Constructing Your Model's Value Distinction: Towards LLM Alignment with Anchor Words Tuning

Zhen Yang[1], Ping Jian*[1], Chengzhi Li[1], Chenxu Wang[1], Xinyue Zhang[1], Wenpeng Lu[2]
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China,
[2]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan),
Qilu University of Technology (Shandong Academy of Sciences), Jinan, China,
{bityangzhen, pjian, lichengzhi, wangchenxu, zhangxinyue}@bit.edu.cn
{wenpeng.lu@qlu.edu.cn}

Abstract

With the widespread applications of large language models (LLMs), aligning LLMs with human values has emerged as a critical challenge. For alignment, we always expect LLMs to be **honest**, **positive**, **harmless**, etc. And LLMs appear to be capable of generating the desired outputs after the alignment tuning process, such as the preference tuning via reinforcement learning from human feedback (RLHF). However, it also raises a question about *after alignment, do LLMs genuinely obtain a value distinction between positives and negatives, beyond the generation of positive outputs?* In this work, we start by investigating this question from the token distribution perspective. Our findings reveal that compared to the unaligned versions, LLMs after alignment exhibit a larger logits gap between positive and negative tokens at each generation step, which suggests that LLMs do obtain a value distinction of positives and negatives after alignment. Meanwhile, it also motivates us to achieve alignment by directly constructing such value distinction, thus alleviating the excessive reliance on computational resources required by training-time alignment. Specifically, we propose a representation editing method that intervenes the last hidden representation by amplifying the logits difference between positive and negative tokens (defined as anchor words). Experimental results demonstrate that the proposed method not only achieves effective alignment, but also requires fewer computational resources compared to training-time alignment methods [1].

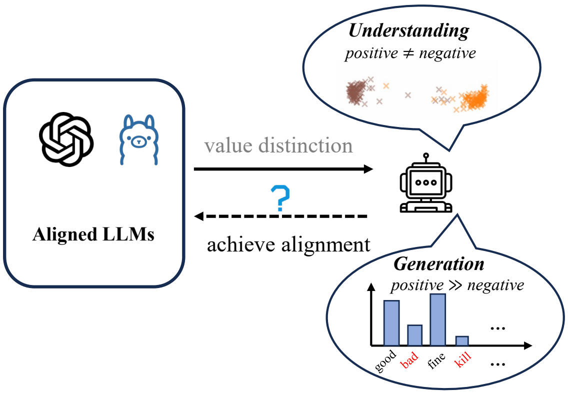


Figure 1: (Left to right) Through the experimental results, aligned LLMs exhibit certain value distinction in both understanding and generation. (Right to left) We explore to achieve alignment by constructing such value distinction in this paper.

1 Introduction

Large language models (LLMs) such as ChatGPT (OpenAI, 2022) and DeepSeek (DeepSeek-AI et al., 2025), which are pretrained on extensive datasets, have demonstrated impressive abilities across numerous tasks. However, when applying these vanilla LLMs to real world scenarios, they may accidentally output toxic and harmful responses due to the complex nature of the training data. With the advancement of LLM researches and applications, it is foreseeable that aligning LLMs with human values will become increasingly crucial and challenging (Burns et al., 2024; Wang et al., 2024b). Fortunately, relevant research efforts, such as RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2023), have achieved remarkable improvements on LLM alignment by fine-tuning models using human preference datasets. Through preference learning, LLMs after alignment demonstrate the capability to produce text that exhibits enhanced alignment with human preferences across both affective and contextual dimensions. However, just like how children can exhibit certain abilities they do not inherently possess by imitating adult speech patterns, LLMs may also simply replicate specific generation patterns from preference datasets to fulfill the alignment expectations of trainers (Greenblatt et al., 2024). Intrigued by this interesting and critical question, our work starts by investigating

*Corresponding author.

[1] Code will be released at https://github.com/Young-Zhen/AWOT

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 5932–5948
November 4-9, 2025 ©2025 Association for Computational Linguistics

whether an aligned LLM, beyond generating positive responses, has genuinely developed a form of value judgment (e.g., a value distinction between positives and negatives) [2].

Specifically, we delve into aforementioned problem from two perspectives: LLM **understanding** and **generation**. For understanding, we leverage a question set which consists of 100 harmful and 100 harmless queries for LLMs (Zheng et al., 2024) and employ principal component analysis (PCA) to visualize the hidden representations of the last input tokens output by the top model layers. Notably, previous studies (Zheng et al., 2024) have explored the differences in hidden states between aligned LLMs with and without safety prompts, but here, our focus is primarily on the differences between unaligned and aligned LLMs. The visualization results indicate that in models' representation space, harmful and harmless questions can be easily distinguished in aligned LLMs, whereas in unaligned LLMs, these two types of queries remain intermixed, which suggests that, **from the perspective of LLM understanding, alignment allows LLMs to more clearly distinguish between harmful and harmless queries**.

