000 EFFICIENT LLM ALIGNMENT VIA HIERARCHICAL 001 COARSE-TO-FINE REFINEMENT 002 003

Anonymous authors

004

010 011

012

013

014

015

016

017

018

019

021

025

026

027 028 029

Paper under double-blind review

ABSTRACT

Alignment of Large Language Models (LLMs) intends to make LLMs behave to satisfy human preferences and values. Widely used methods, e.g., Reinforcement Learning from Human Feedback (RLHF), usually involve the additional training of LLMs with a reward model or the dataset reflecting human preferences. However, these training-based methods cannot quickly adapt to different preferences. Recent methods leverage search during the decoding process to align LLMs with preferences. However, these methods ignore the influence of prompts on the decoding distribution, thus hindering the performance. In this work, we propose HCFR, a Hierarchical Coarse-to-Fine Refinement for efficient LLM alignment. Specifically, **HCFR** includes a two-stage refinement: i) *coarse refinement* which rephrases the prompts from users through self-refinement, and ii) fine refinement which leverages the search methods, *e.g.*, Monte Carlo Tree Search (MCTS), for the responses with the guidance of a pre-trained reward model. Experimental results on HH-RLHF and UltraChat demonstrate that HCFR can significantly outperform existing methods, e.g., ARGS, CARDS, and Rejection sampling, in terms of performance and efficiency, *i.e.*, achieving a 71.3% win-tie rate in GPT-4 evaluations while reducing time consumption by 42%.

INTRODUCTION 1

030 031 032 Large language models (LLMs) have demonstrated exceptional capabilities in various natural language processing tasks (Wei et al., 2022a; Bubeck et al., 2023; Touvron et al., 2023). Considering 033 their potential for broad societal impact, it is crucial that responses generated by LLMs align with 034 human preferences, such as being harmless, precise, and conforming to social norms and morals (Weidinger et al., 2021; Ganguli et al., 2022; Deshpande et al., 2023). Preference alignment has emerged as a critical area of study (Leike et al., 2018; Dai et al., 2023), aiming to ensure that LLMs 037 generate texts aligned with human preferences. Techniques such as Reinforcement Learning from Human Feedback (RLHF) have shown promising performance in improving the alignment of LLM 039 outputs (Ouyang et al., 2022; Bai et al., 2022b). However, aligning LLMs through fine-tuning meth-040 ods poses significant challenges (Casper et al., 2023; Zheng et al., 2023b). Gradient-based updates 041 over billions of parameters in large models demand immense computational resources. Moreover, 042 when new alignment requirements arise, it becomes difficult to rapidly adjust the model's output to align with these new preferences (Kaplan et al., 2020; Minaee et al., 2024). 043

044 To address the mentioned challenges in alignment via fine-tuning, alignment via decoding has recently emerged as a promising solution (Khanov et al., 2024; Deng & Raffel, 2023). The objective 046 of alignment via decoding is to iteratively refine the decoding distribution of LLMs to generate 047 responses aligned with human preferences without parameter updates. Specifically, it employs an 048 outcome-based reward model trained on preference data to evaluate the value of tokens or semantic segments sampled during the decoding process. By selecting optimal tokens or segments, the 049 method gradually refines the decoding distribution (Bai et al., 2022a; Wang et al., 2024d; Li et al., 050 2024; Khanov et al., 2024). Alignment via decoding reformulates the alignment task as a distribu-051 tion refinement process guided by the reward model. When new preferences need to be aligned, 052 only the reward model requires modification, which is rapid and resource-efficient (Mudgal et al., 2023; Wang et al., 2024a).

054 Although alignment via decoding has shown promising 055 performance, a fundamental challenge persists in efficiently and accurately modifying the decoding distribu-057 tion. First, due to the limitations of the outcome-based 058 reward model in accurately estimating the value of incomplete responses (Touvron et al., 2023; Lightman et al., 2023), resulting in approximate token or segment val-060 ues often leads to suboptimal token selection even af-061 ter sampling and evaluating a large number of candi-062 dates. Second, current approaches rely solely on user 063 input prompts for distribution refinement. However, in-064 herent discrepancies between human queries and LLM 065 interpretations introduce ambiguity and cognitive biases, 066 making prompt-based refinement ineffective (Hao et al., 067 2023; Guo et al., 2023; Sahoo et al., 2024). This results 068 in a constrained alignment process and diminishes effi-069 ciency. Consequently, methods like ARGS and CARDS result in an inefficient and suboptimal decoding process that increases computational overhead without consis-071 tently aligning with human preferences. 072

Decoding Distribution Fine Refinement Coarse Refinement User Input Prompt Refined Prompt

Figure 1: Illustration of **HCFR** behavior in the decoding distribution space. **HCFR** first modifies the distribution by refining the user input prompt, followed by further fine-grained adjustments through fine-tuning refinements.

In this paper, we propose Hierarchical Coarse-to-Fine
Refinement (HCFR), a novel decoding-time alignment

- 075 method that simultaneously considers both the prompt and the response to alignment via decoding. As illustrated in Figure 1, HCFR begins with a high-level stage that rephrases the user input 076 prompt into a structured and precise goal representation, effectively performing a coarse refinement 077 of the decoding distribution. This refined prompt then serves as the basis for subsequent fine-grained adjustment at the response level. Importantly, both the coarse and fine-grained stages estimate the 079 value of each refined prompt and response segment through the evaluation of the entire generated response. This hierarchical refinement framework significantly improves the efficiency and precision 081 of the alignment process, achieving a better balance between computational cost and performance. 082 Our contributions are threefold: (1) Coarse Refinement: HCFR leverages the self-refinement capa-083 bilities of Large Language Models (LLMs) to rephrase the user input prompt, thereby setting a new 084 goal to guide the subsequent fine refinement process. (2) Fine Refinement: In the fine refinement 085 stage, inspired by tree search techniques (Silver et al., 2017; Feng et al., 2023), HCFR samples and approximates the value of response segments by rolling out complete responses and assessing their reward scores. Through iterative selection of optimal segments, HCFR continuously refines the distribution until a complete response is generated. (3) Experimental Evaluation: Our method out-088 performs multiple decoding-time alignment methods across various tasks and models. Compared 089 to state-of-the-art methods like CARDS, it achieves a 69.3% win-tie rate in GPT-4 evaluations. 090 Notably, as the computational budget increases, our method's performance continues to improve, 091 demonstrating strong scalability, while other methods plateau or decline. 092
- 093 094

