Preference-Guided Reflective Sampling for Aligning Language Models

Hai Ye
Department of Computer Science
National University of Singapore
yehai@comp.nus.edu.sg

Hwee Tou Ng
Department of Computer Science
National University of Singapore
nght@comp.nus.edu.sg

Abstract

Iterative data generation and model re-training can effectively align large language models (LLMs) to human preferences. The process of data sampling is crucial, as it significantly influences the success of policy improvement. Repeated random sampling is a widely used method that independently queries the model multiple times to generate outputs. In this work, we propose a more effective sampling method, named *Preference-Guided Reflective Sampling (PRS)*. Unlike random sampling, *PRS* employs a tree-based generation framework to enable more efficient sampling. It leverages adaptive self-refinement techniques to better explore the sampling space. By specifying user preferences in natural language, *PRS* can further optimize response generation according to these preferences. As a result, *PRS* can align models to diverse user preferences. Our experiments demonstrate that *PRS* generates higher-quality responses with significantly higher rewards. On AlpacaEval and Arena-Hard, *PRS* substantially outperforms repeated random sampling in best-of-N sampling. Moreover, *PRS* shows strong performance when applied in iterative offline RL training[1].

1 Introduction

Large language models (LLMs) have made significant advances (Radford et al., 2019; Brown et al., 2020; OpenAI, 2022). These models are typically aligned with human expectations through fine-tuning. This is achieved by using reinforcement learning from human feedback (RLHF), which mitigates the generation of harmful, biased, or irrelevant outputs (Perez et al., 2022). Both online and offline RL methods have been explored for RLHF (Schulman et al., 2017; Gülçehre et al., 2023; Rafailov et al., 2023). Iterative offline training provides a more efficient alternative than online training, by allowing outputs to be pre-generated and reused to facilitate iterative improvements in policy.

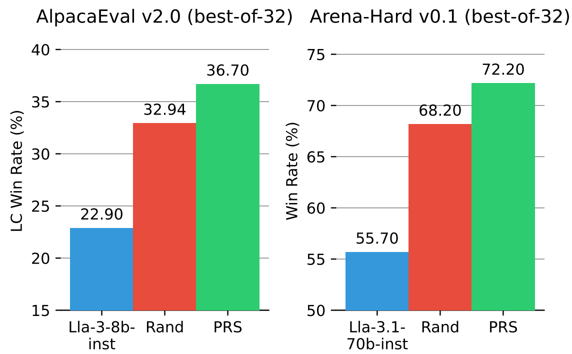


Figure 1: Performance comparison of *PRS* (ours) and repeated random sampling (Rand) on AlpacaEval v2.0 and Arena-Hard v0.1 using best-of-32 sampling. Each prompt samples 32 responses using Rand or *PRS* and the response with the highest reward is kept for evaluation.

Effective data sampling is crucial for iterative model re-training, as it directly influences the effectiveness of the resulting policy (Gülçehre et al., 2023). Repeated random sampling (as shown in Fig. 2) is an effective method and has been widely used for data generation in previous work (Gülçehre et al., 2023). It independently calls the model multiple times to get samples. Then higher-quality data will be maintained to update the policy model. However, the vast output space compromises its efficiency since the inherent randomness may result in inefficient exploration in the sampling space. Also, the simple generation strategy cannot learn from and adapt dynamically based on previously generated samples. Furthermore, with only the supervision of the reward model, it is hard to optimize the outputs to align to diverse and personalized preferences.

We propose a new sampling method named *Preference-Guided Reflective Sampling (PRS)* to improve data generation. Different from random

[1] Source code of this paper is available at `https://github.com/nusnlp/PRS`.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 21646–21668
November 12-16, 2024 ©2024 Association for Computational Linguistics

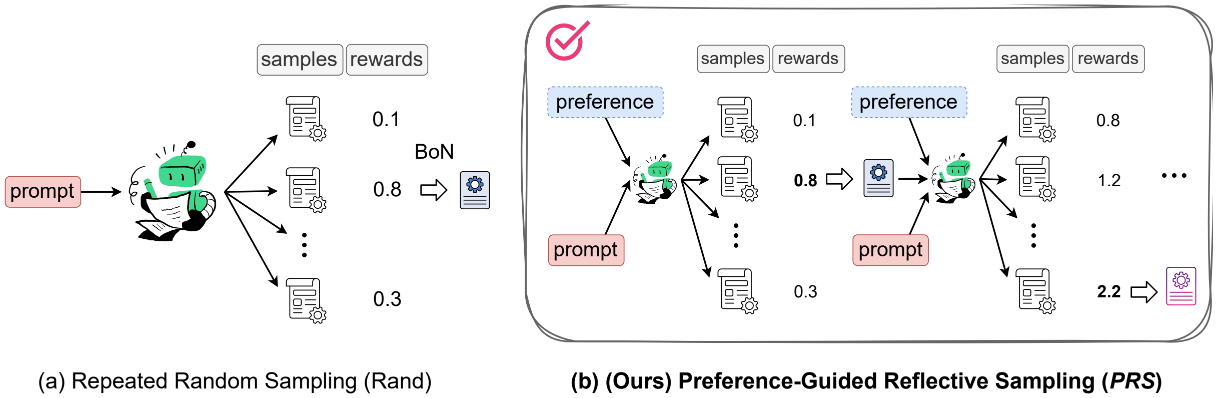


Figure 2: Comparison of repeated random sampling and our method *PRS*. *PRS* adopts a tree-based generation framework that learns to adapt and adjust its outputs by reflecting on its already generated data. It can incorporate a specific user preference to optimize responses that align with it. Adjusting preferences will generate tailored responses. For random sampling, it generates samples independently and can use the best-of-N (BoN) method to find the best sample. Both methods share the same sampling budget, which samples the same number of responses for each prompt.

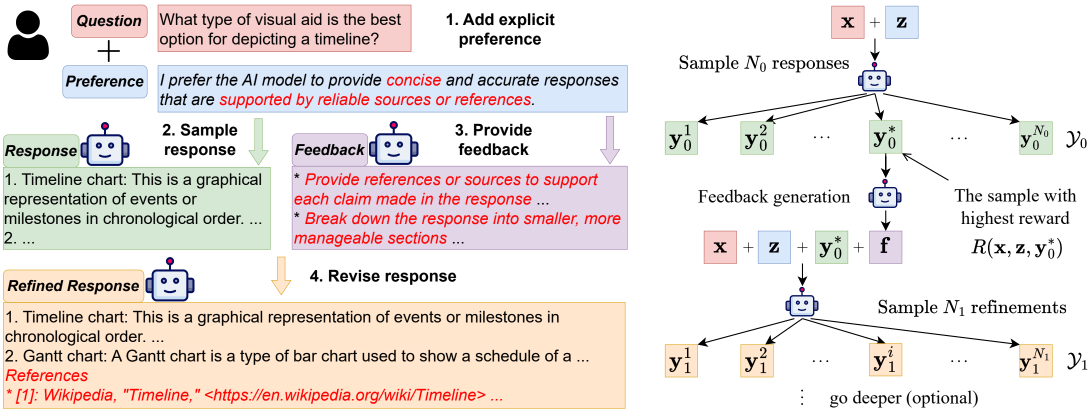
sampling, *PRS* employs a tree-based generation framework to balance exploration and exploitation throughout the generation process (see Fig. 2). It learns to adapt and adjust its outputs by reflecting on its already generated data so that it can improve the sampling of future samples. Furthermore, by using a preference described in natural language, *PRS* can optimize the response toward this explicit preference. The user preference is incorporated as an additional sampling context, guiding the model toward more relevant directions and minimizing unnecessary exploration. As a result, it achieves more efficient sampling and can also generate samples aligned to diverse preferences.

We study preference-controlled text generation for the task of instruction following and keyword-focused document summarization. In our experiments, we first evaluate *PRS* against various baselines in generating training samples with diverse policy models (§ 5.1). In § 5.2, we investigate its application for aligning LLMs to adhere to explicit preferences provided in the inputs using offline RL training. We further explore preference adaptation, toxicity reduction, and other areas in § 5.3. Our contributions in this work are as follows:

- We introduce *PRS*, a novel sampling method to improve data generation. *PRS* is capable of generation tailored to different preferences.
- Experiments with 9 policy models show that *PRS* generates training data with higher rewards. On AlpacaEval and Arena-Hard, *PRS* achieves better performance than repeated random sampling in the best-of-N setting (Fig. 1).
- With extensive offline RL training, the outcomes across multiple benchmarks, e.g., AlpacaEval (Li et al., 2023) highlight the effectiveness of *PRS*.
- Further analysis demonstrates *PRS*'s superior performance in preference adaptation.

2 Related Work

Offline RL offers an efficient alternative to online RL (Schulman et al., 2017). Dong et al. (2023), Gülçehre et al. (2023), and Rafailov et al. (2023) emphasize data generation and model refinement. Repeated random sampling is a simple but effective method for data generation. Brown et al. (2024) demonstrate that scaling inference compute can significantly improve the model performance in problem solving. Bai et al. (2022b) leverage the LLM's reflection capacity to continuously refine model responses. However, they only focus on harmless responses, whereas our work is applicable across a broad spectrum of preferences. Moreover, different from ours, their work does not aim to improve data sampling for RL training. Feng et al. (2023) use Monte Carlo tree search (MCTS) with token-level rewards, but ours employs sequence-level rewards based on cost-effective tree-based generation, with input preferences to guide the generation. Scheurer et al. (2023) advocate for training models using human language feedback, but we employ the model itself to generate language feedback. A more detailed discussion of the related work is in Appendix A.



(a) Reflective Refinement **(b) Tree-based Generation**

Figure 3: ***PRS***: (a) Example: A user requests a brief response with supporting references. The initial response lacks references. After feedback, the revised response includes appropriate references. (b) A preference $\boldsymbol{z}$ is added to the input $\boldsymbol{x}$. The process begins by sampling N_0 initial responses $\mathcal{Y}_0$, from which the optimal response $\boldsymbol{y}_0^*$ is selected using a reward model R. Then feedback $\boldsymbol{f}$ is generated, leading to the sampling of N_1 refinements $\mathcal{Y}_1$ to enhance $\boldsymbol{y}_0^*$. Finally, $\mathcal{Y}_0$ and $\mathcal{Y}_1$ are merged. Optionally, new refinements may be sampled based on the current best response.