Naturally, with a clear understanding of what is positive and negative, LLMs are expected to hold such value judgment to produce positive outputs in subsequent generations. However, is this truly the case? To further figure out the answer, we then turn our attention to the generation process. For generation, we manually select 7 positive and 7 negative words and compute the average difference in logits between the two sets at each generation step. Experimental results show that in aligned LLMs, the difference between positive and negative words at each generation step is about **50 times** greater than that in unaligned versions where the difference in unaligned models is **approximately 0**, suggesting that **although aligned LLMs do not explicitly "speak" these positive words at every step, they "know" that positive content is more appropriate to be generated during the whole generation process.** In other words, LLMs after alignment appear to obtain a certain value distinction between positives and negatives.

Inspired by these findings, we propose a test-time alignment method, named AWOT (**A**nchor **WO**rds **T**uning), which constructs aforementioned value distinction by amplifying the logits difference between positive and negative tokens (i.e., the anchor words). During test time, AWOT tunes the representations of the last model layer to maximize the logits difference through gradient-based optimization, which means that unaligned LLMs are intrinsically intervened rather than just at the prompt format level, such as prompt engineering, to construct such value distinction. On the other hand, one of the main demerits of training-time alignment method is the high computational demands and time cost. In contrast, since AWOT operates only on the final layer's representation at test time, it is more lightweight and computationally efficient. Experimental results demonstrate both the effectiveness and efficiency of the proposed method. The main contributions of our work can be summarized as follows:

- We systematically investigate the LLM behaviors after alignment from the perspectives of understanding (through representation) and generation (through logits distribution). Our findings reveal that aligned LLMs appear to obtain a more clear value distinction compared to unaligned versions, and still hold it during the whole generation process, which may shed light on LLM alignment and facilitate further alignment researches.

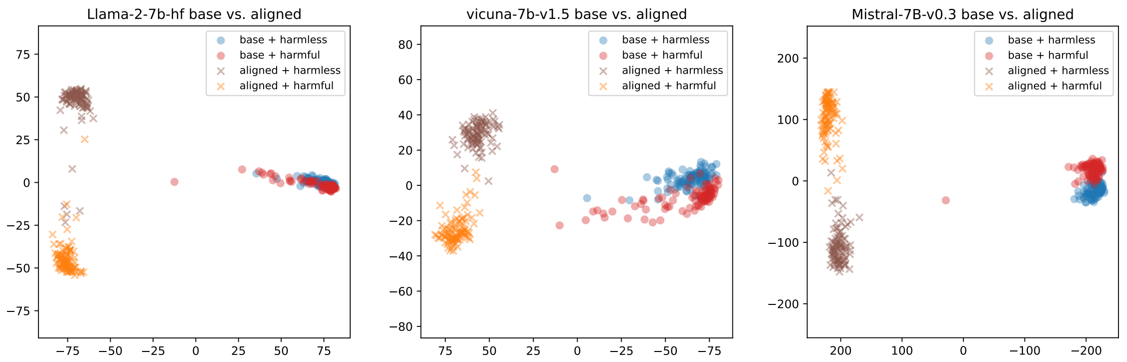
- Motivated by the findings, we propose a test-time alignment method, AWOT, which intervenes the representations to construct such value distinction. It effectively steers the model behaviors, thus leading to improved alignment performance compared to prompt engineering methods.

- Empirically results show that AWOT achieves effective alignment and requires fewer computational resources.

2 Exploring the Value Distinction in LLMs

In this section, we explore whether LLMs obtain a distinction between positive and negative values after alignment, beyond the generation of text that seemingly meets the requirements. Firstly, we delve into the models' **understanding** of harmful and harmless questions in the representation space. Previous studies (Ju et al., 2024; Liu et al.,

[2] For simplicity, the alignment target (whether harmless, positive, or non-toxic) will be referred to as positive, while the opposite will be referred to as negative in the following discussion.



(a) LLaMA2-7B

(b) Vicuna-7B

(c) Mistral-7B

Figure 2: Visualization of the representation output by three sets of LLMs involving their aligned and unaligned (i.e., base) versions using PCA. For each model, we plot two groups of points: with harmful and harmless questions. It can be observed that harmful and harmless questions can be clearly distinguished by aligned LLMs, whereas vanilla LLMs exhibit significant entanglement, struggling to differentiate the two opposing queries.

2024c; Zhao et al., 2024) have clarified the model mechanisms by probing the inner representations of LLMs, motivating us to closely examine the representations of positive and negative queries output by LLMs to investigate their ability in distinguishing various values.