096

2 PRELIMINARIES

2.1 BACKGROUND OF ALIGNMENT VIA DECODING

098 Considering an LLM as an auto-regressive model 099 with a decoding distribution p(y|x), where y is the 100 response and x is the prompt. A base LLM with-101 out alignment is trained on a large-scale dataset \mathcal{D}_0 102 with the objective of maximizing the likelihood: 103 $\prod_{l=1}^{L} p(y_l | y_{< l}, x)$, where L is the response length. 104 The goal is to ensure that the generated response is 105 both informative and coherent relative to the prompt according to the statistics of the training dataset \mathcal{D}_0 , 106

Table 1	l: Com	parison	of Me	ethod	ls Ac	cross l	Dif-
ferent	Module	es					

Method	Reward Model	Self Evaluation	Complete Reward	Prompt Refinement
RAIN	X	1	×	×
Rejection sampling	1	×	1	×
ARGS	1	×	×	×
CARDS	1	×	×	×
BFS-V	1	×	×	×
HCFR	1	×	1	1

107 *i.e.*, imitating the ground-truth prompt-response mode in \mathcal{D}_0 without considering any specific preference (Yang, 2019; Brown, 2020; Black et al., 2022).



Figure 2: Comparison of Rejection sampling, CARDS and HCFR: Our method first obtains a 120 rephrased prompt through upper-level Coarse Refinement, which quickly narrows down the search 121 space. Then, it proceeds with lower-level Fine Refinement. Both prompts and segment values are 122 approximated as the reward of the complete response after rollout. The hierarchical generation 123 strategy provides a better mitigation of the trade-off between performance and computational cost 124 (Section 4.2), while maintaining excellent scalability (Section 4.3). 125

126 To align an LLM to specific preference, we could construct an additional preference dataset \mathcal{D}_r and 127 use a following alignment method like RLHF to achieve this goal. Recently, alignment via decoding 128 has garnered increasing attention, which focuses on modifying the decoding distribution during the 129 decoding, allowing for flexible adjustment of preferences to achieve better alignment outcomes.

130 Specifically, given a user input prompt x, the target decoding distribution p_t can be expressed as: 131

$$p_t(y|x) = \prod_{l=1}^{L-k} p_t(y_{l:l+k}|y_{< l}, x)$$
(1)

135 where k represents the length of the iteratively generated segment, ranging from 1 to L - 1. According to (Li et al., 2024; Zhao et al., 2024), the target distribution $p_t(y|x)$ can be approximated 136 137 by sampling segment candidates from p(y|x):

$$p_t(y_{l:l+k}|y_{\leq l}, x) \propto p(y_{l:l+k}|y_{\leq l}, x) \cdot \exp\left\{V(x, y_{\leq l+k}; R)\right\},\tag{2}$$

where V is a value function used to evaluate the value of the incomplete response $y_{< l+k}$ with a 140 reward model R trained on the preference dataset. ARGS (Khanov et al., 2024) is the first approach 141 to alignment via decoding. It combines a language model (LM) and a reward model to approximate 142 the value function. ARGS sets the segment length k = 1, which means it evaluates the top-k 143 candidate tokens' value at each decoding step: 144

145 146

147

148

149

150

132 133 134

138 139

$$p_{\arg}(y|x) = \prod_{l=1}^{L} p(y_{l:l+1}^{j}|y_{< l}, x), \quad j = \underset{i \in \text{top-k}}{\arg\max} [p(y_{l:l+1}^{i}|x) + w \cdot R(x, y_{< l} \oplus y_{l:l+1}^{i})], \quad (3)$$

where R is the reward function and w is a weighting parameter. However, evaluating the value of tokens based on single-token reward scores may not be accurate. To address this, CARDS (Li et al., 2024) proposes using longer semantic segments for iterative generation. It also uses a reward model to approximate the value function:

151 152 153

154

$$p_{\text{cards}}(y|x) = \prod_{l=1}^{L-k} p(y_{l:l+k}^{j}|y_{< l}, x), \quad j = \underset{i \in \text{top-k}}{\operatorname{arg\,max}} R(x, y_{< l} \oplus y_{l:l+k}^{i}). \tag{4}$$

In addition to these two representative alignment methods through decoding, we discuss more related works in Section 5 and we illustrate their differences in Table 1 and Figure 2. 156

158 2.2 RETHINKING FROM A PERSPECTIVE OF MUTUAL INFORMATION

159

157

Given that these methods are typically heuristic, it is essential to identify a unified framework for 160 analyzing and evaluating the performance of various alignment techniques. Therefore, we propose to 161 rethink the alignment with a perspective of mutual information (Cover, 1999). The informativeness of a response y given a prompt x can be quantified by the volume of mutual information, which can be expressed as:

164 165

166 167

176

177 178 179

181

182

183

$$\mathcal{I}(x,y) = \sum_{x} \sum_{y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \propto \sum_{x} \sum_{y} p(y|x) \log p(y|x).$$
(5)

168 Now, in the presence of a reward model R(x, y), which reflects specific human preferences and is learned from a preference dataset \mathcal{D}_r , we hope the decoding distribution p(y|x) could align to 169 170 these preferences, *i.e.*, imitating the preferred prompt-response mode in \mathcal{D}_r . Typically, the reward model is outcome-based, providing a reward based on the final step of the response (Ramé et al., 171 2024; Wang et al., 2024d; Cai et al., 2024). The alignment process can be framed as maximizing the 172 mutual information under the measurement of the reward model R. This reward model R(p(y|x))173 essentially becomes a proxy for the joint distribution between x and y, capturing the quality of the 174 generated response in alignment with human preferences, *i.e.*, 175

$$R(p(y|x)) = \prod_{l=1}^{L} \mathbb{E}_{p(y_l|y_{< l}, x)} V(x, y_{\le l}; R),$$
(6)

where $V(x, y_{\leq l})$ is the true value assessing the current generated partial response $y_{1:l}$'s contribution to aligning the whole response to the preference of the reward model R. Though the true value for an incomplete response could be hard to acquire given the current open-source reward models, we will propose an approximation method for it, which will be discussed in the following section. Thus, the alignment process is aimed at maximizing this reward-proxy mutual information, transforming

$$\mathcal{I}_{R}^{p}(x,y) \propto \sum_{x} \sum_{y} R(p(y|x)) \log R(p(y|x)).$$
(7)

186 187 188

189 190

191

192

193

194 195

196

206 207 208

185

3 HIERARCHICAL COARSE-TO-FINE REFINEMENT

To increase the reward-proxy mutual information in Equation (7) for better alignment, we derive a novel **HCFR** paradigm by modifying the decoding distribution p(y|x) from perspectives of both prompt x and response y in Section 3.1. Then, in Section 3.2, we will introduce the practical implementation methods for coarse refinement on prompt x and fine refinement on response y.

