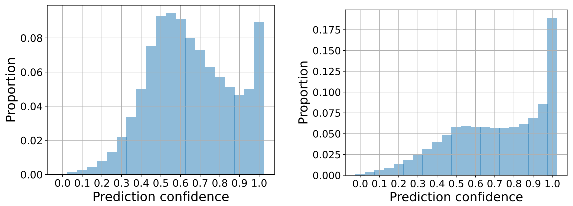
We demonstrate the effectiveness of SSRM-enhanced reward models in aligning LMs. Our experiment compares two models: SSRM-Gemma-2b-it (SSRM [t=3] in Table 2) and partial SRM-Gemma-2b-it (partial SRM in Table 2). Both models are used to pseudo-label the same set of data described in Section 3.1. We then perform one iteration of DPO on each of the resulting pseudo-labeled datasets to align a Gemma-2B model (pretrained, not instruction-tuned). We report the GPT-4 judgment scores of the aligned model on MT-bench in Table 5. The policy model aligned by the SSRM model significantly outperforms its partial SRM counterpart, demonstrating that the enhanced reward modeling abilities directly contribute to more effective alignment of the policy model. These findings align with our RewardBench evaluation, further confirming that stronger reward modeling



(a) Partial SRM.

(b) SSRM [t=3].

Figure 2: The Gemma-2B undergone three iterations of SSRM demonstrates better calibration, especially in the high-confidence score range, showing the effectiveness of confidence thresholding.



(a) Partial SRM.

(b) SSRM [t=3].

Figure 3: The prediction confidence noticeably improves on Gemma-2B models after three iterations of SSRM. Combined with better calibration, it shows the prediction more accurately reflects the actual outcome.

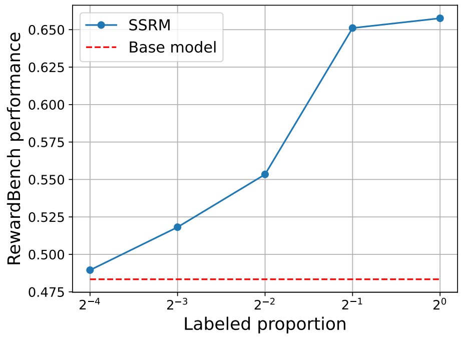


Figure 4: With more labeled data, the performance of SSRM consistently increases.

capabilities translate into improved performance in the aligned policy model.

3.4 Empirical Analysis

Calibration We show how SSRM improves the calibration of Gemma-2B models by comparing the calibration curves before and after SSRM in Figure 2. Calibration curves, which plot the true probability against the predicted probability, serve as indicators of how well the predicted probabilities of a model represent the actual outcomes. A perfectly calibrated model would have all points lying on the diagonal line. For the model after partial SRM (Figure 2a), the curve deviates from the diagonal, especially in the mid-range probabilities between 0.2 and 0.8. This deviation suggests that the model tends to be under-confident, as it predicts lower probabilities than the actual likelihood of the correct outcomes. In comparison, the model after three iterations of SSRM (Figure 2b) shows a curve that adheres more closely to the diagonal across the entire probability spectrum. This closer alignment indicates that the SSRM Gemma-2B model's predictions are more reliable and accurately reflect the true likelihoods of outcomes. The improvement implies that the SSRM has effectively used unlabeled data to correct the under-confidence and improve the overall prediction accuracy. Moreover, the notable improvement in calibration at higher confidence scores underscores the effectiveness of confidence thresholding based on the predicted probability.

Prediction confidence We show how SSRM improves the prediction confidence of the Gemma-2B models in Figure 3. The prediction confidence is computed as the predicted probability of the ground-truth label, i.e., $\pi_\theta(y|\mathbb{T}(x, a_1, a_2))$. Initially, after SRM on a quarter of the dataset, the prediction confidence on remaining samples are relatively low, with the majority concentrating around 0.5. This suggests that the model's responses on many samples are akin to random guesses at this stage. However, subsequent application of SSRM after three iterations markedly shifts the confidence distribution towards the right, as depicted in Figure 3b. This shift indicates a significant increase in the model's prediction confidence at the dataset level. Importantly, this confidence increase is not merely a case of the model becoming more confidently incorrect. Rather, it is backed by improved model calibration, as previously analyzed. The enhanced model calibration indicates the correctness of confidence scores, hence higher confidence scores indeed imply more accurate pseudo-labels.