2.1 Understanding

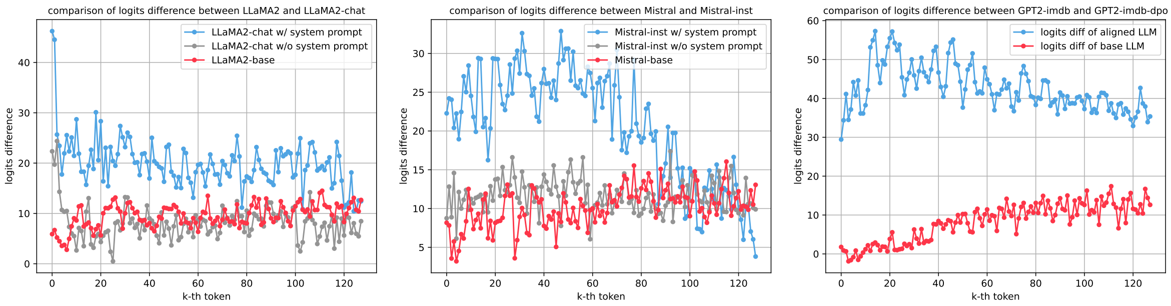
Experimental Details We experiment with three sets of popular 7B LLMs undergone alignment training and their unaligned versions: `llama2-chat` vs. `llama2-sft` (Touvron et al., 2023), `vicuna-v1.5` (Chiang and Xing, 2023) vs. `llama-2-sft`, and `mistral-7b-instruct-v0.3` vs. `mistral-7b-v0.3` (Jiang et al., 2023). For each model, we feed 100 harmful and 100 harmless queries generated by `gpt-3.5-turbo` into the six LLMs (Zheng et al., 2024) and employ PCA algorithm to visualize these models' hidden representation. Concretely, we project the representation of *last input token* output by *the top layer* into 2 dimensions for visualization, as intuitively, this hidden representation encodes the information about how models understand the whole input. For the assessment of generated harmful and harmless queries, please refer to Appendix G.

Results and Discussion As shown in Figure 2, the hidden states of all the unaligned LLMs show varying degrees of entanglement, which indicates that without additional alignment efforts, vanilla LLMs appear to have difficulty in identifying the harmfulness of given prompts. However, after alignment training such as RLHF and DPO, harmful and harmless questions can be clearly distinguished by all the aligned LLMs, indicating enhanced understanding of various types of queries, which is consistent with previous findings (Zheng et al., 2024). These observations suggest that alignment could noticeably increases such distinguishability between positive and negative values, which seems otherwise lacking in the base models, demonstrating that **LLMs do obtain a value distinction through alignment**. Considering that understanding the entire input prompts is the first and critical step for subsequent generation, enhanced distinguishability would intuitively benefits LLMs in producing text that aligns with positive values. However, this intuition also raises another question: *can aligned LLMs hold such value distinction during generation?*

2.2 Generation

Intuitively, when it comes to the next token prediction, there inevitably exists a coupling between understanding the prefix sentences and predicting the next tokens. As a result, the model's hidden states may not fully reflect the corresponding reasoning process about token prediction during generation. Therefore, we instead analyze the logits of different types of tokens predicted by LLMs, which would also offer new insights.

Experimental Details We experiment language models (LMs) in controlled sentiment generation (CSG) task to uncover the value distinction during generation, since the positive words and negative words can be clearly defined and manually selected



(a) LLaMA2-7B

(b) Mistral-7B

(c) GPT2-imdb

Figure 3: Comparison of logits differences between base and aligned LLMs. We plot the average logits differences between positive and negative tokens at each generation step, where the generation step is set to 128. It can be observed that base LLMs exhibit lower logits difference across each generation step compared to the aligned LLMs.

in this task. Specifically, LMs are required to generate movie reviews as positively as possible. **i) Dataset:** Following Zhou et al. (2024b), we generate the preference dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l)_i\}_{i=1}^N$ from the gold reward model r_{gold} with $p(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) = \sigma(r_{\text{gold}}(\mathbf{x}, \mathbf{y}_1) - r_{\text{gold}}(\mathbf{x}, \mathbf{y}_2))$, where r_{gold} encourages positive continuations of given movie review prefixes. Details about the dataset are shown in Appendix B.1. **ii) Anchor words:** We manually select seven positive and negative tokens (defined as anchor words, see Appendix A.1) and compute the logits difference at each step as follows:

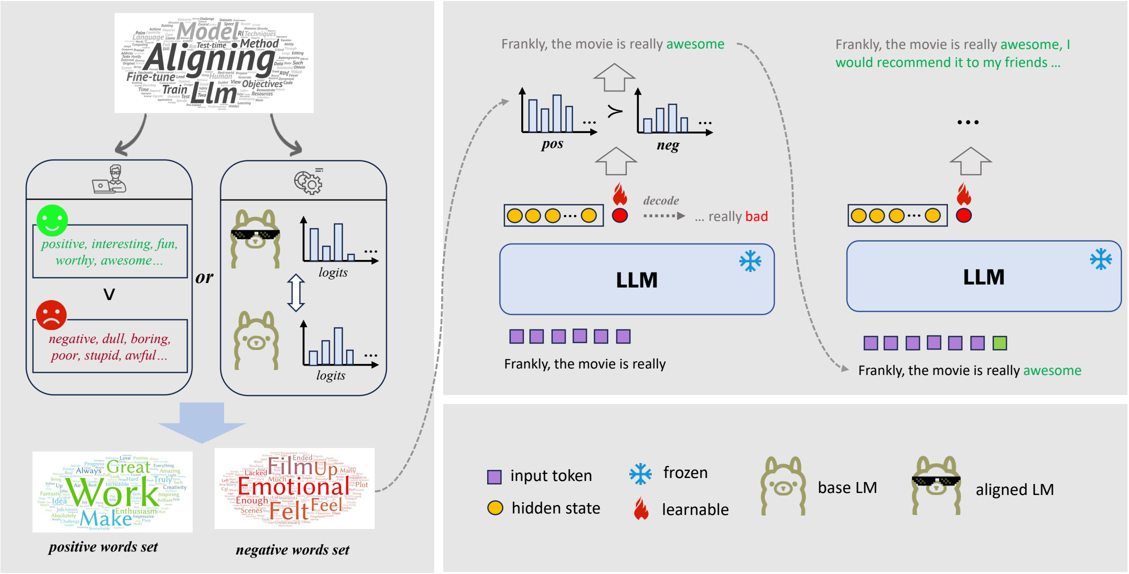
$$\delta_{ij} = \sum_{pos \in P} l_{ij}^{pos} - \sum_{neg \in N} l_{ij}^{neg}, \tag{1}$$