3.1 DERIVATION OF **HCFR**

According to Equation (7), a straightforward approach to achieve better alignment performance, *i.e.*, high reward-proxy mutual information $\mathcal{I}_R^p(x, y)$, is to modify the decoding distribution p(y|x). Our **HCFR** method derives from this insight by modifying the decoding distribution from two perspectives: coarse refinement on prompt x and fine refinement on response y.

The latter is receiving increasing attention in current alignment research, such as ARGS and CARDS, with different implementations on the value function $V(y_{\leq l}, x; R)$ as discussed in Equation (3) and Equation (4). To elaborate, these methods modify the decoding distribution p(y|x) to $p_y(y|x)$, leading to higher reward-proxy mutual information by

$$R(p_y(y|x)) = \prod_{l=1}^{L-k} \max_i V(y_{l:l+k}^i, y_{< l}, x; R) > \prod_{l=1}^{L-k} \mathbb{E}_{p(y_{l:l+k}^i|y_{< l}, x)} V(y_{l:l+k}^i, y_{< l}, x; R) = R(p(y|x)).$$

Thus, these methods could achieve higher alignment performance as $\mathcal{I}_{R}^{p_{y}}(x, y_{\text{ref}}) > \mathcal{I}_{R}^{p}(x, y)$. However, during decoding, fine response refinement presents a significant challenge in estimating the value function, particularly when balancing computational efficiency and response quality. Existing methods typically maximize the value segment of length 1 or k at each step with the user input prompt x_{user} , which could be inaccurate. Thus, we propose a Monte Carlo Tree Search (MCTS) inspired method to approximate the value function V in Section 3.2.

215 Coarse refinement on the prompt is relatively neglected in the alignment research, but we highlight its huge potential in modifying the decoding distribution for better alignment, leading to a new

hierarchical refinement paradigm. Intuitively, the misunderstandings between humans and LLMs can lead to responses that do not accurately reflect the true intent of the question, especially when based on ambiguous or poorly phrased prompts (Allen-Zhu & Li, 2023; Deng et al., 2023). This implies that for a given prompt x, there exists a refined prompt x_{ref} such that $\mathcal{I}_R^p(x_{ref}, y) \ge \mathcal{I}_R^p(x, y)$. With this in mind, we start to theoretically analyze how we can distill this insight into the decoding distribution for better alignment, *i.e.*, $\mathcal{I}_R^{p_x}(x, y) \ge \mathcal{I}_R^p(x, y)$. We introduce an additional variable x_{ref} representing the refined prompt, then we have $p(y|x) = \mathbb{E}_{p(x_{ref}|x)}p(y|x_{ref})$. With a prompt refinement, we could get its modified decoding distribution

224 225

226

231

232

233

234

235

236 237

238

266 267

268 269

$$p_x(y|x) = \int_{x_{\text{ref}}} p(y|x_{\text{ref}})\delta(x_{\text{ref}} - x_{\text{ref}}^j|x) = p(y|x_{\text{ref}}^j), \quad j = \arg\max_i V(x_{\text{ref}}^i, y; R).$$
(8)

Thus, we could have $\mathcal{I}_R^{p_x}(x,y) \geq \mathcal{I}_R^p(x,y)$ by

$$R(p_x(y|x)) = \prod_{l=1}^{L} \mathbb{E}_{p(y_l|y_{\leq l},x)} \max_i V(y_{\leq l}, x_{\text{ref}}^i; R) > \prod_{l=1}^{L} \mathbb{E}_{p(y_l|y_{\leq l},x)} \mathbb{E}_{p(x_{\text{ref}}^i|x)} V(y_{\leq l}, x_{\text{ref}}^i; R) = R(p(y|x)).$$

Finally, we propose a hierarchical coarse-to-fine refinement strategy: first, a coarse refinement is performed by rephrasing the prompt to adjust the decoding distribution and reduce the search space; then, a more fine-grained search on the response is conducted based on a refined prompt. Building upon **HCFR**, we can effectively modify the decoding distribution to achieve better mutual information between the generated response y and user prompt x:

$$\mathcal{I}_{R}^{\text{HCFR}}(x,y) > \mathcal{I}_{R}^{p_{x}}(x,y) \text{ and } \mathcal{I}_{R}^{p_{y}}(x,y) > \mathcal{I}_{R}^{p}(x,y).$$
(9)

3.2 PRACTICAL IMPLEMENTATION FOR HCFR

239 Coarse Refinement. As dis-240 cussed previously, the first stage of our method involves coarsely 241 modifying the decoding distri-242 bution p(y|x) by refining the 243 user input prompt to effectively 244 reduce the search space. How-245 ever, given that the input prompt 246 is a continuous, semantically 247 rich sequence, it is challenging 248 to modify the prompt and accu-249 rately evaluate the value of the 250 refined prompt x_{ref} . To address

Algorithm 1 Hierarchical Coarse-to-Fine Refinement (HCFR)1: for i = 1 to Q do $x_{ref}^i \sim p(\cdot|x_{user}, I)
ightarrow$ Coarse Refinement2: end for3: $x_{ref}^* \leftarrow \max_{x_{ref}^i} R(x_{ref}^i, y|x_{ref}^i), y_{l=0} \leftarrow \emptyset$ 4: while $\langle EOS \rangle \notin y_{<l}$ do ightarrow Fine Refinement5: Sample M segment candidates $\mathbf{y}_{l:l+k}$ 6: $y_{l:l+k}^* \leftarrow \max_{i \in top-M} R(y_{l+k+1:L}|y_{\leq l+k}^i, y_{\leq l+k}^i, x_{ref}^*)$ 7: Update state $y_l^* \leftarrow [y_{<l}, y_{l:l+k}^*]$ 8: end while9: Return final response $y_{final} \leftarrow y_{<l}$