Number of labeled data In Figure 4, we plot the effect of varying proportions of labeled data on the performance of SSRM applied to the Gemma-2B model. In this experiment, SSRM is conducted over a single iteration. The x-axis represents different fractions of the dataset that are labeled, ranging from one-sixteenth to fully labeled. The horizontal dashed line indicates the baseline performance of Gemma-2B-it out of the box. The performance of the SSRM-enhanced model improves consistently as the amount of labeled data increases. Notably, when half of the dataset is labeled, the SSRM model's performance nearly matches that achieved by finetuning on a fully supervised dataset. This trend underscores the data efficiency of SSRM, demonstrating its ability to leverage increasing amounts of labeled data effectively.

4 Related Works

Semi-Supervised Learning Semi-supervised learning (SSL) aims at blending both labeled and unlabeled data to improve the performance of a model. SSL has been widely applied to NLP tasks, including text classification (Gururangan et al., 2019; Meng et al., 2020), dependency parsing (Li et al., 2019), NER (Chen et al., 2020), and sequence generation (He et al., 2020). A cornerstone of SSL is self-training (ST), also known as pseudo-labeling (Grandvalet and Bengio, 2004; Lee et al., 2013). During ST, models assign pseudo-labels to unlabeled data and subsequently train on this augmented dataset in an iterative manner. This method has inspired numerous adaptations, including mean teacher (Tarvainen and Valpola, 2017), noisy student (Xie et al., 2020) and FixMatch (Sohn et al., 2020). These ST-based techniques have proven particularly effective in applications where labeled data is scarce or expensive to obtain, such as unsupervised domain adaptation (Li et al., 2019; Kumar et al., 2020; He et al., 2023). These methods have also been applied in preference learning (Cao et al., 2021; Park et al., 2021) in the general RL settings, but mainly with applications in the vision and robotics domains. In this work, we extend these methodologies to the domain of reward modeling in RLHF, aiming to address the specific challenges posed by the high dependency on labeled data in training language models.

Reinforcement Learning from Human Feedback (RLHF) Traditional RLHF approaches (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Bai et al., 2022b) involve training a reward model using human-annotated preference data, and then use the reward signal provided by the reward model to align the behavior of language models with human values, using RL techniques such as Proximal Policy Optimization (PPO) (Schulman et al., 2017). However, the training of PPO is challenging due to its inefficiency and instability compared with SFT. This drawback makes the results of PPO-based RLHF model such as ChatGPT (Achiam et al., 2023) largely non-reproduced in the open-source community. To overcome these limitations, alternative approaches like Rejection Sampling Finetuning (RSF) (Dong et al., 2023; Gulcehre et al., 2023; Yuan et al., 2024b) have emerged. RSF learns from high-quality examples chosen by a reward model, then SFT on them. The pipeline has also shown success in aligning language models, including Llama-2 (Touvron et al., 2023). Both RSF and RLHF operate in an online manner, meaning that the preference is judged based on the generation of the models. Alternatively, Direct Preference Optimization (DPO) (Rafailov et al., 2024) aligns models in an offline manner, simplifying the process by tuning models directly on a curated preference dataset without the need for a separate reward model.

Our proposed SSRM is beneficial across both online and offline frameworks. In the online scenarios, SSRM produces an enhanced reward model without requirement on more labeled data. In the offline case, the improved reward model can serve as an annotator that produces high-quality preference dataset for the subsequent procedure such as DPO. This versatility makes SSRM a valuable asset in advancing the field of RLHF.

Reinforcement Learning from Artificial Intelligence Feedback (RLAIF) The training of reward models in the traditional RLHF process requires substantial human-annotated preference data, incurring high cost for labeling. To mitigate these challenges, researchers have explored using language models themselves as a source of preference feedback, thus replacing the need for extensive human intervention. For instance, Bai et al. (2022a) uses a language model to provide feedback and refine responses, enhancing reward models. This approach has been further validated by Lee et al. (2023), which demonstrates that AI-generated feedback can achieve comparable results to human feedback while significantly reducing the need for human labor. More recent advancements involve using LLMs directly as judges in a process known as LLM-as-a-Judge prompting (Li et al., 2023; Dubois et al., 2024; Bai et al., 2024), where LLMs are provided with specific rubrics and tasked with scoring the responses accordingly.