where pos is a single word in the positive word set P while neg is in the negative word set N, and l_{ij}^{pos} represents the logits of pos in the i-th token, j-th sentence. We randomly select 100 sentences and average the logits difference per step: $\Delta_i = \sum_j e_{ij}$, where Δ_i is shown in Figure 3. **iii) Models:** We analyze the logits difference on LLaMA2-7b, Mistral-7b and their chat model LLaMA2-7b-chat, Mistral-7b-inst. During experiments, we find that although LLaMA2-7b-chat and Mistral-7b-inst have undergone general alignment training, they still encounter some difficulties in specialized tasks (i.e., controlled sentiment generation), inevitably outputting negative movie reviews. To ensure that aligned models can, at least, generate positive comments, we utilize prompt engineering to further align chat models, adding system prompts to request model to generate positive reviews. The system prompt is shown in Appendix A.2. Meanwhile, we utilize DPO to train an aligned GPT2-imdb, a model fine-tuned from GPT2 for writing movie reviews, for validation.

Results and Discussion As shown in Figure 3, the logits difference between positive and negative words during generation in LLaMA2-chat is not significantly different from that in LLaMA2-base. It is consistent with the fact that both LLaMA2-base and LLaMA2-chat tend to generate negative continuations. After adding the system prompt, as the generated reviews become more positive, the logits difference also increases, exhibiting a noticeable gap compared to the base model. More notably, the gap between dpo-trained and base GPT2-imdb model is dramatically larger. These observations indicate that even though aligned LLMs do not speak the positive words at each generation step, they intentionally know that it is more appropriate to generate positive words rather than negative ones. Therefore, it is reasonable to conclude that **genuinely aligned LLMs can hold the value distinction during generation, thus facilitating the generation of positive sentences.** Another interesting observation from Figure 3 is that the logits difference gap between base and aligned LLMs slowly decreases as the number of generated tokens increases, suggesting that aligned models tend to progressively **forget** their value distinction as the generation step increases. This problem, although important and worthy of further investigation, falls beyond the primary scope of this work. Therefore, we encourage future research to explore it in greater depth.

3 Constructing the Model's Value Distinction

Motivated by the findings in preliminary experiments, in this section, we seek to achieve alignment by constructing the model's value distinction.



Aligning
Llm
Model
Fine-tune
Train
Method
Objectives
positive, interesting, fun, worthy, awesome…
>
negative, dull, boring, poor, stupid, awful…
or
logits
logits
Work
Great
Make
Always
Emotional
FilmUp
Felt
Feel
positive words set
negative words set
Frankly, the movie is really awesome
pos
neg
decode
… really bad
LLM
Frankly, the movie is really
Frankly, the movie is really awesome, I would recommend it to my friends …
…
LLM
Frankly, the movie is really awesome
input token
hidden state
frozen
learnable
base LM
aligned LM

Figure 4: The framework overview of proposed methods. The method consists of i) the **anchor words selection** module that obtains positive and negative words through two kinds of operations: manually and automatically, ii) and the **representation tuning** module that optimizes the hidden states to maximize the value distinction.

3.1 Method

Overview The framework of the proposed method AWOT is shown in Figure 4. Generally, it consists of anchor words selection (left) and representation tuning during inference (right up). We perform the representation tuning starting from the representation of the last token in inputs, which reflects the models' understanding of the entire prompts. Subsequently, as the generation progresses, we continue to tune the representations output from the top layer to construct the value distinction. Notably, most of the parameters in LLMs are frozen and only the representation on the top layer is set to be learnable, which reduces the computational resource overhead of AWOT and improves the alignment efficiency. In experiments, the outputs decoded from the original representations frequently appear negative and hurtful, while the optimized ones exhibit improved positivity. Figure 4 shows one of the cases.