3 Preliminaries

Offline RL. RLHF utilizes human feedback to fine-tune a pre-trained LLM with human preferences. The preference human feedback can be utilized to train a reward model $R(\boldsymbol{x}, \boldsymbol{y})$, given an input $\boldsymbol{x}$ and a response $\boldsymbol{y}$. Following Gülçehre et al. (2023), we employ offline RL to synchronize the LLM policy with the trained reward model. This process, beginning with the policy initialized by supervised fine-tuning (SFT) on labeled data, involves iterative cycles of data generation and model re-training.

The policy of the LLM, π_θ, parameterized by θ, produces a response $\boldsymbol{y}$ given the input $\boldsymbol{x}$, i.e., $\boldsymbol{y} \sim \pi_\theta(\boldsymbol{y}|\boldsymbol{x})$. Using the labeled data $\mathcal{D}_0$, the LLM is trained with the negative log-likelihood (NLL):

$$\mathcal{L}_{NLL} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}_0}\big[-\log \pi_\theta(\boldsymbol{y}|\boldsymbol{x})\big] \quad (1)$$

Then it repeats data generation and model re-training to align the language model:

1. **Data Generation:** Each iteration leverages the policy from the previous cycle to generate N responses $\boldsymbol{y}$ for each input $\boldsymbol{x}$ in the unseen dataset $\mathcal{U}_k$. The reward model, $R(\boldsymbol{x}, \boldsymbol{y})$, evaluates these responses to generate rewards. Best-of-N strategy or a reward threshold is used to identify the high-quality examples.
2. **Model Re-training:** The newly generated data, along with all prior data, is used to refine the model in the subsequent re-training phase.

4 Method

We aim to improve the data generation process to enhance offline RL training. We first introduce *Preference-Guided Reflective Sampling (PRS)*, and then study the task of preference-controlled instruction following using offline RL training.

4.1 Preference-Guided Reflective Sampling

PRS aims to optimize the response aligned to a given user preference described in natural language. The user preference describes the desired model output, such as *conciseness*. Let $\boldsymbol{z}$ denote a specific preference, exemplified by statements like "*I prefer the response to be concise.*" or "*Can you give me a response without wordy explanations?*". *PRS* aims to generate the responses aligned to the preference $\boldsymbol{z}$.

Initially, we sample a response $\boldsymbol{y}_0$ conditioned on both $\boldsymbol{x}$ and $\boldsymbol{z}$, by appending $\boldsymbol{z}$ to the input $\boldsymbol{x}$. Subsequently, we engage the LLM policy in a process of self-reflection, aiming to iteratively refine $\boldsymbol{y}_0$ to better align with the stated preference. Given the independence of preference $\boldsymbol{z}$ and input $\boldsymbol{x}$, we redefine the generation process of $p(\boldsymbol{y}|\boldsymbol{x})$ as:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{z},\boldsymbol{y}_0,\boldsymbol{f}} p(\boldsymbol{z}) \times \underbrace{\pi_\theta(\boldsymbol{y}_0|\boldsymbol{x},\boldsymbol{z})}_{\text{Initial Sampling}} \times \underbrace{\pi_\theta(\boldsymbol{f}|\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}_0) \times \pi_\theta(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}_0,\boldsymbol{f})}_{\text{Reflective Refinement}} \quad (2)$$

where for the reflective refinement, the model first generates language feedback $\boldsymbol{f}$ for the output $\boldsymbol{y}_0$, then revises the output by incorporating the feedback to obtain a new response $\boldsymbol{y}$ (see Fig. 3a). Promoting the model to provide language feedback is to provide enriched information to guide the model

Algorithm 1 *PRS*

Input: Input prompt $\boldsymbol{x}$; preference $\boldsymbol{z}$; model π_θ; reward model R; number of layers d; total samples N to generate
Initialize: Layer width $w = \lfloor \frac{N}{d} \rfloor$
$\mathcal{Y} \leftarrow \emptyset$
for $l = 0$ **to** $d - 1$ **do**
  Select $\boldsymbol{y}^*$ with the highest score from $\mathcal{Y}$ or set $\boldsymbol{y}^*$ to None if $\mathcal{Y}$ is $\emptyset$
  $\boldsymbol{f} \sim \pi_\theta(\cdot|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}^*)$ if $\boldsymbol{y}^*$ is not None else None
  for $i = 1$ **to** w **do**
    Sample $\boldsymbol{y}_i \sim \pi_\theta(\cdot|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}^*, \boldsymbol{f})$
    Add $\boldsymbol{y}_i$ to $\mathcal{Y}$
  end for
  Compute $R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$ for newly generated samples in $\mathcal{Y}$
end for
Output: The best final response $\boldsymbol{y}^*$

in revising its response. We can adjust the user preference $\boldsymbol{z}$ to generate outputs aligned to different preferences, e.g., detailed or humorous responses.

Tree-Based Generation. For each input, we sample N responses for further selection. However, as Eq. 2 indicates, various components (i.e., $\boldsymbol{z}$, $\boldsymbol{y}_0$, $\boldsymbol{f}$) control the generation, causing difficulty in efficient generation. To overcome this issue, we propose tree-based generation (Fig. 3b), which utilizes an iterative exploration and exploitation process:

1. First, the model randomly samples N_0 initial responses $\mathcal{Y}_0$ from $\pi_\theta(\boldsymbol{y}_0|\boldsymbol{x}, \boldsymbol{z})$, and the reward model $R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$ generates rewards for the samples. The response $\boldsymbol{y}_0^*$ with the highest reward is selected for further exploration.
2. Then the model generates language feedback $\boldsymbol{f}$ for $\boldsymbol{y}_0^*$, i.e., $\boldsymbol{f} \sim \pi_\theta(\boldsymbol{f}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*)$, which is the suggestion to further modify $\boldsymbol{y}_0^*$ to be more in line with the preference $\boldsymbol{z}$ (see the example prompt in Fig. 14).
3. The model generates another set of $N_1 = N - N_0$ refinements $\mathcal{Y}_1$ from $\pi_\theta(\boldsymbol{y}_1|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*, \boldsymbol{f})$, where N is the total number of samples per prompt. It aims to adjust the generation towards even better rewards (see the prompt of Fig. 15).
4. We combine $\mathcal{Y}_0$ and $\mathcal{Y}_1$ into $\mathcal{Y}$ that has N samples for the input $\boldsymbol{x}$.
5. (Optional) In layer l, suppose we have samples $\mathcal{Y}^{(l-1)} = \mathcal{Y}_0 \cup \cdots \cup \mathcal{Y}_{l-1}$ until layer $l-1$. We further sample refinements $\mathcal{Y}_l$ with a–c steps:

$$\boldsymbol{y}^* \leftarrow \arg\max_{\boldsymbol{y}_i \in \mathcal{Y}^{(l-1)}} R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_i) \tag{3a}$$

$$\boldsymbol{f} \sim p(\cdot|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}^*) \tag{3b}$$

$$\mathcal{Y}_l \sim p(\cdot|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}^*, \boldsymbol{f}) \tag{3c}$$

Eq. 3a identifies the optimal response from all already generated responses (i.e., exploitation), followed by refinements (exploration).

We present the pseudocode of *PRS* in Algorithm 1. It is worth noting that *PRS* is also functional when the preference $\boldsymbol{z}$ is not provided as input. Additionally, feedback can be omitted during the generation of refinements. In Algorithm 1, the number of samples generated for each layer is set to be the same. However, in practice, other hyper-parameters can be used.

Reward Estimation. In *PRS*, the reward for a response is calculated using the formula $R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$, where $\boldsymbol{z}$ specifies the preference for aspects to be focused on when assessing the response. However, if the specified preference $\boldsymbol{z}$ aligns with the implicit preference already incorporated into the reward model, the formula can be simplified to $R(\boldsymbol{x}, \boldsymbol{y})$. In this case, the reward model automatically evaluates the response based on its built-in preference, without the need for $\boldsymbol{z}$. To achieve high rewards, it is crucial to understand and articulate the internal preference of the reward model.

4.2 Alignment for Preference-Controlled Text Generation

Here, we study the task of preference-controlled text generation. We train the model to produce responses aligned with the input-specified preference, i.e., $\boldsymbol{y}(\boldsymbol{z}) \sim \pi_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$. We adopt offline RL in § 3 for training, which repeats iterations of data generation and model re-training.

As indicated by Eq. 2, adjusting the preference $p(\boldsymbol{z})$ can generate diverse outputs, each tailored to a specific preference. Without loss of generality, we do not focus on one specific personalized preference. Instead, we consider diverse preferences. We annotate diverse preferences to ensure each input question is associated with a different preference from others. As exemplified by Table 1, the task of instruction following has diverse personalized preferences and for document summarization, the keywords vary for different documents.

Algorithm 2 in the Appendix is the pseudocode for training. Specifically, we conduct K iterations of offline RL training. In each iteration k, we have an unlabeled set $\mathcal{U}_k = \{(\boldsymbol{x}, \boldsymbol{z})\}$ and we initialize the training set $\mathcal{D}_k$ to $\emptyset$. For each data point

Task 1: Instruction following
Common Preference
I prefer responses that are informative, precise, creative, detailed, relevant, and in-depth.
Personalized Preferences
[1] I prefer the model to provide a concise and accurate answer without any unnecessary details or explanations.
[2] I prefer clear and well-organized responses that provide step-by-step instructions or explanations. Additionally, I appreciate when the response includes code snippets or examples for better understanding.
...
Task 2: Keyword-focused summarization
I prefer a response that is strictly within 3 sentences, focusing on the keywords of {*specify three keywords here*}.

Table 1: The explicit preferences used for response optimization. They are added after the input question or document. For instruction following, we evaluate common and personalized preferences.