While RLAIF has shown potential by leveraging powerful LLMs to simulate human feedback, this approach often relies on highly capable and therefore expensive LLMs (e.g., GPT-4) to achieve human-comparable feedback quality. This dependency means that querying these advanced models still incurs significant costs, limiting the accessibility and scalability of RLAIF approaches. In contrast, SSRM is designed to enhance language models of various sizes and capacities, including models with size as small as 0.4B parameters. This method allows even smaller, less resource-intensive models to improve their performance significantly, thus broadening the accessibility of effective training techniques and substantially reducing the costs associated with acquiring accurate feedback.

5 Conclusion

In this work, we address the substantial reliance of reward model training on extensive human-annotated preference data. We introduce Semi-Supervised Reward Modeling (SSRM), a method that mitigates this issue by effectively utilizing unlabeled data in conjunction with limited labeled data. The SSRM process consists of three primary steps: pseudo-labeling the unlabeled examples, selecting those examples with high prediction confidence, and finetuning the model on this augmented dataset. Our extensive experimental evaluations across models of varying sizes demonstrate that SSRM significantly enhances the performance of reward models with minimal requirement of labeled data. Furthermore, the performance of models trained using SSRM closely approaches that of models trained with equivalent volumes of fully supervised data. This performance is further supported by detailed analyses of calibration and prediction confidence, which underscore the robustness and effectiveness of SSRM. Overall, SSRM offers a highly efficient strategy for improving reward models, significantly reducing the need for costly and time-consuming data annotation.

Limitations

One constraint of the SSRM framework is its dependency on the initial reward modeling capabilities of the model. Especially when labeled data is scarce, the model after only the initial stage of supervised training might not acquire enough knowledge to accurately assign pseudo-labels to the unlabeled dataset. This limitation poses a risk of generating low-quality pseudo-labels, which could potentially propagate errors through the training process. Nevertheless, it is important to note that we incorporate a confidence thresholding step, which substantially mitigates this issue.

Acknowledgement

YH, HW and HZ are partially supported by an NSF IIS grant No. 2416897. HZ would also like to thank the support from a Google Research Scholar Award. YH and HW are partially supported by a research grant from the Amazon-Illinois Center on AI for Interactive Conversational Experiences (AICE). The views and conclusions expressed in this paper are solely those of the authors and do not necessarily reflect the official policies or positions of the supporting companies and government agencies.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

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

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, et al. 2024. Benchmarking foundation models with language-model-as-an-examiner. *Advances in Neural Information Processing Systems*, 36.

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

Zehong Cao, KaiChiu Wong, and Chin-Teng Lin. 2021. Weak human preference supervision for deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5369–5378.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. 2006. Semi-supervised learning. adaptive computation and machine learning series.

Jiaao Chen, Zhenghui Wang, Ran Tian, Zichao Yang, and Diyi Yang. 2020. Local additivity based data augmentation for semi-supervised NER. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1241–1251, Online. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Luigi Daniele and Suphavadeeprasit. 2023. Amplify-instruct: Synthetically generated diverse multi-turn conversations for effecient llm training. *arXiv preprint arXiv:(coming soon)*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. 2023. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *Preprint*, arXiv:2405.07863.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning*, pages 5988–6008. PMLR.

Yoav Freund. 1995. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285.

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. 2019. Variational pretraining for semi-supervised text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5880–5894, Florence, Italy. Association for Computational Linguistics.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

Yifei He, Haoxiang Wang, Bo Li, and Han Zhao. 2023. Gradual domain adaptation: Theory and algorithms. *arXiv preprint arXiv:2310.13852*.

Shengyi Costa Huang, Agustín Piqueres, Kashif Rasul, Philipp Schmid, Daniel Vila, and Lewis Tunstall. 2024. Open hermes preferences. `https://huggingface.co/datasets/argilla/OpenHermesPreferences`.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178.