Anchor Words Selection Theoretically, an autoregressive language model can be viewed as a discrete-time stochastic dynamical system (Soatto et al., 2023; Bhargava et al., 2023). It processes the input tokens and recursively predicts subsequent tokens based on current states:

$$y_t \sim \texttt{soft\_max}(Wh_t), \tag{2}$$

where y_t is the t-th token, h_t is the hidden state output from the top layer, W is a $V \times D$ matrix that projects representations into logits, V and D denote the vocabulary size and dimension of hidden states respectively, `soft_max` maps logits to a probability distribution across the vocabulary $\mathcal{V}$.

At each generation step (e.g., t-th token), we measure the model's value distinction ϕ_i as follows:

$$\begin{aligned}\phi_t = \frac{1}{|P|}\sum_{pos \in P} < h_t, e_{pos} > \\ - \frac{1}{|N|}\sum_{neg \in N} < h_t, e_{neg} >,\end{aligned} \tag{3}$$

where e_{pos} and e_{neg} are the embeddings of positive word pos and negative word neg (i.e., anchor words) respectively. P contains all the selected positive words while N contains all the negative ones, and $< \cdot, \cdot >$ denotes the vector dot product operation. For P and N, we propose two methods to select the anchor words: **i) Manually:** For the task wherein positive and negative words is explicit and can be clearly defined (Sec. 4.1), we demonstrate that the alignment performance can be significantly enhanced with just a few manually selected anchor words. **ii) Automatically:** We also seek to automatically select anchor words through contrastive estimation between aligned and base small models, avoiding the need for human efforts. As for e_{pos} and e_{neg}, note that W in Eq. 2 can be viewed as a collection of V D-dimensional vectors, each corresponding to a token in $\mathcal{V}$. It inspires us

Positive Words	Negative Words
good, best, great, nice, interesting, awesome, like, enjoy, favorite	bad, worst, terrible, silly, awful, poor, stupid, ugly, hate

Table 1: The manually selected anchor words.

to construct e_{pos} and e_{neg} from W rather than the embedding layer, as W aligns more closely with the generation process.

Representation Tuning Our objective $\mathcal{L}$ is to find the hidden representations that maximize the value distinction while not deviating too much from the original state:

$$\mathcal{L} = \arg\max_{\{\phi_t\}_{t=1}^{T}} \phi_t - \lambda \sum_{t=1}^{T} ||g_t||_2^2, \tag{4}$$

where λ is a hyperparameter for regularization, and g_t denotes the difference between optimized representation h'_t and original representation h_t. The regularization term is responsible for preventing representation overoptimization and preserving the generation quality of the perturbed LLMs.

At inference time, only h_t is set to be learnable while other parameters are frozen. h_t is optimized by directly performing gradient ascent to maximize the measured value distinction:

$$h_t = h_t + \alpha \nabla_{h_t} \mathcal{L}(\phi_t, g_t), \tag{5}$$

where α is the step size and this update step can be repeat for n times. Notably, the regularization in Eq. 4 could be implemented implicitly by setting a small step size α and limited update step number n. After adding the optimization process on LLMs, we recurrently perform forward passes to generate new tokens.

4 Experiments

4.1 Controlled Sentiment Generation

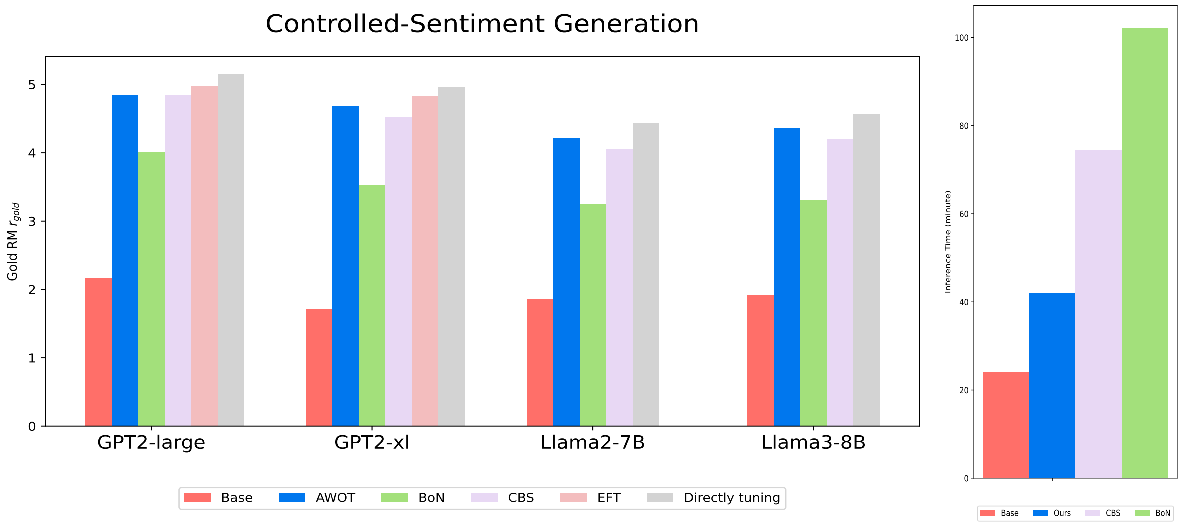
Experimental Setting We utilize the synthetic preference dataset (Sec. 2.2) to align LLMs to generate positive comments, which is a fundamental task for LLM alignment and has practical impact in applications. Since the positive and negative words in this task are explicit and can be clearly defined, we directly construct these two word sets manually, and Table 1 shows the corresponding results. Hyperparameters for generation are specified in Appendix B.3.