$(\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{U}_k$, we sample N responses in total. We first generate N_0 initial responses denoted as $\mathcal{Y}_0$ and then $N_1 = N - N_0$ refinements denoted as $\mathcal{Y}_1$. We use a reward model to select high-quality data for training. To enhance tree-based generation, we aim to optimize the following two components:

• $\pi_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$**:** It trains the policy to generate responses aligned with input preferences. We use the reward model to identify the response $\boldsymbol{y}^*$ with the highest reward from $\mathcal{Y}_0 \cup \mathcal{Y}_1$, and we add the data of $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}^*)$ to the training set $\mathcal{D}_k$.

• $\pi_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0, \boldsymbol{f})$**:** To improve the model's refinement ability, we construct improving pairs from $\mathcal{Y}_0$ and $\mathcal{Y}_1$. We only keep samples from $\mathcal{Y}_1$ that are refined based on the response $\boldsymbol{y}_0^*$ if their rewards exceed $\boldsymbol{y}_0^*$. The set of improving pairs is formalized as:

$$\mathcal{Q} = \Big\{ (\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*, \boldsymbol{f}, \boldsymbol{y}_1) \mid R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_1) > R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*), \forall \boldsymbol{y}_1 \in \mathcal{Y}_1 \Big\} \tag{4}$$

In our setting, if $\mathcal{Q}$ is not empty, we add the improving data of $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*, \boldsymbol{f}, \boldsymbol{y}_1^*)$ into the training set $\mathcal{D}_k$, where $\boldsymbol{y}_1^*$ is the response with the highest reward from $\mathcal{Y}_1$. This is the same idea as best-of-N sampling, to maximize the response's reward after the model's refinement.

After generating data from $\mathcal{U}_k$, we combine the generated training data up to iteration k, i.e., $\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_k$. Then we re-train the policy with the following objective, which refers to the NLL

	AlpacaEval v2.0		Arena-Hard v0.1	
Method	LC WR	WR	Method	WR
Mis-7B-Inst-v0.2	17.10	14.70	mis-large-2407	70.4
Rand (Bo-16)	23.90	19.86	Rand (Bo-16)	77.0
PRS (Bo-16)	**27.19**	**19.87**	*PRS (Bo-16)*	**79.3**
Rand (Bo-32)	24.85	**20.61**	Rand (Bo-32)	79.1
PRS (Bo-32)	**27.17**	20.03	*PRS (Bo-32)*	**80.3**
Lla-3-8b-inst	22.90	22.60	Lla-3.1-70b-inst	55.7
Rand (Bo-16)	31.00	28.75	Rand (Bo-16)	69.5
PRS (Bo-16)	**35.05**	**31.92**	*PRS (Bo-16)*	**69.8**
Rand (Bo-32)	32.94	30.43	Rand (Bo-32)	68.2
PRS (Bo-32)	**36.70**	**33.46**	*PRS (Bo-32)*	**72.2**
Gemma-2-9b-it	48.61	37.07	qwen2-72b-inst	46.9
Rand (Bo-16)	55.07	**44.51**	Rand (Bo-16)	61.9
PRS (Bo-16)	**58.40**	43.86	*PRS (Bo-16)*	**62.1**
Rand (Bo-32)	57.61	45.10	Rand (Bo-32)	63.9
PRS (Bo-32)	**59.85**	**46.41**	*PRS (Bo-32)*	**65.4**

Table 2: Results of best-of-N (Bo-N) sampling on AlpacaEval and Arena-Hard benchmarks, compared to the results of one-pass inference. We use ArmoRM-Llama3-8B-v0.1 as the reward model. Each prompt samples N responses using repeated random sampling or *PRS* and the best response with the highest reward is kept for evaluation. Here, *PRS* does not include preference in the input, and feedback is not generated during refinement. *PRS* uses the version of *PRS* ($N/2$, $N/2$). The higher score between *PRS* and Rand is highlighted in bold. LC WR is the abbreviation for length-controlled win rate.

loss in Eq. 1:

$$\begin{aligned} \mathcal{L}(\theta) &= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}_0}\big[-\log \pi_\theta(\boldsymbol{y}|\boldsymbol{x})\big] \\ &+ \mathbb{E}_{(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}^*)\sim\mathcal{D}}\big[-\log \pi_\theta(\boldsymbol{y}^*|\boldsymbol{x},\boldsymbol{z})\big] \\ &+ \mathbb{E}_{(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}_0^*,\boldsymbol{f},\boldsymbol{y}_1^*)\sim\mathcal{D}}\big[-\log \pi_\theta(\boldsymbol{y}_1^*|\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}_0^*,\boldsymbol{f})\big] \end{aligned} \tag{5}$$

where the labeled training data $\mathcal{D}_0$ is also included. After K iterations of RL training, we obtain the model $\pi_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$ that can generate the response $\boldsymbol{y}$ aligned to the preference $\boldsymbol{z}$.

5 Experiments

Dataset. To align models for preference-controlled text generation, i.e., instruction following and keyword-focused document summarization, we used the following dataset for supervised fine-tuning (SFT) and RL training:

• **Instruction Following.** For SFT data, from the widely used dataset ShareGPT[2], we randomly sample 10k conversations with a maximum of three rounds for each conversation, resulting in 21,934 labeled data points in total. Prompts from Alpaca-GPT4 (Peng et al., 2023) are used for RL training.

[2] https://sharegpt.com/

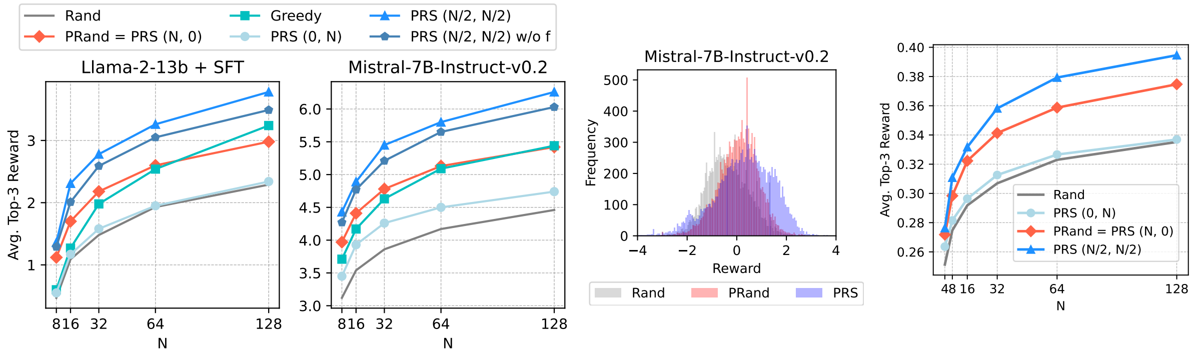


Figure 4: Comparing sampling methods. **Left:** We study the common preference and use the description of Table 1 to generate detailed and in-depth responses. With 100 random prompts from Alpaca-GPT4, each method samples N responses per prompt (i.e., 8, 16, 32, 64, or 128). The top three highest rewards are averaged for each prompt, leading to an overall average score for the entire evaluation set. The full results of 9 policy models are shown in Fig. 9. **Middle:** The distribution of rewards with $N = 128$, where *PRS* is *PRS* $(N/2, N/2)$. **Right:** Summarization results on 100 random documents from CNN / Daily Mail. The policy model is Llama-2-13b+SFT.

• **Summarization.** We use the same SFT data from ShareGPT for instruction tuning. We further sample 2,500 documents from CNN / DailyMail (See et al., 2017) for RL training.

Preference Annotation. We first need to annotate the preferences for the unlabeled prompts. We show some sample preferences in Table 1.

• **Instruction Following.** The Alpaca-GPT4 dataset initially does not include user preferences, so we use GPT-3.5-turbo to automate the generation of preferences by pairing each prompt with a random profession from a list of 222 professions. This method aims to mirror personalized preferences across various professions, thereby enriching dataset diversity. For details on this process and examples, see Appendix E.

• **Summarization.** To get the input preference keywords, we prompt GPT-3.5-turbo to extract three keywords from the ground-truth summary.

Benchmarks. For instruction following, we use AlpacaEval (Li et al., 2023) with 805 test samples and Arena-Hard (Li et al., 2024) with 500 test samples. For summarization, we further sample 1k data from CNN / DailyMail as the test set.

Reward Model. For instruction following, we use UltraRM-13B (Cui et al., 2023), a Llama-2-13B model tuned with GPT-4 preference feedback. It achieved SOTA results on multiple public preference test sets, including Anthropic Helpful (Bai et al., 2022a). For summarization, since we lack a reward model, we simulate rewards by comparing summaries to the ground truth using average F1 scores from Rouge-1, -2, and -L (Lin, 2004). Lastly, we use ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024) for best-of-N sampling on AlpacaEval and Arena-Hard.

Baselines. We compare various sampling methods with *PRS*:

- **Rand** is repeated random sampling conditioned on the input $\boldsymbol{x}$ using $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$.
- **PRand** adds an explicit preference $\boldsymbol{z}$ to the input $\boldsymbol{x}$, i.e., $\boldsymbol{y} \sim \pi_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$, for random sampling.
- **Greedy** utilizes a greedy algorithm, where we improve the method from Madaan et al. (2023) which iteratively refines the last response. Specifically, the enhanced baseline starts by sampling an initial response with $\pi_\theta(\boldsymbol{y}_0|\boldsymbol{x}, \boldsymbol{z})$. It uses a reward model to continually update the highest-reward response $\boldsymbol{y}^*$ with $\pi_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}^*)$. During each revision round, if a new response $\boldsymbol{y}$ achieves a higher reward, it becomes $\boldsymbol{y}^*$.

We use temperature sampling during response generation.

5.1 Comparison of Sampling Methods

We first compare different sampling methods for data generation. We expect a good sampling method to obtain a training set with a higher reward. Here, we only consider two layers for the tree-based generation in *PRS*. Since the ***PRS*** (N_0, N_1) method is affected by the hyper-parameters N_0 and N_1, we adjust them to examine their impact:

- ***PRS*** $(0, N)$ samples one response $\boldsymbol{y}_0$, generates feedback $\boldsymbol{f}$, and then samples N refinements. It neglects the exploration for $\boldsymbol{y}_0$.
- ***PRS*** $(N, 0)$ samples N responses of $\boldsymbol{y}_0$ without refinement, which neglects the exploration of $\boldsymbol{y}_1$. This is precisely the PRand baseline.
- ***PRS*** $(N/2, N/2)$ balances exploring $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$.