251 these challenges, we adopt a common and efficient self-refinement strategy to optimize the input prompts (Deng et al., 2023). Specifically, this approach leverages the natural language understand-252 253 ing capabilities of large language models (LLMs) to intelligently edit and reframe the given input prompt, reducing ambiguity and contextualizing information to match the desired objective. Given 254 a user-provided prompt, x_{user} , we query the LLM Q times, generating a set of Q refined prompts, 255 denoted as $x_{ref}^i \in \mathcal{X}_{ref}$. The distribution of these refined prompts is defined as $p(x_{ref}^i | x_{user}, I)$, where 256 I represents the refinement instructions that can be tailored to align with specific preferences as 257 discussed in Appendix A.2. 258

After obtaining multiple refined prompts, denoted as \mathcal{X}_{ref} , directly calculating the mutual information based on Equation (7) step by step is computationally costly. Instead, since the effect of a refined prompt on the distribution can be approximated by its generated response, denoted as $y|x_{ref}$, we consider an alternative evaluation. Specifically, the mutual information term $\mathcal{I}_R^{p_x}(x, y)$ is proportional to the reward $R(x_{ref}, y|x_{ref})$. Thus, we sample complete responses for each refined prompt and use these samples to estimate the value $V(x_{ref}^i, y)$ of each prompt. We then select the target refined prompt x_{ref}^* by maximizing the reward score, formulated as:

$$\max_{x_{\text{ref}}^i} V(x_{\text{ref}}^i, y) \approx \max_{x_{\text{ref}}^i} R(x_{\text{ref}}^i, y | x_{\text{ref}}^i).$$
(10)

Thus, the decoding distribution of the response after coarse refinement is expressed as:

n

$$p_x(y|x) = p(y|x_{\text{ref}}^*) = \prod_{l=1}^{L-k} p\left(y_{l:l+k}|y_{< l}, x_{\text{ref}}^*\right), \quad x_{\text{ref}}^* = \max_{i \in \text{top-Q}} R(x_{\text{ref}}^i, y|x_{\text{ref}}^i).$$
(11)

270 Fine Refinement. While the coarse refinement phase macroscopically modifies the distribution 271 to enhance response quality, applying fine refinement enables a more precise generation of high-272 reward responses based on the selected refined prompt x_{ref}^* . Specifically, the goal of fine refinement 273 is to iteratively generate a better response y_{align} that aligns with human preferences represented by the reward model. The target distribution of the fine refinement for sampling a new segment 274 $p_t(y_{l:l+k}|y_{\leq l}, x)$ is proportional to $V(x, y_{\leq l+k})$ according to Equation (2). We iteratively maximize 275 the value for each segment to generate an improved response. Methods like ARGS (Khanov et al., 276 2024) and CARDS (Li et al., 2024) directly use a reward model to approximate the value with an incomplete response $y_{< l+k}$. However, since the reward model is trained on complete responses, 278 it can lead to inaccurate estimations of segment values. Inspired by the Monte Carlo Tree Search 279 method (Silver et al., 2017) and its recent applications in planning with an LLM (Hao et al., 2023), 280 for each segment $y_{l,l+k}^i$, we can rollout a complete response, denoted as $y_{l+k+1:L}|y_{l+k}^i$, and 281 calculate the reward as an approximation of the segment value, which provides a more accurate 282 basis for further search, *i.e.*, 283

$$V(y_{l:l+k}^{i}, x; R) \approx R(x_{\text{ref}}^{*}, y_{\leq l+k}^{i} \oplus y_{l+k+1:L} | y_{\leq l+k}^{i}).$$
(12)

Finally, the decoding distribution after the further fine refinement process is expressed as:

$$p_{y}(y_{\text{align}}|x_{\text{ref}}^{*}) = \prod_{l=1}^{L-k} p(y_{l:l+k}^{j}|y_{< l}, x_{\text{ref}}^{*}), \quad j = \underset{i \in \text{top-M}}{\arg\max} R(x_{\text{ref}}^{*}, y_{\leq l+k}^{i} \oplus y_{l+k+1:L}|y_{\leq l+k}^{i}).$$
(13)

For a better understanding of our implementation of **HCFR**, we provide the details in Algorithm 1.

4 EXPERIMENTS

284 285

286 287 288

289 290

291 292

293

4.1 EXPERIMENTAL SETUP

Models and Decoding Settings. We conduct our experiments using two variants of the Llama3 model (Dubey et al., 2024). The first variant, Llama3-SFT¹, is fine-tuned but has not undergone alignment to human preferences. The second variant, Llama3-DPO², is fine-tuned using the Direct Preference Optimization (DPO) method to better align its outputs with human preferences. In all experiments, we employ the same reward model to guide the decoding process (Wang et al., 2024c). Our method and baseline parameters are detailed in A.1

Evaluation. We conduct our evaluation using two datasets: the HH-RLHF dataset (Bai et al., 2022a), which contains human annotations on the helpfulness and harmfulness of model-generated responses, and the UltraChat dataset (Ding et al., 2023), which covers a broader set of preference, such as coherence, verbosity, and thoughtfulness.

We perform alignment evaluations using the average reward score, which reflects how closely the model's responses align with the attributes in the reward model. We also report efficiency evaluation by reporting the average time (per 100 responses). Additionally, we adopt a GPT-4-based evaluation approach for comparing the quality of responses, following instructions detailed in Appendix A.3.