Ananya Kumar, Tengyu Ma, and Percy Liang. 2020. Understanding self-training for gradual domain adaptation. In *International conference on machine learning*, pages 5468–5479. PMLR.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta.

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

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. 2022. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Zhenghua Li, Xue Peng, Min Zhang, Rui Wang, and Luo Si. 2019. Semi-supervised domain adaptation for dependency parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2386–2395, Florence, Italy. Association for Computational Linguistics.

W Lian, B Goodson, E Pentland, et al. 2023. Openorca: An open dataset of gpt augmented flan reasoning traces.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. 2023. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*.

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

Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. 2021. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In *International Conference on Learning Representations*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Robert E Schapire. 1990. The strength of weak learnability. *Machine learning*, 5:197–227.

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

Matthias Seeger. 2000. Learning with labeled and unlabeled data.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024. Helpsteer2: Open-source dataset for training top-performing reward models. *Preprint*, arXiv:2406.08673.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, et al. 2023. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *arXiv preprint arXiv:2311.09528*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. 2024a. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024b. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

	# Data (Pseudo-labeled portion)
`Llama3-8B-it`	0 (0)
Partial SRM	43.75K (0)
SSRM [t=1]	95.15K (54.02%)
SSRM [t=2]	290.75 (84.95%)
SSRM [t=3]	351.75 (87.56%)
Full SRM	700K (0)

Table 6: `Llama3-8B` models with number of data.

	# Data (Pseudo-labeled portion)
`PairRM`	0 (0)
SSRM [t=1]	153K (100%)
SSRM [t=2]	198K (100%)
SSRM [t=3]	281K (100%)

Table 7: `PairRM` models with number of data.

A More experimental results

Implementation We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 5×10^{-6} and a cosine learning rate schedule that includes 40 warmup steps. We use a context window of 3072 tokens with sample packing. The training process for each iteration is completed over one epoch, utilizing a global batch size of 128. In our implementation, each iteration starts from the same initial checkpoints (e.g., `Gemma-2B-it` for the Gemma experiments) instead of the checkpoint from the previous iteration, as training LLMs for more than one epoch is likely to lead to overfitting (Wu et al., 2021; Ouyang et al., 2022), hurting the performance. This also ensures a fair comparison with the full SRM model reported at the end, as they both execute with one epoch. To ensure the reliability of our semi-supervised learning process, we choose a confidence threshold of 0.8.

The experiments are run on NVIDIA A6000 GPUs with 48GB memory. In terms of running SRM on the full dataset, `PairRM` requires 60 GPU hours, `Gemma-2B` requires 20 GPU hours, and `Llama3-8B` requires 128 GPU hours.

Number of data used for training In Tables 6 and 7, we report the number of data used in each iteration of SSRM for `Llama3-8B` and `PairRM`. Similar to the findings as Table 2, with more iterations, an increasing number of pseudo-labeled data is included in the augmented dataset, as a result of the model's growing confidence in prediction.

	Chat	Chat hard	Safety	Reasoning	Average
Llama3-8B-it	44.69	53.29	59.15	50.82	51.99
Partial SRM (1/16)	96.09	40.79	62.37	76.11	68.83
Partial SRM (1/4)	98.04	59.21	84.37	93.26	83.72
Full SRM	98.60	65.35	88.81	92.07	86.21

Table 8: Performance comparison for Llama3-8B using different number of data for SRM.

Number of data used for initial SRM Here, we explain the reason to use different amount of data for the initial SRM for Gemma-2B and Llama3-8B experiments. As shown in Table 8, using only one-fourth of the data for SRM on Llama3-8B-it already achieves an average performance only 2.5% worse than that of the full SRM result, demonstrating saturation. In this case, we do not expect using more unlabeled data can continually enhance the performance, so we use one-sixteenth for the partial SRM instead, which leaves a larger room for improvement.

B Dataset Details

HH-RLHF, UltraFeedback and UltraInteract are under MIT license. HelpSteer and PKU-SafeRLHF are under CC-BY-4.0 license. Distilabel-Capybara and Distilabel-Orca are under Apache-2.0 license. We cannot find license information for SHP and OpenHermesPreferences.

The data does not contain information that can be used to uniquely identifies individual people or offensive content.

C Potential Risks

This paper presents work whose goal is to advance the field of NLP. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.