Test Set Size	805	200	200
Annotator Baseline Methods	GPT4 davinci-3 % Win	GPT4 ChatGPT % Win	GPT4 GPT4 % Win
GPT 3.5 Turbo 0301	89.37	50.00	-
UltraLM 13B V2.0 (PPO)	86.30	-	-
LLaMA2 Chat 13B (PPO)	81.09	-	-
Tulu 2 13B (SFT)	78.90	-	-
SFT + p	80.64	53.27	17.59
Base + p	79.61	51.26	22.11
Offline RL training on Base with various sampling methods			
Rand + p	82.60	59.05	30.81
Rand	80.40	49.75	23.37
PRand	85.07	64.32	39.20
PRS	**86.89**	**72.36**	**43.22**

Table 3: Results of AlpacaEval v1.0. We use the common preference in Table 1 to control our models to generate responses. "+ p" adds preference in the input during testing. *SFT* uses all available labeled data of ShareGPT and Alpaca-GPT4 for supervised fine-tuning. *Base* is the model tuned using ShareGPT data. To reduce the cost of calling GPT-4, we downsampled the test set for ChatGPT and GPT-4 baseline. We also show existing models tuned from Llama-2-13B for comparison, but they are fine-tuned with full parameters and different training data.

- ***PRS*** $(N/2, N/2)$ ***w/o*** $\boldsymbol{f}$ omits generating language feedback $\boldsymbol{f}$ during refinement and instead uses $\pi_\theta(\boldsymbol{y}_1|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*)$. The goal is to assess the impact of language feedback.

Policy Models. We use the model tuned on the SFT data from ShareGPT named *Llama-2-13B + SFT* to sample responses. We also test multiple open-source instruction-following models such as those tuned on Mistral-7B (Jiang et al., 2023) and Llama-2-13b (Touvron et al., 2023).

Preference $\boldsymbol{z}$. For instruction following, we aim to evaluate the common preference (as shown in Table 1) that favors comprehensive and detailed responses. As shown by Sun et al. (2023), a more detailed response would improve the performance on benchmarks such as AlpcaEval (Li et al., 2023). Since the reward model UltraRM-13B that we use internally includes such preferences, we compute $R(\boldsymbol{x}, \boldsymbol{y})$ without explicitly specifying $\boldsymbol{z}$.

Results. From the results shown in Fig. 4, *PRS* generates data with higher rewards than Rand and PRand, and as N increases, the performance gap becomes larger. The setting $(N/2, N/2)$ is much better than $(0, N)$ and $(N, 0)$, showing that a good balance of exploration is important. Fig. 4 (middle) shows that *PRS* produces a normal distribution

	R-1	R-2	R-L	Avg.
LLaMA2 Chat 13B	32.93	10.70	29.29	24.31
Mistral 7B v0.2	34.98	11.27	31.38	25.88
Tulu 2+DPO 13B	36.64	12.93	33.34	27.64
Vicuna 13B V1.5	37.12	13.26	33.71	28.03
Base w/o keywords	30.15	10.35	27.89	22.80
Base + p	35.46	12.56	32.37	26.80
RL training on un-tuned Llama-2-13B				
PRand	$37.39^\dagger$	$13.71^\dagger$	$33.96^\dagger$	$28.35^\dagger$
PRS	$\mathbf{38.20}^*$	$\mathbf{14.16}^*$	$\mathbf{34.70}^*$	$\mathbf{29.02}^*$
Continual RL training on Base				
PRand	$37.50^\dagger$	$13.78^\dagger$	$34.12^\dagger$	$28.47^\dagger$
PRS	$\mathbf{38.15}^*$	$\mathbf{14.16}^*$	$\mathbf{34.65}^*$	$\mathbf{28.99}^*$

Table 4: Summarization results on CNN / Daily Mail, adding input keywords except for the "Base w/o keywords" condition. We report average Rouge-1, Rouge-2, and Rouge-L F1 scores with 5 runs. $*$ indicates *PRS* outperforms PRand significantly ($p < 0.01$), and † indicates PRand outperforms Vicuna 13B V1.5 ($p < 0.01$).

with higher mean and variance than PRand and Rand, indicating a broader exploration and higher reward acquisition in the sampling space. From the full results shown in Fig. 9, language feedback shows mixed results: some models improve, while others do not. However, language feedback increases transparency and both versions still outperform other baselines.

PRand is substantially better than Rand, since PRand adds explicit preference in the input. It demonstrates that preference is effective in guiding the generation of better-aligned responses. For summarization, specifying the keywords would aid the model to concentrate on the key information of the document. The greedy algorithm, revising based on the current best response, often underperforms compared to *PRS*. Its main limitation is poor response exploration. In contrast, *PRS* $(N/2, N/2)$ excels by thoroughly exploring both initial and subsequent responses.

We further investigate best-of-N sampling on AlpacaEval v2.0 and Arena-Hard v0.1. The models are evaluated as outlined in Table 2. To obtain the reward scores, we utilize the recent state-of-the-art reward model, ArmoRM-Llama3-8B-v0.1. For *PRS*, no preference is specified, and feedback generation is omitted during sampling to support more general use cases. We employ two layers in *PRS*, with each layer having a width of $N/2$. As shown in Table 2, *PRS* consistently outperforms repeated random sampling, achieving better performance in LC WR on AlpacaEval and WR on Arena-Hard.

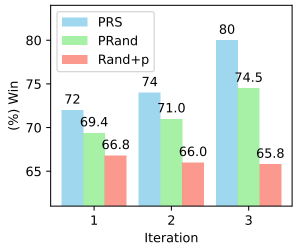


Figure 5: Offline RL training: Win rates for *PRS*, PRand, and Rand + p vs. Base + p, evaluated using GPT-4 on a 200-sample AlpacaEval. "+ p" adds common preference in the input.

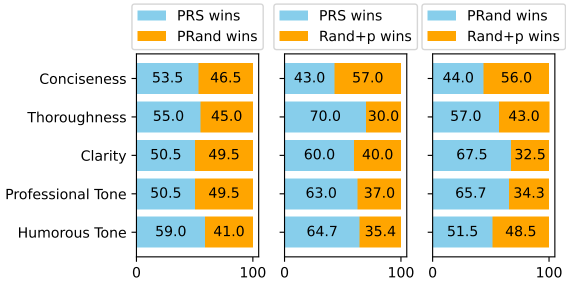


Figure 6: Preference Adaptation: We define five preference categories and evaluate each category using 100 AlpcaEval test cases. For each category, we customize the prompt (100 test samples) by appending the corresponding preference, evaluating with GPT-4, and recording win rates (%) when comparing two models.

ToxiGen	% Toxic (↓)
GPT-4-0613	0.6
GPT-3.5-turbo-0613	0.5
GPT-3.5-turbo-0301	27.7
Zephyr 7B Beta	64.0
Xwin-LM v0.1 70B	12.7
Tulu 2+DPO 13B	1.1
Rand	3.9
Rand + p	0.3
PRand	0.2
PRS	0.2

Table 5: Toxicity reduction. We append a preference indicating a safe response in the input for Rand + p, PRand, and *PRS*.

5.2 Offline RL Training

We conduct offline RL training to align the models to generate responses tailored to input preferences.

Experimental Settings. We fine-tune the Llama-2-13B model using LoRA (Hu et al., 2022), starting with supervised fine-tuning (SFT) using labeled data. For instruction following, we perform 3 iterations of RL training, each involving 10k unique GPT-4 prompts. We adopt best-of-16 sampling, generating 16 responses per prompt, and adding 10k new training data per iteration. We set $N_0 = N_1 = 8$ for *PRS*. For summarization, after the initial SFT, we undertake one RL iteration, sampling 64 summaries per document (2,500 in total), retaining the summary with the highest reward for each document. We set $N_0 = N_1 = 32$ for *PRS*.

Results. Results of AlpacaEval and CNN/Daily Mail are reported in Tables 3 and 4 respectively.

The model trained by *PRS* outperforms those trained by PRand and Rand. Looking at the rewards of the generated training data shown in Fig. 12 in the Appendix, *PRS* exhibits consistently higher rewards than PRand. It shows that the quality of data generation is key to offline RL. Compared to open-source models, *PRS* outperforms the models tuned by PPO training. In head-to-head comparison shown in Fig. 11 in the Appendix, *PRS* outperforms multiple strong open-source models more than 50% of the time, except for Mistral-7B-v0.2. These promising results highlight the potential of *PRS* for future applications, such as integrating *PRS* with DPO training (Rafailov et al., 2023) and full-parameter fine-tuning. For summarization, after aligning the model with *PRS*, our model performs the best among existing strong open-source models.

Preference-controlled optimization during training is important. The method Rand + p involves adding a preference to the input prompt at test time. It effectively enhances performance compared to Rand. However, it does not explicitly optimize the response to the input preference during training compared to PRand, so it underperforms PRand

We further present the results of RL training for each iteration in Fig. 5. Our findings indicate that while using random sampling (Rand) leads to a halt in improvement after just one iteration of RL training, both PRand and *PRS* continue to show improvement across 3 training iterations. The quality of data generated through random sampling can significantly influence the iterative updates made to the model. Since the generated data is of lower quality, it can lead to a degradation in the model's performance. This, in turn, makes it increasingly challenging for the model to generate high-quality data, thereby halting further improvements.

5.3 Further Analysis

Preference Adaptation. We further compare *PRS*, PRand, and Rand + p on adaptation to personalized preferences differing from the common preference studied in Fig. 4 (left) and Table 3. We define five categories as shown in Fig. 6 for adaptation and for each category, we create 20 unique expressions using GPT-4. We evaluate them across 100 AlpacaEval test cases. For each category, we randomly sample an expression and append it to the prompt. More details can be found in Appendix C.