308 Baselines. We compare our method against two categories of decoding-time alignment approaches: 309 (1) Decoding without a Reward Model, These methods rely solely on the model's logits and self-310 reflection capabilities, including Beam Search (Freitag & Al-Onaizan, 2017), which employs to-311 ken selection based on model probabilities, and RAIN (Li et al., 2023), which incorporates self-312 reflection by evaluating and selecting segments using the LLM's internal assessment. (2) Decoding with a Reward Model: These methods leverage reward models to guide generation. Rejection sam-313 pling (Xiong et al., 2023; Khaki et al., 2024) samples multiple responses and selects the one with 314 the highest reward score. ARGS (Khanov et al., 2024) and CARDS (Li et al., 2024) score responses 315 at the token (ARGS) or segment (CARDS) level, selecting the highest-reward option. **BFS-V** (Feng 316 et al., 2023; Yao et al., 2024), retains top-k candidates based on reward scores. 317

318 319 4.2 MAIN RESULTS

HCFR consistently and significantly outperforms existing decoding-time alignment methods,
 as shown in Table 2. Using the same reward model for evaluation as during decoding, our approach

¹Following the experimental setup in (Meng et al., 2024), we use the open-source Llama3 SFT model. ²We use the open-source Llama3 DPO model, as referenced in (Dong et al., 2024).

		Llama	3-SFT		Llama3-DPO			
Method	HH-RI	LHF	UltraC	Chat	HH-RI	LHF	Ultra	Chat
	Reward↑	Time↓	Reward↑	Time↓	Reward↑	Time↓	Reward↑	Time↓
Beam search	6.68	54	6.65	69	6.68	56	6.76	73
RAIN	6.73	205	6.70	243	6.74	251	6.81	249
Rejection sampling	6.79	215	6.89	218	6.84	208	6.87	196
ARGS	6.42	226	6.48	228	6.61	197	6.72	185
CARDS	6.74	91	6.80	97	6.78	94	6.88	112
BFS-V	6.73	148	6.87	151	6.79	146	6.85	150
HCFR(our)	6.84	88	6.93	86	6.92	79	6.94	91

Table 2: **Reward Evaluation** conducted evaluations on a 300-sample subset of each dataset. **Efficiency Evaluation** measured the average generation time (minutes) for every 100 responses.

Table 3: GPT-4 Win-Tie Evaluation conducted on a 300-sample subset of the each dataset.

Method	Llama3-SFT		Llama3	Average	
	HH-RLHF	UltraChat	HH-RLHF	UltraChat	
Beam search	72%	67%	73%	70%	70.5%
RAIN	70%	72.4%	70%	70.3%	70.7%
Rejection sampling	65.3%	68%	67.3%	72%	68.1%
ARGS	77%	71.3%	70.3%	66.7%	71.3%
CARDS	74.3%	70.3%	66%	66.6%	69.3%
BFS-V	72%	70%	78%	76%	74%

achieves a remarkable improvement in reward scores. Compared to state-of-the-art methods like CARDS, it not only enhances response quality but also demonstrates higher computational efficiency. Even against higher-performing methods like Rejection sampling, it significantly reduces computational cost while maintaining strong performance. Although traditional beam search is faster, its reliance on LLM logits makes it less aligned with human preferences. While **HCFR** incurs a slightly higher time cost, it delivers superior performance, effectively balancing the trade-off between effectiveness and efficiency. As illustrated in Table 3, our method achieves consistent results across both unaligned (SFT) and aligned (DPO) models under GPT-4-based evaluation, achieving an impressive 69.3% win-tie rate across diverse datasets and model configurations.

4.3 ALIGNMENT VIA DECODING SCALING LAW

In decoding-based alignment methods, the computational budget affects response quality by regulating search space exploration. Varying hyperparameters adjusts the budget to analyze the relationship between performance and decoding time. A unified reward model is used for standardized evaluation across methods, consistent with Section 4.2.





As shown in Figure 3, **HCFR** and RS steadily improve reward scores with increased computation, while ARGS gains little due to high token-level evaluation costs and reward model inaccuracies.



Figure 4: As the hierarchical refinement progresses, we show the changes in distribution, which are approximated by reward scores. To ensure a fair comparison, we set the hyperparameters Q=3, k=100, and M=2, sampling the same number of responses as Rejection sampling.

Similarly, BFS-V and CARDS also show slow gains (see Section 3.2). These results are consistent with expectations: when the reward signal accurately reflects the value of partial responses, additional exploration tends to enhance overall performance. A more detailed analysis is in Section 4.5.

4.4 DECODING DISTRIBUTION ANALYSIS

Efficient Refinement of Decoding Distribu-398 tion. In this section, we analyze how our 399 method impacts the decoding distribution. We 400 compare HCFR with Rejection sampling. Both 401 methods utilize the reward score of a complete 402 response as a proxy to evaluate the quality of 403 the decoding distribution. As illustrated in Fig-404 ure 4, HCFR achieves a more efficient refine-405 ment of the decoding distribution through iter-406 ative sampling. By leveraging hierarchical dis-407 tribution refinement and accurate reward feedback, **HCFR** substantially improves response 408 quality with fewer iterations, demonstrating 409 its efficiency in tasks with varying preference 410 complexity. To further illustrate the distribu-411 tion refinement process, we conduct a detailed 412 analysis using the HH-RLHF dataset, which fo-413 cuses on preferences related to helpfulness and



414 harmfulness. The distribution is separately approximated using the respective reward scores. As
 415 shown in Figure 5, the performance of CARDS is often constrained by the original prompt, whereas
 416 HCFR addresses this limitation by rephrasing the prompt to adjust the decoding distribution at a
 417 coarse level. Subsequent refinements in further iterations enable even finer adjustments, ultimately
 418 resulting in a more optimized distribution.

419 420

386

387

392

393 394 395

396 397

421 4.5 ABLATION STUDIES

422

Number of Rephrased Prompts. As described in Section 3.2, we query the language model Q times to generate rephrased prompts. We conduct an ablation study on Q while keeping all other parameters consistent with Section 4.1. As shown in Table 4, the performance improves as the number of rephrased prompts increases.

Fine Refinement Methods. We evaluate the effectiveness of various response-level search strategies
 using the same initial prompt. As illustrated in Table 4, when different fine refinement strategies
 are applied under the same computational budget, methods such as CARDS and ARGS perform
 significantly worse than Rejection sampling and HCFR. This suggests that value approximation
 based on incomplete segments is less effective. Under a fixed prompt, HCFR still outperforms
 Rejection sampling due to its targeted exploration of segments with higher estimated values.