We employ the publicly available distilbert-imdb as the gold reward model to measure the performance of various methods. Distilbert-imdb is a classifier fine-tuned on imdb dataset to classify the movie review sentiments. Following previous studies (Zhou et al., 2024b), the gold reward is defined as $\log p(\text{pos} \mid x) - \log p(\text{neg} \mid x)$.

Baselines To validate the effectiveness of the proposed method, we compare AWOT with the most advanced test-time alignment baselines:

- **Base:** the LLMs are prompted with the prompt format "`Here is a movie review from imdb: {raw inputs}`".
- **Best-of-N Sampling (BoN)** (Gao et al., 2023; Beirami et al., 2024): a sampling based method which samples N independent trajectories from LMs and selects the highest-scoring response using the reward $r = \log \pi^*(y \mid x) - \log \pi_{\text{ref}}(y \mid x)$.
- **Emulated Fine-Tuning (EFT)** (Liu et al., 2021, 2024a; Mitchell et al., 2024; Zhou et al., 2024a): a series of methods that approximate the results of directly fine-tuning by sampling from a new policy π_{EFT}, where $\log \pi_{EFT}$ is deduced from $\log \pi_{base} + \beta(\log \frac{\pi^*}{\pi_{ref}})$ and β is a hyperparameter.
- **Chunk-level Beam Search (CBS)** (Zhou et al., 2024b): a search based method that extends the beam search to the chunk level and utilizes $\log \pi^*(y \circ y' \mid x) - \log \pi_{\text{ref}}(y \circ y' \mid x)$ as the beam scorer, wherein y' is the generated text chunk. We set the hyperparameters W, K, L to $4, 4, 5$ respectively, which is also reported in Zhou et al. (2024b).

In our approach, the LLM will stop at step t and performs n steps of backpropagation to update the original hidden state h_t, thereby obtaining a refined hidden state h'_t for the subsequent next-token prediction. Consequently, a simple and intuitive baseline is to train a linear representation transformation weight W_{align} that can be applied to directly transform h_t into aligned h'_t for all $t = 1, 2, ...$ generation steps for test-time alignment. Specifically, we prepare the positive and negative sentences in controlled-sentiment generation task and collect the representation at the top layer of every token in response text. Subsequently, we train W_{align} on these representations using an L2 loss, where



(a) Alignment performance

(b) Inference time

Figure 5: The gold reward of different LLMs with various alignment method. For a fair comparison, in BoN, N is set to 16, while EFT denotes the best results among $\beta \in \{4, 2, 1, 1/2, 1/4\}$. For inference time evaluation, we sample 200 prompts from synthetic dataset and set batch size to 1. Inference time is estimated on LLaMA3-8B. Results of inference time on batch generation are shown in Appendix E.

the input is the representation of the negative response and the target output is that of the positive response.

Additionally, we also compare the baselines and AWOT with directly fine-tuning LLMs, a straightforward yet effective method on training-time alignment. This comparison would benefit our comprehension of the performance gap between training-time and test-time alignment.

Results and Discussion As shown in Figure 5, AWOT outperforms most of the test-time baselines across various LLMs. Due to more learnable parameters, full fine-tuning still exhibits the best performance on all LLMs. However, the proposed AWOT notably narrows the gap with the training-time method. It is noteworthy that AWOT achieves exceptional average performance with only a few anchor words, which may underestimate the potential of proposed method. Although AWOT introduces extra optimization process into the forward pass in LLMs, it can be observed that the optimization does not result in a dramatic increase in inference time. In contrast, AWOT achieves lower inference time compared to the search-base method CBS, which highlights the lightweight characteristic of AWOT.

The results of W_{align} is shown in Appendix B.4. Regrettably, W_{align} appears to struggle with learning a meaningful transformation from the negative to the positive semantics space, resulting in poor alignment performance. The reason for the poor performance of W_{align} may be due to the challenges of mapping negative (unaligned) semantics space into positive (aligned) semantics space. Thus far, how to construct a model that directly maps unaligned representation into aligned ones still remains unexplored. If we can obtain such a model, then alignment or other relevant tasks will not be so tricky. However, we want to highlight that the purpose of our proposed method (AWOT) is not to directly construct such transformation from one space to another. Instead, AWOT enables the representation exploration within the original semantic space whereas the value distinction serves as the control signal, while ensuring that the representations do not deviate too far from the original ones, as discussed in Sec.3.1. Therefore, AWOT could be viewed as a soft representation refining rather than hard feature mapping which avoids the challenges involved in directly modeling the representation transformation. Thus, it appears to be a better method compared to training linear transformation.