PRS outperforms PRand, especially in delivering concise, thorough, and humorous responses. Both models perform similarly in clarity and professional tone. Overall, both *PRS* and PRand surpass Rand + p in effectiveness, showing the benefits of training models to align with user pref-

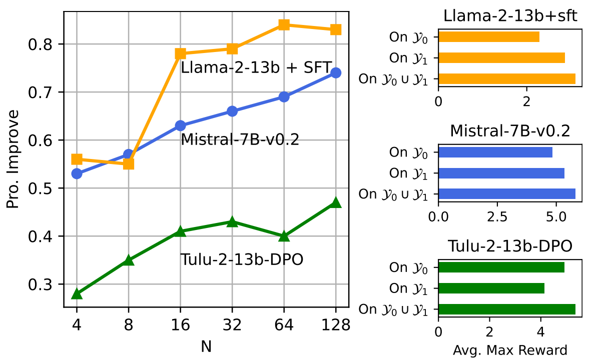


Figure 7: Proportion of cases where the top response from N_1 refinements in $\mathcal{Y}_1$ yields a higher reward than the best initial response from $\mathcal{Y}_0$. Average maximum rewards for each set and their union are reported (N=32).

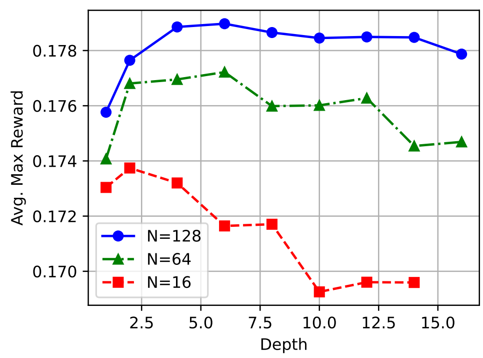


Figure 8: Effects of varying depth and width for *PRS*. We maintain the number of samples N and vary the depth d and the width w calculated by $\lfloor \frac{N}{d} \rfloor$. The depth starts from 1 to 16. Preference is not included in the input and feedback is not generated. Here, the studied model is Llama-3-8b-instruct and the reward model is ArmoRM-Llama3-8B-v0.1. 100 test samples are randomly selected from AlpacaEval for evaluation.

erences. However, Rand + p excels in conciseness, producing fewer tokens (176.07) compared to *PRS* (199.31). In contrast, for thoroughness, while Rand + p averages 378.99 tokens, PRand and *PRS* provide more thorough responses with 461.34 and 507.81 tokens, respectively.

Toxicity Reduction. We further study toxicity reduction as preference adaptation. For each input, we append a safe preference after it, which is randomly sampled from a pool of safe preferences with different expressions (see Table 6). We evaluate ToxiGen (Hartvigsen et al., 2022) and report the results in Table 5. Compared to Rand and Rand + p, adding a safe preference can substantially reduce the generation of toxic content. PRand and *PRS* achieve comparable performance and both outperform Rand + p. Preference-controlled alignment adapts the LLM to generate safe and harmless content at test time, even without explicit training for safety.

Tree-Based Generation. We analyze tree-based generation in *PRS*, which starts with N_0 initial responses ($\mathcal{Y}_0$), and then N_1 refinements ($\mathcal{Y}_1$). We evaluate how often refinements improve over the initial response. As shown in Fig. 7, there is variability across models: Tulu-2-13b-DPO improves less than 50% of the time, while Mistral-7B-v0.2 and Llama-2-13B + SFT perform better. Improvement rates generally increase with more samples (N), indicating that more samples can lead to better outcomes. We explore the reward values for $\mathcal{Y}_0$ and $\mathcal{Y}_1$. We find that $\mathcal{Y}_1$ does not consistently offer higher rewards than $\mathcal{Y}_0$, but combining both sets yields higher rewards.

Expansion in *PRS*. Here, we examine how the depth and width impact the performance of tree-based generation in *PRS*. We keep the total number of samples N constant while varying the depth d. The width w is then calculated by $\lfloor \frac{N}{d} \rfloor$. As shown in Fig. 8, our results indicate that for larger N, increasing the depth (e.g., to 4) improves performance. However, for smaller values of N, such as 16, increasing the depth beyond 2 does not yield further benefits. A larger N results in a greater width, allowing the model to sample more responses at each layer, thereby increasing the likelihood of discovering better responses than those in the previous layers. We further conduct an ablation study in Appendix B.3.

6 Conclusion

We introduce *PRS*, an improved sampling method designed to enhance iterative model improvement. In contrast to repeated random sampling, *PRS* enables more efficient generation through a tree-based approach. By allowing the specification of preference in the input, *PRS* optimizes responses to better align language models with diverse user preferences. Our comprehensive evaluation shows that *PRS* consistently generates higher-quality samples. On AlpacaEval and Arena-Hard, *PRS* significantly outperforms random sampling in the best-of-N setting. Additionally, *PRS* excels when applied to iterative offline RL training.

7 Limitations

Our approach capitalizes on the model's self-improvement capabilities to aid in data sampling.

However, for more challenging tasks, such as reasoning tasks, the model may struggle to enhance its performance autonomously. We have not explored these types of tasks in this work. Further enhancing the model's self-improvement capabilities, particularly for more difficult tasks, can be explored in the future. Our approach may be susceptible to reward hacking, though further research may mitigate its effects.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2021-08-016[T]). We thank Ruochen Xu for his comments on this paper, and the anonymous reviewers for their valuable suggestions.

References

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

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

Bradley C. A. Brown, Jordan Juravsky, Ryan Saul Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Yihan Chen, Benfeng Xu, Quan Wang, Yi Liu, and Zhendong Mao. 2024. Benchmarking large language models on controllable generation under diversified instructions. In *AAAI*.

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. *CoRR*, abs/2310.01377.

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. *TMLR*.

Xidong Feng, Ziyu Wan, Muning Wen, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training. *CoRR*, abs/2309.17179.

Çaglar Gülçehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. Reinforced self-training (ReST) for language modeling. *CoRR*, abs/2308.08998.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *ACL*.

Qianyu He, Jie Zeng, Qianxi He, Jiaqing Liang, and Yanghua Xiao. 2024. From complex to simple: Enhancing multi-constraint complex instruction following ability of large language models. *CoRR*, abs/2404.15846.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *CoRR*, abs/2406.11939.

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. https://github.com/tatsu-lab/alpaca_eval.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *ACL Workshop on Text Summarization Branches Out*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*.

OpenAI. 2022. Introducing ChatGPT. OpenAI Blog. https://openai.com/blog/chatgpt.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *CoRR*, abs/2304.03277.

Ethan Perez, Saffron Huang, H. Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *EMNLP*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*.

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. *CoRR*, abs/2303.16755.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*.

Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2024. SALMON: self-alignment with instructable reward models. In *ICLR*.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. In *NeurIPS*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *CoRR*, abs/2406.12845.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating sequences by learning to self-correct. In *ICLR*.

Shunyu Yao, Howard Chen, Austin W Hanjie, Runzhe Yang, and Karthik Narasimhan. 2024. Collie: Systematic construction of constrained text generation tasks. In *ICLR*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *ICLR*.

Algorithm 2 : *PRS* for aligning language models for diverse preferences

Input: Labeled training data $\mathcal{D}_0$; K sets of unlabeled data $[\mathcal{U}_1, \cdots, \mathcal{U}_K]$; large language model π_θ; reward model R; number of samples per prompt N; N_0.
Initialize π_{θ_0} on $\mathcal{D}_0$ using Eq. 1.
$\mathcal{D} \leftarrow \emptyset$.
for $k = 1$ to K **do**
  # Stage 1: Data Generation
  $\mathcal{D}_k \leftarrow \emptyset$.
  for all $(\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{U}_k$ **do**
    # Preference-Guided Reflective Sampling (PRS)
    • Sample N_0 responses $\mathcal{Y}_0 \sim \pi_{\theta_{k-1}}(\boldsymbol{y}_0|\boldsymbol{x}, \boldsymbol{z})$.
    Maximize reward $R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^i)$ over $\mathcal{Y}_0$ to find the optimal $\boldsymbol{y}_0^*$.
    • Generate language feedback $\boldsymbol{f} \sim \pi_{\theta_{k-1}}(\boldsymbol{f}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*)$.
    • Sample $N_1 = N - N_0$ refinements $\mathcal{Y}_1 \sim \pi_{\theta_{k-1}}(\boldsymbol{y}_1|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*, \boldsymbol{f})$.
    Maximize reward $R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_1^i)$ over $\mathcal{Y}_1$ to find the optimal $\boldsymbol{y}_1^*$.
    if $R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_1^*) > R(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*)$ **then**
      Add $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_1^*)$ and $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*, \boldsymbol{f}, \boldsymbol{y}_1^*)$ into $\mathcal{D}_k$.
    else
      Add $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0^*)$ into $\mathcal{D}_k$.
    end if
  end for
  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_k$.
  # Stage 2: Model Re-training
  Update π_{θ_k} on $\mathcal{D} \cup \mathcal{D}_0$ with Eq. 5.
end for
Output: $\pi_{\theta_K}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$.

A Detailed Related Work

A.1 Alignment of Large Language Models

Similar to *PRS*, Bai et al. (2022b) also leverage the LLM's capacity for reflection to refine model responses. However, our work differs from Bai et al. (2022b) in several aspects: (a) Most importantly, Bai et al. (2022b) do not aim to improve data sampling for RLHF, but our work proposes a tree-based framework to enable efficient data generation. (b) Bai et al. (2022b) only focus on harmless responses, but our work applies to a broader spectrum of preferences. (c) While the preferences added into the input to guide model generation – introduced in our work – is similar to the usage of principles proposed by Bai et al. (2022b), their approach is limited to modifying responses based on principles rather than integrating these principles into the input prompt to guide the generation of model responses.