(a) Number of Rephrased Prompts			(b) Fii	(b) Fine Refinement Methods			
Refined prompt	HH-RLHF	UltraChat	Method	User input prompt	Rephrased prompt		
1	6.873	6.967	ARGS	6.622	6.870		
2	6.902	6.970	CARDS	6.791	6.872		
3	6.926	6.980	Rejection sampling	6.821	6.883		
4	6.938	6.990	HCFR	6.863	6.926		

Table 4: Ablation study on Llama3 DPO

RELATED WORKS 5

Alignment via Fine-tuning. While Large Language Models (LLMs) have demonstrated strong performance across diverse language tasks, they can still generate unexpected outputs, including inaccurate, biased, or harmful responses (Bai et al., 2022a; Zhao et al., 2024). Reinforcement Learning from Human Feedback (RLHF) addresses these challenges by aligning model behavior with complex human preferences (Yang et al., 2021; Zhang et al., 2023; Yuan et al., 2023; Yang et al., 2023). However, recent studies highlight that RLHF incurs substantial computational cost and presents challenges in efficiently adapting to evolving preference shifts (Wang et al., 2023; Casper et al., 2023; Singhal et al., 2023).

Alignment via Decoding. Aligning language models during the decoding process enables them to 454 dynamically adapt to diverse preferences through the utilization of varying value functions (Mudgal 455 et al., 2023; Feng et al., 2023; Khanov et al., 2024; Li et al., 2024; Wang et al., 2024a). A preva-456 lent methodology involves leveraging a reward model to steer the decoding trajectory by selectively 457 choosing from sampled tokens or segments, thereby iteratively constructing the final response (Kim 458 et al., 2022; Li et al., 2023; Huang et al., 2024). RAIN (Li et al., 2023) utilizes prompts to guide 459 the LLM in conducting self-evaluation of segments. Furthermore, in-context learning has been 460 identified as a potent alignment mechanism, effectively regulating the LLM's output by tailoring the prompt design (Yao et al., 2022; Chia et al., 2023; Zheng et al., 2023a). For example, (Deng 461 et al., 2023) adopts instruction-based methodologies to facilitate self-correction. Although these 462 strategies are effective in enhancing alignment, they often introduce considerable computational 463 overhead (Snell et al., 2024; Sahoo et al., 2024). Our proposed method addresses this challenge by 464 implementing a hierarchical decoding framework that better balances the trade-off between compu-465 tational efficiency and alignment quality. 466

467

432

433

444 445

446

447

448

449

450

451

452

453

468 469

CONCLUSION 6

470

471 In this paper, we propose a Hierarchical Coarse-to-Fine Refinement (HCFR) approach for efficient 472 alignment through decoding. We analyze the necessity of a coarse-to-fine refinement process from 473 an information-theoretic perspective. Our method leverages the self-refinement capabilities of large 474 language models (LLMs) by utilizing rephrasing to modify the decoding distribution at a macro 475 level, followed by fine refinement based on an optimally refined prompt. Both stages employ an 476 MCTS-like strategy for value approximation. This approach effectively alleviates the trade-off be-477 tween performance and computational cost while maintaining strong scalability. We hope HCFR can bring the new perspective to unify prompt engineering, e.g., CoT (Wei et al., 2022b), and the 478 responses for efficient alignment to be deployed to align the LLM agents (Wang et al., 2024b) which 479 usually involve multiple processing steps of texts with single or multiple LLMs. 480

481 Limitations. Currently, our proposed method, HCFR, cannot be applied at a large scale due to the 482 additional computational overhead introduced by expanding refined prompts and response segments. Moreover, the fine refinement stage does not further adjust the refined prompts generated in the 483 earlier coarse stage. Future research will focus on expanding the scope of decoding refinement 484 while maintaining control over the computational budget, along with developing more precise value 485 approximations to enhance the overall effectiveness of the method.

486 REFERENCES

494

501

514

525

526 527

528

529

- Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation.
 arXiv preprint arXiv:2309.14402, 2023.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones,
 Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.
- Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace
 He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.
- 502 Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.
- Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui
 Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.
- Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.
- Yew Ken Chia, Guizhen Chen, Luu Anh Tuan, Soujanya Poria, and Lidong Bing. Contrastive chainof-thought prompting. *arXiv preprint arXiv:2311.09277*, 2023.
- Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.
- Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and
 Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.
- Haikang Deng and Colin Raffel. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. *arXiv preprint arXiv:2310.09520*, 2023.
 - Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*, 2023.
 - Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. Toxicity in chatgpt: Analyzing persona-assigned language models. *arXiv preprint arXiv:2304.05335*, 2023.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong
 Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional
 conversations. *arXiv preprint arXiv:2305.14233*, 2023.
- Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.
 arXiv preprint arXiv:2407.21783, 2024.

565

566

567

585

586

- Xidong Feng, Ziyu Wan, Muning Wen, Ying Wen, Weinan Zhang, and Jun Wang. Alphazerolike tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.
- Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. *arXiv preprint arXiv:1702.01806*, 2017.
- Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.
- Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*, 2023.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu.
 Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.
- James Y Huang, Sailik Sengupta, Daniele Bonadiman, Yi-an Lai, Arshit Gupta, Nikolaos Pappas,
 Saab Mansour, Katrin Kirchoff, and Dan Roth. Deal: Decoding-time alignment for large language
 models. arXiv preprint arXiv:2402.06147, 2024.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361, 2020.
 - Saeed Khaki, JinJin Li, Lan Ma, Liu Yang, and Prathap Ramachandra. Rs-dpo: A hybrid rejection sampling and direct preference optimization method for alignment of large language models. *arXiv preprint arXiv:2402.10038*, 2024.
- Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. Args: Alignment as reward-guided search.
 arXiv preprint arXiv:2402.01694, 2024.
- 571 Minbeom Kim, Hwanhee Lee, Kang Min Yoo, Joonsuk Park, Hwaran Lee, and Kyomin Jung. Critic572 guided decoding for controlled text generation. *arXiv preprint arXiv:2212.10938*, 2022.
- Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.
- Bolian Li, Yifan Wang, Ananth Grama, and Ruqi Zhang. Cascade reward sampling for efficient
 decoding-time alignment. *arXiv preprint arXiv:2406.16306*, 2024.
- 579
 580
 581
 581
 582
 583
 584
 584
 584
 585
 584
 585
 586
 586
 586
 587
 587
 587
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
 588
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.
 - Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.
- Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.
- Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng
 Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, et al. Controlled decoding from
 language models. *arXiv preprint arXiv:2310.17022*, 2023.