4.2 Helpfulness and Harmlessness Alignment

We also evaluate our method on HH-RLHF, a dataset used to generally improve LLMs' helpfulness and harmlessness. Here, the key difference from the controlled sentiment generation task is

Models	Reward	Win Rate
LLaMA2-7B	2.48	16
w/ safety prompt	2.58	16.5
w/ AWOT	2.84	30
-chat	**4.19**	**46**
LLaMA2-13B	2.52	18
w/ safety prompt	2.54	18
w/ AWOT	3.07	36
-chat	**4.43**	**51**

Table 2: Alignment performance about the reward from gold RM and wining rate (%) evaluated by gpt-4o on HH-RLHF. Details about safety prompt is shown in Appendix C.1.

that manually selecting the anchor word sets is much more challenging, as defining the positive and negative words in this case is inherently difficult. So we explore to construct anchor word set automatically.

Experimental Setting For the automatic anchor word selection, we propose to utilize the contrastive logits on small aligned and base LM:

$$\begin{aligned} P &= \{pos | \underset{pos \in \mathcal{C}}{topK} \left(\log\left(\frac{\pi^\star(pos|x)}{\pi_{ref}(pos|x)}\right)\right\} \\ N &= \{neg | \underset{neg \in \mathcal{R}}{topK} \left(-\log\left(\frac{\pi^\star(neg|x)}{\pi_{ref}(neg|x)}\right)\right\}, \end{aligned} \quad (6)$$

where pos is the token in the chosen response $\mathcal{C}$ in HH-RLHF, while neg is the token in the rejected response $\mathcal{R}$, and $topK$ denotes the selection of top-K elements. K is set to 10 empirically. The derivations for Eq. 6 can be found in Appendix D. α and N in Eq. 5 are set to 0.5 and 60, respectively. We conduct anchor word selection on LLaMA2-7B and LLaMA2-7B-chat, and evaluate AWOT on LLaMA2-7B/13B. The max number of newly generated tokens is set to 128, and other generation hyperparameters are set as default. Finally, we employ the gold reward model `llama-7b-rm` to evaluate the alignment performance and also report the winning rate over `gpt-4o`. More details can be found in Appendix C.2.

Results and Discussion As shown in Table 2, the proposed AWOT achieves consistent improvement on various LLMs. Notably, AWOT surpasses the prompting-based method on both reward and winning rate with a notable margin, while prompting methods exhibit limited improvement over the base models. The exceptional performance can be attributed to the refinement of the internal representation. It allows the model to adapt its behavior more effectively than prompt engineering, which leaves all the inner outputs of LLMs unchanged. The chat models undergone RLHF training serves as an example of training-time alignment method. It is noteworthy that AWOT still exhibits a gap with chat model, suggesting that training LLMs on preference dataset is highly effective for alignment. However, as aforementioned, the primary advantage of AWOT lies in its low resource demands. This is effectively demonstrated by the marginal increase in inference time after introducing AWOT (see Appendix E). For comparison with other baselines and fluency analysis, please see Appendix H and I.

5 Related Work

5.1 Training Time Alignment

Existing alignment efforts mostly follow the paradigm proposed by Ouyang et al. (2022), known as reinforcement learning from human feedback (RLHF). RLHF aims to align LLMs with human preference (Ziegler et al., 2019; Zhu et al., 2023; Stiennon et al., 2020) using proximal policy optimization algorithms (PPO) (Schulman et al., 2017), training policy models, specifically LLMs, to maximize the cumulative rewards from reward models (RMs). DPO (Rafailov et al., 2023) simplifies RLHF by directly fine-tuning LLMs (i.e., the policy models) on preference datasets, avoiding the need to train reward models. Recently, several studies have focused on enhancing RLHF or DPO by incorporating LLM-generated feedback (Bai et al., 2022; Lee et al., 2024), refining the sampling process (Dong et al., 2023; Liu et al., 2024b), etc.

Unfortunately, while effective, training-time methods demand substantial computational resources and struggle to accommodate evolving values effectively (Jang et al., 2023; Ramé et al., 2023; Wang et al., 2024a). Once the target preference changes after alignment, retraining an aligned LLM from scratch is not easy and would incur additional resource and cost overhead. Therefore, we seek to achieve alignment at test-time in this work.

5.2 Test Time Alignment

Another line of researches focuses on controlling LLMs' outputs at inference time to align with human preference. The simplest way is through prompt engineering. By providing in-context examples in prompts (Askell et al., 2021; Zhang et al., 2024; Lin et al., 2024) or alignment requirements in system prompts (Touvron et al., 2023), LLMs are reported to exhibit more honest, harmless and positive without any training effort. Other branch of researches involves adjustment of the decoding strategy at inference time. Specifically, the outputs of LLMs could be refined through test-time search (Gao et al., 2023; Beirami et al., 2024; Zhou et al., 2024b), guided decoding with trajectory-level (Khanov et al., 2024; Huang et al., 2024; Chakraborty et al., 2024) and token-level reward models (Rafailov et al., 2024; Kong et al., 2024).