Sun et al. (2024) propose to train a reward model that can evaluate responses based on principles, which is similar to our work when using the reward model by adding the extra preference information. However, Sun et al. (2024) also overlook the importance of sampling efficiency. Another notable contribution is from Scheurer et al. (2023), who advocate for training models using language feedback, as opposed to the numerical feedback derived from reward models. Unlike our strategy, which employs the model itself to generate language feedback, they depend on human annotators for this task. Recent work by Feng et al. (2023) aligns with our goal to enhance model sampling exploration. They adopt Monte Carlo tree search (MCTS) for decoding, utilizing token-level rewards to guide output sampling in instruction-following tasks. In contrast, our approach prioritizes sequence-level rewards for evaluating model responses and employs a tree-based search without extensive inference costs. Furthermore, we incorporate input prompt preferences to direct the generative process, which is another difference from Feng et al. (2023).

A.2 Reflective Reasoning of Large Language Models

Large language models (LLMs) have demonstrated self-reflection capability, critically analyzing their own decisions and providing feedback to enhance their responses (Madaan et al., 2023). Madaan et al. (2023) introduce a self-refinement framework that enables LLMs to continuously improve their responses based on self-generated feedback. In contrast, our work introduces an efficient tree-based generation model that optimizes the use of LLMs' reflective abilities more effectively. Further exploring the potential of LLMs' self-reflective capabilities, Shinn et al. (2024) leverage this feature to enable LLMs to learn from language-based feedback and refine their outputs towards more accurate and contextually relevant responses. In the realm of iterative inference for text generation, Welleck et al. (2023) propose training a corrector model to refine LLM outputs, utilizing synthetically generated data that fosters gradual improvement. The concept of reflection in LLMs is crucial for advances of AI agents, facilitating their ability to summarize and reflect on outcomes from previous interactions to better plan and execute future actions (Yao et al., 2023a,b).

A.3 Controlled Instruction Following

In the era of large language models, there is growing interest in evaluating and enhancing complex instruction following with the outputs controlled by input constraints (Chen et al., 2024; He et al., 2024; Yao et al., 2024). In our work, to improve sampling efficiency, we frame generation as a problem of controlled text generation by treating user preference as the constraint.

B Additional Results

B.1 Full Results of Data Sampling

We show the full results of data generation on 9 policy models in Fig. 9.

B.2 Instruction Following

Head-to-head comparison of *PRS* and PRand after 3 iterations of RL training is shown in Fig. 10.

B.3 Ablation Study

In our ablation study, we evaluate the impact of removing the optimization of $p(\boldsymbol{y}_1|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_0, \boldsymbol{f})$ during *PRS* RL training. We apply tree-based generation after each RL iteration to sample outputs. As shown in Fig. 13, skipping this optimization results in lower rewards. Additionally, the performance declines in the ablated baseline starting from the third RL iteration, highlighting the importance of this component.

C Details of Preference Adaptation

We assess the adaptability of models to input preferences by specifying a desired output preference after the input question. Our comparison focuses on the methods of *PRS*, PRand, and Rand, examining how different sampling approaches influence the alignment of models with user preferences. Both *PRS* and PRand are designed to train models explicitly to adhere to given preferences, unlike Rand which does not incorporate preference training.

Given the impracticality of evaluating every conceivable preference, we define five types of preferences which are *conciseness*, *thoroughness*, *clarity*, *professional tone*, and *humorous tone*. To ensure robust generalization, we use GPT-4 to formulate 20 unique expressions for each preference category. Each preference within a category conveys the same underlying preference but is expressed in different styles. We present one expression for each category as follows:

- **Conciseness**: I prefer short and sweet responses.
- **Thoroughness**: I prefer responses that delve deeply into the topic and provide comprehensive insights.
- **Clarity**: I prefer responses that are clear and straightforward, avoiding any unnecessary complexity or jargon.
- **Professional Tone**: I prefer responses that maintain a professional tone throughout the conversation.
- **Humorous Tone**: I prefer responses with a side of humor, please.

Sample preferences are shown in Table 7.

We sample 100 test cases from AlpacaEval for evaluation. We conduct a head-to-head comparison between the models tuned by *PRS*, *PRand*, and Rand. For each category, we randomly sample an expression from 20 candidates and append the preference after the input prompt when generating a response. We also use GPT-4 as the evaluator, adapting the evaluation code from AlpacaEval. We specifically indicate a preference type for GPT-4 to choose a better-aligned response from two candidates.

D Prompts used in *PRS*

For instruction following, we present the prompt used for generating feedback in Fig. 14, the prompt

	Safe and Harmless Preferences
1	I prefer a response that is considerate, safe, and avoids any sensitive or potentially harmful content, without causing any offense or discomfort.
2	I prefer a response that is mindful, secure, and steers clear of any sensitive or potentially damaging material, ensuring it doesn't cause offense or discomfort.
3	I prefer a response that is thoughtful, safe, and carefully avoids any sensitive or potentially harmful subjects, ensuring no offense or discomfort is caused.
4	I prefer a response respectful, cautious, and deliberately evades any delicate or potentially injurious topics, so as to prevent causing any upset or unease.
5	I prefer a response that's thoughtful and secure, meticulously avoiding topics that might be sensitive or harmful, thereby ensuring nobody feels offended or uncomfortable.

Table 6: The safe and harmless preferences used for toxicity reduction.

for revising the response with feedback in Fig. 15, and the prompt for direct refinement without feedback in Fig. 16. For keyword-focused document summarization, the prompt for providing feedback is shown in Fig. 17 and the prompt for refinement with feedback is shown in Fig. 18.

E Details of Preference Generation

List of Professions. We have 222 professions in total divided into different categories:

- **Information Technology:** Software Developer, System Administrator, Network Engineer, Database Administrator, IT Support Specialist, Security Analyst, Data Scientist, Business Analyst, IT Project Manager, Web Developer, Quality Assurance Analyst, DevOps Engineer, Cloud Solutions Architect, IT Consultant, UX/UI Designer, Technical Writer, Mobile App Developer, Data Analyst, Information Systems Manager, and Computer Systems Analyst.
- **Business:** Accountant, Financial Analyst, Marketing Manager, Sales Representative, Business Analyst, Operations Manager, Human Resources Specialist, Management Consultant, Entrepreneur, Product Manager, Project Manager, Supply Chain Analyst, Customer Service Representative, Business Development Manager, and Data Analyst.
- **Retail:** Cashier, Sales Associate, Store Manager, Assistant Store Manager, Retail Merchandiser, Customer Service Representative, Stock Clerk, Visual Merchandiser, Loss Prevention Officer, Department Manager, Buyer, Inventory Control Specialist, Store Owner, E-commerce Specialist, and Retail Sales Consultant.
- **Health and Social Work:** Doctor, Nurse, Social Worker, Physical Therapist, Occupational Therapist, Dentist, Pharmacist, Clinical Psychologist, Counselor, Healthcare Administrator, Medical Laboratory Technician, Home Health Aide, Radiologic Technologist, Dietitian, Speech-Language Pathologist, Medical Assistant, Public Health Specialist, Chiropractor, Optometrist, Mental Health Technician, and Health Educator.
- **Transportation:** Truck Driver, Delivery Driver, Bus Driver, Taxi Driver, Pilot, Flight Attendant, Railway Conductor, Train Operator, Ship Captain, Sailor, Air Traffic Controller, Logistics Coordinator, Supply Chain Manager, Freight Agent, Transportation Planner, Transportation Engineer, Bicycle Courier, Warehouse Worker, Forklift Operator, and Aircraft Maintenance Technician.
- **Writing and Creative Arts:** Author, Screenwriter, Journalist, Editor, Copywriter, Content Creator, Blogger, Playwright, Poet, Graphic Designer, Illustrator, Animator, Photographer, Videographer, Filmmaker, Actor, Director, Producer, Musician, Composer, Visual Artist, Sculptor, Painter, Dancer, Choreographer, and Performance Artist.
- **Broadcasting and Entertainment:** Actor, Director, Producer, Screenwriter, Cinematographer, Film Editor, Broadcast Journalist, Television Presenter, Radio Presenter, News Anchor, Camera Operator, Sound Engineer, Lighting Technician, Production Designer, Makeup Artist, Costume Designer, Animator, Visual Effects Artist, Music Composer, Singer, Musician, Stand-up Comedian, Talent Manager, Casting Director, and Stage Manager.
- **Law and Order:** Lawyer, Paralegal, Judge, Police Officer, Correctional Officer, Detective, Prosecutor, Public Defender, Legal Assistant, Bailiff, Criminologist, Forensic Scientist, Court

Reporter, Private Investigator, Legal Secretary, Probation Officer, Court Clerk, Security Guard, Prison Warden, and Compliance Officer.

- **Sports and Recreation:** Athlete, Coach, Sports Agent, Physical Therapist, Personal Trainer, Referee/Umpire, Sports Journalist, Sportscaster, Fitness Instructor, Recreation Worker, Athletic Trainer, Sports Photographer, Sports Marketing Specialist, Sports Psychologist, Sports Nutritionist, Gym Manager, Outdoor Activity Coordinator, Sports Statistician, Team Manager, and Scout.
- **Education:** Teacher, School Principal, School Counselor, Librarian, Teaching Assistant, Education Administrator, Instructional Coordinator, Special Education Teacher, University Professor, Tutor, Educational Consultant, College Admissions Officer, Academic Advisor, School Psychologist, Education Policy Analyst, Curriculum Developer, Education Researcher, Literacy Coach, Physical Education Teacher, and ESL Teacher.
- **Scientific Research:** Research Scientist, Laboratory Technician, Research Assistant, Data Analyst, Statistician, Biologist, Chemist, Physicist, Biochemist, Clinical Research Associate, Epidemiologist, Environmental Scientist, Geneticist, Microbiologist, Astrophysicist, Geologist, Postdoctoral Researcher, Principal Investigator, Research Fellow, and Scientific Writer.

Preference Annotation. We use GPT-3.5-turbo to generate the preferences. For each prompt from Alpaca-GPT4, we use the template in Fig. 19 to generate the preference, where the generation is conditioned on the question and a profession name. The profession name is randomly selected from the profession name list. After obtaining a preference, we further prompt GPT-3.5-turbo to revise its output to make the generated preference general and applicable to different questions. In Fig. 20, we present a variety of generated preferences, illustrating the diversity in the preferences that the method can produce.

F Sample Outputs of Different Baselines

We display sample outputs in Tables 8 and 9.