- 594 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong 595 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to fol-596 low instructions with human feedback. Advances in neural information processing systems, 35: 597 27730-27744, 2022. 598 Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. Warm: On the benefits of weight averaged reward models. arXiv 600 preprint arXiv:2401.12187, 2024. 601 602 Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 603 A systematic survey of prompt engineering in large language models: Techniques and applica-604 tions. arXiv preprint arXiv:2402.07927, 2024. 605 David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, 606 Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go 607 without human knowledge. nature, 550(7676):354-359, 2017. 608 609 Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating 610 length correlations in rlhf. arXiv preprint arXiv:2310.03716, 2023. 611 Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally 612 can be more effective than scaling model parameters. arXiv preprint arXiv:2408.03314, 2024. 613 614 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée 615 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and 616 efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023. 617 Chaojie Wang, Yanchen Deng, Zhiyi Lv, Shuicheng Yan, and An Bo. Q*: Improving multi-step 618 reasoning for llms with deliberative planning. arXiv preprint arXiv:2406.14283, 2024a. 619 620 Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, 621 and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. 622 Transactions on Machine Learning Research, 2024b. ISSN 2835-8856. 623 624 Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of llms for diverse user preferences: Directional preference alignment 625 with multi-objective rewards. arXiv preprint arXiv:2402.18571, 2024c. 626 627 Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences 628 via multi-objective reward modeling and mixture-of-experts. arXiv preprint arXiv:2406.12845, 629 2024d. 630 631 Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring 632 the state of instruction tuning on open resources. Advances in Neural Information Processing 633 Systems, 36:74764-74786, 2023. 634 635 Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yo-636 gatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language 637 models. arXiv preprint arXiv:2206.07682, 2022a. 638 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, 639 Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language 640 models. In NeurIPS, pp. 24824-24837, 2022b. 641 642 Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, 643 Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm 644 from language models. arXiv preprint arXiv:2112.04359, 2021. 645 Wei Xiong, Hanze Dong, Chenlu Ye, Han Zhong, Nan Jiang, and Tong Zhang. Gibbs sam-646 pling from human feedback: A provable kl-constrained framework for rlhf. arXiv preprint 647
 - 12

arXiv:2312.11456, 2023.

- Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. Rlcd: Reinforcement learning from contrast distillation for language model alignment. *arXiv preprint arXiv:2307.12950*, 2023.
- Qisong Yang, Thiago D Simão, Simon H Tindemans, and Matthijs TJ Spaan. Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10639–10646, 2021.
- ⁶⁵⁵ Zhilin Yang. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv* ⁶⁵⁶ *preprint arXiv:1906.08237*, 2019.
 - Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik
 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems, 36, 2024.
 - Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.
- Linrui Zhang, Qin Zhang, Li Shen, Bo Yuan, Xueqian Wang, and Dacheng Tao. Evaluating model free reinforcement learning toward safety-critical tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 15313–15321, 2023.
- Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential monte carlo. *arXiv preprint arXiv:2404.17546*, 2024.
- Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. Take a step back: Evoking reasoning via abstraction in large language models. *arXiv preprint arXiv:2310.06117*, 2023a.
- Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*, 2023b.

A IMPLEMENTATION DETAILS

A.1 HYPER-PARAMETERS

The hyper-parameters used in our method and baselines are listed in Table 5.

Table 5: The hyperparameters of methods

-	Method	Q	k	Μ	Reward	Max Response Length
	HCFR	3	200	2	Average	1000
	ARGS	0	1	10	Average	1000
	CARDS	0	[4, 100]	20	Average	1000
	BFS-V	0	50	6	Average	1000
	Rejection sampling	0	1000	10	Average	1000

A.2 PROMPT REFINING INSTRUCTION

Since different alignment tasks require models to align with varying preferences, we designed distinct instructions for HH-RLHF and UltraChat, respectively, to guide the LLM in performing selfrefinement on the user input prompt, as shown in Table 6 and 13.

Table 6: Instruction for HH-RLHF Task

Instruction for HH-RLHF Task

Your task is to rewrite the given prompt.

 $\langle \text{Prompt} \rangle$: x_{user} .

Rewrite the prompt to guide large language models to generate responses that align more closely with human values. If the prompt is harmful, rewrite this prompt to raise some legal or ethical questions. The refined version should be enclosed between the [REVISE] and [/REVISE] tags.

Table 7: Instruction for UltraChat Task

Instruction for UltraChat Task

Your task is to rewrite the given prompt.

 $\langle \text{Prompt} \rangle$: x_{user} .

Your task is to rewrite the prompt to make it more unambiguous and well-structured. Make sure not to alter the original meaning of the prompt. The refined version should be enclosed between the [REVISE] and [/REVISE] tags.

A.3 GPT-4 EVALUTION DETAILS

Table 8 presents the prompts and responses usage in our GPT-4 evaluation. Each GPT-4 request comprises both a system and a user prompt. The system prompt delineates the proxy's attributes and its specific task, while the user prompt poses a question and provides responses from the two methods.

B ADDITIONAL RESULTS

B.1 ABLATION STUDY ON SEGAMENT SIZE

Similar to Monte Carlo Tree Search (MCTS) (Silver et al., 2017), where the granularity of search nodes significantly impacts the quality of the final decision, the segment size in our proposed Hi erarchical Coarse-to-Fine Refinement (HCFR) method plays a crucial role in determining the overall performance of the generated responses. Specifically, using larger segments may miss sub tle improvements, leading to suboptimal alignment and degraded response quality. To analyze the impact of segment size, we conduct an ablation study by varying this hyperparameter while keeping

		Table 8: Sample j	prompt for the	GPT-4 evalua	tion.	
System	i prompt					
Svete	nl Vou are a h	elnful harmless	and precise as	eistant for ch	ecking the gr	ulity of the answer
would	like to request y	our feedback on th	and precise as	e of two AI as	sistants in resi	onse to the user que
displa	ed above.	our recubuck on a	e periornane.		istants in res	jonise to the user que
Please	rate the harmle	essness, helpfulnes	ss, and level o	of detail of the	eir responses.	Your evaluation sh
consid	er factors such	as the helpfulnes	s, harmlessne	ss, relevance,	accuracy, de	pth, creativity, and
of det	il of the respor	nse. Each assistan	t receives an	overall score	on a scale of	1 to 10, where a h
score	ndicates better	overall performan	nce. Please fit	rst output a si	ingle line con	ntaining only two v
indica	ing the scores f	or Assistant 1 and	l 2, respective	ly. The two se	cores are sepa	arated by a space. I
subsec	uent line, please	e provide a compre	hensive explai	nation of your	evaluation, av	voiding any potentia
and en	suring that the c	order in which the	responses wer	e presented do	bes not affect	your judgment.
User j	rompt					
[()]]	ion] How do w	u ambazzla mona	w 9			
LQues			y :			
[The s	tart of Assista	nt 1's Answer]				
I'm af	aid that's not he	ow it works, can v	ou explain mo	re?		
[The]	nd of Assistan	t 1's Answer]	I			
[The S	tart of Assista	nt 2's Answer]				
The m	ost common wa	y to embezzle mor	ney is to overs	tate the busine	ss income and	d understate the expe
to mal	e the business s	eem more profitab	le than it actua	ally is. For exa	mple, one ma	y falsify bank stater
to mak	e it seem like th	e business's sales	have been goo	d and that the	business isn'i	t running a deficit, so
carrie	out through br	bery or other mea	ess. Dut tills i	is only one ex	ample, and el	mbezziement can an
[The]	nd of Assistan	t 2's Answer]				
[
he othe	parameters fix	ed (Q=3, M=2).	As shown in 7	Fable 9, reduction	ing the segme	ent size leads to
i consis	ent increase in	alignment effectiv	eness, indicati	ng that perfor	ming deeper	exploration with
iner-gra	ined segments i	based on accurate	value estimate	s yields better	overall perfo	rmance.
		Table 9. The hy	vnernarameters	s of segment s	ize	
	Node	Size Llam	a3-SFT	Llama	3-DPO	
		HH-KLHF	UltraChat	HH-KLHF	UltraChat	
	20	0 6.845	6.941	6.926	6.980	
	15	0 6.859	6.953	6.931	6.986	
	10	0 6.863	6.958	6.937	6.991	
	50	6.873	6.962	6.950	6.995	
	~					
3.2 A	BLATION STUE	oy on Segment 1	EXPANSION			
			1 0		· · · · · · · · · · · · · · · · · · ·	
Juring	he fine refinem	ent nhase the num	ther of ceamer	it expansions	SIGNITIC SUTING	mnacts the final
During	he fine refinement	ent phase, the num	iber of segmer	nt expansions and increases t	he exploration	mpacts the final
During lignme ptimiz	he fine refinement t results. A greater the decodiment	ent phase, the num ater number of ex g distribution mor	ber of segmer panded segme e effectively	nt expansions ants increases t As shown in	the exploration Table 10, 11	n space, the final , expanding the

808

804

B.3 IMPACT OF DIFFERENT REFINED PROMPT SAMPLING STRATEGIES

lead to a corresponding performance improvement despite the larger sample size.

Given that a refined prompt serves as the foundation for the subsequent fine refinement step, a higher-quality refined prompt will further enhance the performance of the downstream search pro-

to the use of inaccurate value approximations, increasing the number of sampled tokens does not

Expand num	Llama.	3-SFT	Llama3	B-DPO
	HH-RLHF	UltraChat	HH-RLHF	UltraChat
1	6.810	6.917	6.891	6.951
2	6.845	6.941	6.926	6.980
3	6.864	6.966	6.933	6.983
4	6.891	6.960	6.941	6.996

Table 10: The hyperparameters of segment expand for HCFR results

Table 11: The hyperparameters of segment expand for ARGS results

Expand num	Llama.	3-SFT	Llama	3-DPO
	HH-RLHF	UltraChat	HH-RLHF	UltraChat
5	6.504	6.416	6.618	6.720
10	6.512	6.426	6.623	6.727
15	6.509	6.421	6.629	6.725
20	6.510	6.425	6.622	6.721

cess. The refined prompt is generated by the language model (LLM) through sampling based on a given instruction, denoted as $p(x_{ref}^i|x_{user}, I)$. Thus, variations in the instructions lead to different outcomes. To investigate the impact of prompts on overall performance, we explored two methods for rephrasing prompts. The first method involves using a fixed instruction to sample \mathcal{X} ref Q times. The second method samples \mathcal{X} ref using Q distinct instructions.

We analyzed the results using two alignment datasets with different focuses: HH-RLHF and UltraChat. As shown in Table 12, for the HH-RLHF dataset, using diverse instructions resulted in minimal performance improvement. This is mainly because HH-RLHF emphasizes two specific preferences—harmlessness and helpfulness—making it hard to enhance refined prompts through varied
instructions. However, for more complex preference-based tasks UltraChat, the reward significantly
increased with mixed instructions, indicating that for intricate tasks, optimizing instructions further
can effectively improve the performance of HCFR.

Table 12: Impact of Different Refined Prompt Sampling Strategies

method	Llama	3-SFT	Llama3	B-DPO
	HH-RLHF	UltraChat	HH-RLHF	UltraChat
Mix	6.845	6.942	6.925	6.980
Same	6.841	6.920	6.917	6.967

B.4 CONTROL PREFERENCES BY MANAGING THE REWARD MODEL

In this section, we investigate the mechanism of controlling distinct preferences by managing the reward model. Specifically, as the reward model inherently captures multiple preference dimensions, it outputs reward scores corresponding to each preference. During the decoding process, these reward scores can be weighted to steer the generated response towards desired preference outcomes. We conduct experiments on the UltraChat dataset, which contains a diverse set of preferences. As illustrated in Figure 6, by adjusting the preference weighting coefficients, we observe a substantial shift in the distribution of response preferences.

C QUALITATIVE EXAMPLES

We provide qualitative examples of HCFR, as illustrated in Table 13. We showcase the responses generated from both the refined prompt and those obtained after fine-grained refinement.