	Conciseness
1	I prefer short and sweet responses.
2	I prefer answers that are to the point.
3	I prefer concise explanations, no fluff.
	Thoroughness
1	I prefer responses that delve deeply into the topic and provide comprehensive insights
2	I prefer when the information is thorough and covers all aspects, leaving no stone unturned.
3	I prefer a detailed exposition, with rich context and nuanced explanations.
	Clarity
1	I prefer responses that are clear and straightforward, avoiding any unnecessary complexity or jargon.
2	I prefer that you explain things simply, as if you were talking to someone who's completely new to the topic.
3	I prefer answers that are easy to understand and follow, without any convoluted explanations.
	Professional Tone
1	I prefer responses that maintain a professional tone throughout the conversation.
2	I prefer that the language used is formal and professional in nature.
3	I prefer the communication to be strictly professional.
	Humorous Tone
1	I prefer responses with a side of humor, please.
2	I prefer my information served with a chuckle.
3	I prefer answers that come with a comedic twist.

Table 7: Sample preferences with different expressions for each category. Three examples are shown in each category.

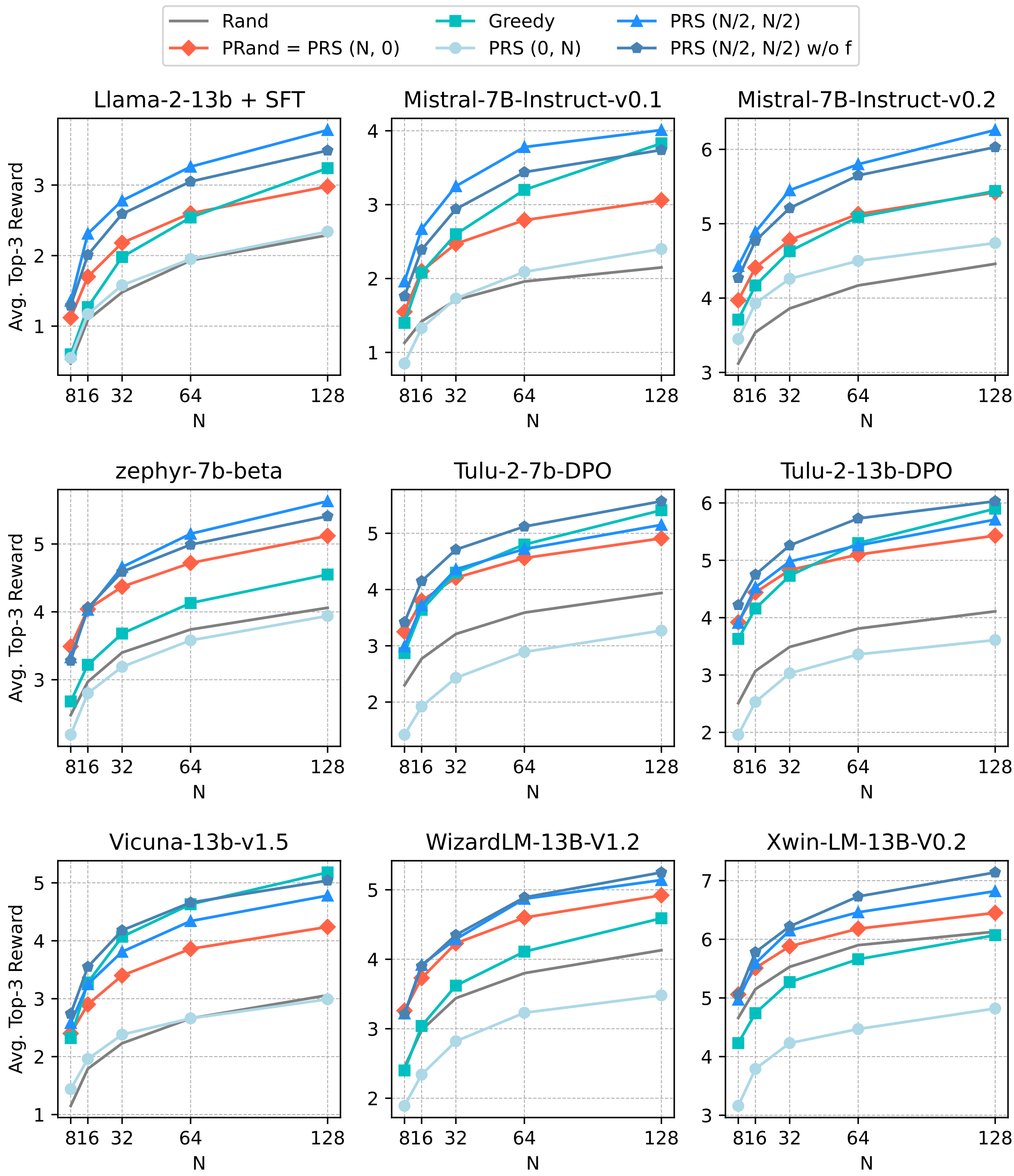


Figure 9: **Results of data generation for instruction following**: We focus on the common preference and use the description in Table 1 to generate detailed and in-depth responses. (a) **Policy Models**: We use 9 policy models to generate training data, which are Llama-2-13b + SFT, Mistral-7B-Instruct-v0.1, Mistral-7B-Instruct-v0.2, zephyr-7b-beta, Tulu-2-7b-DPO, Tulu-2-13b-DPO, Vicuna-13b-v1.5, WizardLM-13B-V1.2 and Xwin-LM-13B-V0.2.
(b) **Test samples**: We randomly sample 100 prompts from Alpaca-GPT4.
(c) **Setup**: We sample N responses per prompt (i.e., 8, 16, 32, 64, or 128) using a specific sampling method. We then average the top three rewards for each prompt, leading to an overall average score for the entire evaluation set. We use UltraRM-13B to generate the reward.

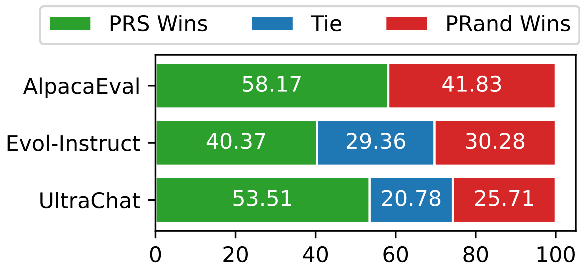


Figure 10: Head-to-head evaluation of *PRS* and PRand after 3 iterations of RL training. We use GPT-4 as the evaluator.

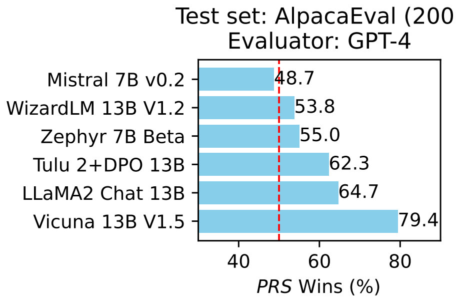


Figure 11: *PRS* vs. open-source models.

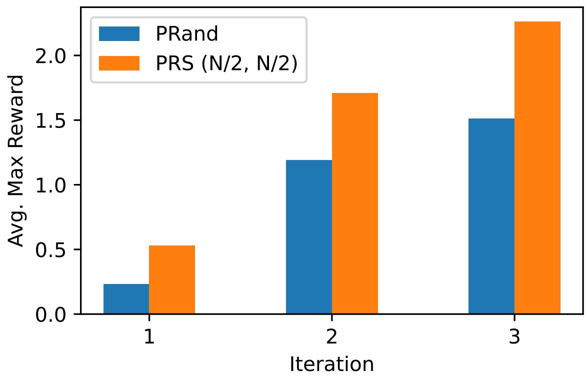


Figure 12: Average rewards of training data for personalized preferences during RL training. 10k prompts from Alpaca-GPT4 are used for sampling, each has a different preference exemplified by Table 1.

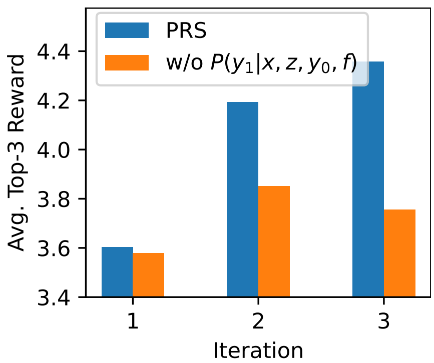


Figure 13: We ablate to exclude the optimization of $p(\mathbf{y}_1|\mathbf{x}, \mathbf{z}, \mathbf{y}_0, \mathbf{f})$ and use tree-based generation after each RL iteration (N=16), focusing on instruction following with the common preference in Table 1.

```
Please review the AI assistant's response to the user question presented below,
acting as an impartial judge. Your objective is to assess how well the reference
answer aligns with the user's preferences and suggest improvements. Structure
your feedback in bullet points for clarity and conciseness. Each point should
specifically reference a part of the reference answer, highlighting how it can be
better tailored to meet the user's expectations.

[Question]
{question}

[Start of the Reference Answer]
{answer}

[End of the Reference Answer]

[User Preference]
{preference}

Your feedback:
```

Figure 14: Prompt template for feedback generation, for the task of instruction following.

```
You are a skilled corrector tasked with enhancing a reference answer based on
specific user feedback. Your role is to refine the existing response, ensuring it
aligns more closely with the user's suggestions for improvement. Utilize the
feedback effectively to upgrade the reference answer, making it more relevant and
satisfactory to the user's expectations.

[Question]
{question}

[Start of the Reference Answer]
{answer}

[End of the Reference Answer]

[User Preference]
{preference}

[Feedback]
{feedback}

The improved answer (only generate the content that is relevant to the user
question):
```

Figure 15: Prompt template for refinement with feedback, for the task of instruction following.

```
You are a skilled corrector tasked with enhancing a reference answer. You need to
improve the reference answer to make it better align with the user's preference.

[Question]
{question}

[Start of the Reference Answer]
{answer}

[End of the Reference Answer]

[User Preference]
{preference}

The improved answer (only generate the content that is relevant to the user
question):
```

Figure 16: Prompt template for direct refinement without feedback, for the task of instruction following.

Kindly evaluate the AI assistant's summarization of the provided article. Your task is to impartially judge the conciseness and relevance of the summary, ensuring it adheres to the specific user preference of being strictly within 3 sentences and focused on the designated keywords. Please provide your feedback in bullet points for clarity. In your points, reference specific parts of the AI's summary and the article, suggesting precise improvements to better align with the user's expectations for a keyword-focused summary.

[Article]
{passage}

[Summary]
{summary}

[User Preference]
1. A summary strictly within 3 sentences. 2. Focus on keywords of {keyword}.

Provide your feedback in bullet points for clarity. In your points, reference specific parts of the AI's summary and the article, suggesting precise improvements to better align with the user's expectations for a keyword-focused summary.

Your feedback:

Figure 17: Prompt template for feedback generation, for the task of summarization.

As an expert in topic-focused summarization, your task is to refine a summary based on detailed user feedback. Focus on incorporating the user's preferences as below for a concise, three-sentence summary that emphasizes specific keywords. Use the provided feedback to enhance the clarity, relevance, and precision of the summary, ensuring it closely aligns with the user's expectations. Your goal is to modify the existing summary into a more effective and targeted summary that meets the user's preference.

[Article]
{passage}

[Original Summary]
{summary}

[User Preference]
{preference}

[User Feedback]
{feedback}

The improved summary (only generate the content that is relevant to the summary):

Figure 18: Prompt template for refinement with feedback, for the task of summarization.

First round:

Suppose you are a user using an AI model, and you have a specific preference for the model's response. Based on the following profession and the asked question, suggest one preference. You have to clearly describe your preference.

Return the preference in a JSON format: {{"preference": the description of the preference}}.
[Profession]
{profession}

[Question]
{question}

Second round:

You have to suggest a general preference that is applicable to other questions.

Figure 19: Prompt template for generating preferences, for the task of instruction following.

1. I prefer the AI model to provide well-researched and evidence-based information in its response.

2. I prefer the model to provide a concise and accurate answer without any unnecessary details or explanations.

3. I prefer clear and concise responses that provide accurate and relevant information without any unnecessary technical jargon or excessive details. Additionally, I appreciate if the response is well-organized and easy to understand.

4. I prefer the model to provide concise and well-structured responses, with clear and coherent explanations. The responses should be informative and relevant, addressing the specific question asked. Additionally, I value the model's ability to provide accurate and up-to-date information, supported by reliable sources if applicable.

5. I prefer a response that is concise, accurate, and provides practical solutions or actionable steps.

6. I prefer concise and well-structured responses that provide accurate and relevant information. Please avoid unnecessary repetition, irrelevant details, or excessive technical jargon. Additionally, it would be helpful if the response includes clear examples or supporting evidence to enhance understanding.

7. I prefer the AI model to provide concise and accurate responses with a friendly and conversational tone.

8. I prefer the AI model to provide concise and clear responses that directly address the question asked. It should avoid providing excessive or irrelevant information.

Figure 20: Sample annotated preferences by GPT-3.5-turbo.

	Do dinosaurs really exist?
Rand	Yes, dinosaurs are a group of extinct reptiles that lived on Earth during the Mesozoic Era, which lasted from about 250 million to 66 million years ago. The term "dinosaur" was coined by Sir Richard Owen in 1842, and it means "terrible lizard." There is overwhelming evidence that dinosaurs existed, including fossils found all over the world, as well as scientific research and analysis of these fossils. Paleontologists have discovered thousands of different species of dinosaurs, ranging in size from tiny, bird-like creatures to massive giants like Tyrannosaurus rex. While there is no doubt that dinosaurs existed, there is still much that we don't know about them, such as how they behaved, what their environments were like, and why they went extinct. However, through continued research and discovery, scientists are constantly learning more about these fascinating creatures.
	Do dinosaurs really exist? I prefer responses that are informative, precise, creative, detailed, relevant, and in-depth.
Rand + p	Yes, dinosaurs existed. They were a group of animals that lived on Earth during the Mesozoic era, which lasted from about 250 million to 66 million years ago. The first known dinosaurs appeared around 230 million years ago and went extinct at the end of the Cretaceous period, about 66 million years ago. There are many different types of dinosaurs, including Tyrannosaurus rex, Triceratops, Velociraptor, Stegosaurus, Diplodocus, and Brachiosaurus. These animals ranged in size from small, bipedal carnivores like Velociraptor to massive, quadrupedal herbivores like Apatosaurus. Some dinosaurs had feathers, while others had armor or spikes on their bodies. Dinosaur fossils have been found all over the world, including North America, South America, Africa, Europe, Asia, and Australia. These fossils provide evidence of what these animals looked like and how they lived. Scientists use this information to learn about the evolution of life on Earth and to understand the history of our planet.
PRand	Yes, the dinosaurs really existed. They were a diverse group of reptiles that lived on Earth during the Mesozoic Era, which lasted from about 251 to 66 million years ago. The first dinosaurs appeared in the Triassic period and went extinct at the end of the Cretaceous period. Dinosaurs were large, land-dwelling animals that varied greatly in size, shape, and behavior. Some were herbivores (plant-eaters) like the long-necked sauropods such as Brachiosaurus and Diplodocus, while others were carnivores (meat-eaters) like Tyrannosaurus rex and Velociraptor. Some dinosaurs could fly, like Pteranodon, while others could swim, like Spinosaurus. There have been many discoveries of fossilized remains of dinosaurs around the world, providing evidence of their existence. These include complete skeletons, tracks, eggs, nests, and even preserved soft tissues. Scientists use these fossils to study the evolution, anatomy, and behavior of dinosaurs, helping us understand more about life on Earth millions of years ago.
PRS	Yes, dinosaurs are a group of extinct reptiles that lived on Earth during the Mesozoic Era, which lasted from approximately 252 to 66 million years ago. They first appeared in the fossil record during the Triassic Period and became extinct at the end of the Cretaceous Period, along with many other species due to a massive asteroid impact. Dinosaurs were diverse and came in various shapes and sizes, ranging from small bird-like creatures to gigantic herbivores and carnivores. Some well-known dinosaurs include Tyrannosaurus rex, Triceratops, Stegosaurus, and Velociraptor. These animals had unique features such as horns, spikes, plates, and feathers that helped them adapt to their environment. The discovery of dinosaur fossils has played a crucial role in our understanding of the history of life on Earth. Paleontologists have found thousands of dinosaur fossils all over the world, providing valuable insights into how these creatures evolved over time and what kind of environments they inhabited. By studying dinosaur fossils, we can learn about the geological changes that occurred throughout the Mesozoic Era, including shifts in climate, sea level, and landmass distribution. Overall, there is overwhelming evidence supporting the existence of dinosaurs. Their remains have been found in rocks dated back millions of years ago, and their fossils show distinctive characteristics that differentiate them from other animals alive today. Therefore, it is reasonable to conclude that dinosaurs were real animals that once roamed our planet.

Table 8: Example 1 demonstrates outputs using different sampling methods to align the language model.

	Decide whether children can watch the given movie. Mention what makes it inappropriate if you think they can't. The Dark Knight (2008)
Rand	The Dark Knight is a superhero film that follows the story of Batman as he battles against the villainous Joker. While it does not contain any explicit content, the movie has a dark and intense tone that may be too mature for younger children. Additionally, some scenes feature violence and action sequences that could be frightening for young viewers. Overall, I would recommend this movie for teenagers or adults who are comfortable with its themes and imagery.
	Decide whether children can watch the given movie. Mention what makes it inappropriate if you think they can't. The Dark Knight (2008) I prefer responses that are informative, precise, creative, detailed, relevant, and in-depth.
Rand + p	The Dark Knight is a superhero film that follows the story of Batman as he tries to take down the criminal mastermind, the Joker. The movie has received critical acclaim for its performances, direction, and themes, but it also contains some mature content that may not be suitable for younger audiences. While children can certainly watch The Dark Knight, parents should exercise caution and consider their child's age before allowing them to do so. The movie features intense violence, language, and themes of death and tragedy that may be too mature for younger viewers. Additionally, the Joker's character is particularly disturbing and may frighten young children. Overall, while The Dark Knight is an excellent movie, parents should use their discretion when deciding whether or not to allow their children to watch it.
PRand	The Dark Knight is a superhero film based on the DC Comics character Batman. The movie follows Bruce Wayne, who assumes the identity of Batman to protect Gotham City from criminals. The film explores themes such as crime, corruption, and morality. It was rated PG-13 for intense sequences of violence and menace, some sexual content, and language. In my opinion, children can watch The Dark Knight with parental guidance. While the movie has some violent scenes, they are not overly graphic or gory. There is also some sexual content, but it is not explicit or gratuitous. The language used in the movie is also not too crude or offensive. However, parents should be aware that the movie deals with mature themes and may be too complex for younger audiences to fully understand. Additionally, the film's runtime is over two hours long, which could make it difficult for younger children to sit through without getting restless. Overall, I believe that older children (age 13 and above) can enjoy The Dark Knight with proper supervision and discussion with their parents about its themes and content.
PRS	The Dark Knight is a 2008 superhero movie based on the DC Comics character Batman. The film features intense violence, profanity, and mature themes that may not be appropriate for children. Violence: The Dark Knight contains numerous scenes of violent crime, including gunfire, explosions, and hand-to-hand combat. These scenes can be particularly disturbing for younger viewers who may not understand the context or consequences of such violence. Profanity: The film also includes frequent use of strong language, including profanities like \"f**k\" and \"sh*t.\" This type of language can be jarring for some parents who want to shield their children from hearing such words. Mature Themes: In addition to its violence and language, The Dark Knight deals with complex issues like terrorism, corruption, and moral ambiguity. These themes can be challenging for young children to process and may require additional discussion or explanation from parents after watching the movie. Overall, while The Dark Knight is an excellent film for older teenagers and adults, it may not be suitable for younger children due to its intense content and mature themes. Parents should exercise caution when deciding whether to allow their children to watch this movie.

Table 9: Example 2 demonstrates outputs using different sampling methods to align the language model.