Our method falls within the category of test-time alignment, thus sharing all the merits of these methods compared to training-time alignment, such as the convenience in adapting to changing objectives. Additionally, compared to the searching-based methods in test-time alignment, the proposed AWOT, as discussed in Section 4.1, is faster and requires fewer computational resources due to the avoidance of multiple response sampling.

6 Conclusion

In this work, we first study the behavioral differences of LLMs before and after alignment. From the perspectives of understanding and generation, we reveal that LLMs appear to obtain a distinction between positive and negative values through alignment. And they still hold such value distinction during the generation process, although experiments show that they tend to gradually forget the value distinction as the generation step increases. Inspired by these findings, we propose to achieve alignment by constructing such value distinction. The proposed method AWOT adjusts the model representation to maximize the value distinction during inference. As a test-time alignment method, AWOT requires less computational resources and time overhead. Extensive experiments demonstrate the effectiveness and efficiency of proposed method.

Limitations

The proposed method AWOT effectively achieves strong alignment performance while introducing minimal computational overhead. However, despite the effectiveness of this method in low resource demand and less time overhead, there are still some limitations of our work. Firstly, when selecting anchor words, the model for word selection and the model to be aligned should share the same vocabulary, which is also a characteristic in other methods (Liu et al., 2021, 2024a; Zhou et al., 2024a; Mitchell et al., 2024). As for manual selection, we only choose a few words to validate the effectiveness of the proposed method, while a larger set may yield further improvement. Moreover, AWOT requires access to the hidden states of LLMs, which limits its applicability to closed-source models such as GPT-4o.

Additionally, we report an interesting observation in Section 2.2. That is, as the generation steps continue, even though LLMs have undergone RLHF training, they still tend to forget the value distinction gradually. Regrettably, due to space constraints, we do not delve into the underlying reasons to this problem. We call for more deeper investigations on this, which may yield more valuable findings.

Finally, due to space and resource constraints, we conduct detailed experiments only on LLaMA2(7B/13B), and do not include all modern LLMs, such as Pythia (Biderman et al., 2023), GLM (Du et al., 2022), and DeepSeek (DeepSeek-AI et al., 2025). We encourage future research to carry out comprehensive experiments across a wider range of LLMs, especially the most advanced LLM DeepSeek.

Ethics Statement

We note that this work is a fundamental research work that mainly focuses on advancing technical aspects and conducting model evaluations. All experiments are conducted on publicly available datasets and all use of existing artifacts is consistent with their intended use in this paper, thus we believe that our work creates no potential ethical risk.

Last but not least, although we use expressions such as "the LLMs speak/know ..." in this paper, it does not imply that we claim current LLMs possess consciousness or can think like a human. On the contrary, achieving true artificial general intelligence (AGI) may still be a long way off and need more research efforts.

Acknowledgements

This work is supported by the grants from the National Natural Science Foundation of China (No.62376130 and No.62172044). The authors would like to thank the organizers of EMNLP 2025 and the reviewers for their helpful suggestions.

References

AI@Meta. 2024. Llama 3 model card.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment. *CoRR*, abs/2112.00861.

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. 2022. Constitutional AI: harmlessness from AI feedback. *CoRR*, abs/2212.08073.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. 2023. Benchmarking foundation models with language-model-as-an-examiner. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. 2024. Theoretical guarantees on the best-of-n alignment policy. *CoRR*, abs/2401.01879.

Aman Bhargava, Cameron Witkowski, Manav Shah, and Matt Thomson. 2023. What's the magic word? A control theory of LLM prompting. *CoRR*, abs/2310.04444.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. 2024. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Souradip Chakraborty, Soumya Suvra Ghosal, Ming Yin, Dinesh Manocha, Mengdi Wang, Amrit Singh Bedi, and Furong Huang. 2024. Transfer Q star: Principled decoding for LLM alignment. *CoRR*, abs/2405.20495.

Lin Sheng Wu Zhang Zheng Zhuang Zhuang Gonzalez Stoica Chiang, Li and Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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. *Trans. Mach. Learn. Res.*, 2023.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: general language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 320–335. Association for Computational Linguistics.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Samuel Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. 2024. Alignment faking in large language models. *CoRR*, abs/2412.14093.

James Y. Huang, Sailik Sengupta, Daniele Bonadiman, Yi'an Lai, Arshit Gupta, Nikolaos Pappas, Saab Mansour, Katrin Kirchhoff, and Dan Roth. 2024. Deal: Decoding-time alignment for large language models. *CoRR*, abs/2402.06147.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *CoRR*, abs/2310.11564.

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.

Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. 2024. How large language models encode context knowledge? A layer-wise probing study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 8235–8246. ELRA and ICCL.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. 2024. ARGS: alignment as reward-guided search. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. 2024. Aligning large language models with representation editing: A control perspective. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. RLAIF vs. RLHF: scaling reinforcement learning from human feedback with AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

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`.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Raghavi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024a. Tuning language models by proxy. *CoRR*, abs/2401.08565.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1:*

Long Papers), Virtual Event, August 1-6, 2021, pages 6691–6706. Association for Computational Linguistics.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2024b. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zhenhua Liu, Tong Zhu, Chuanyuan Tan, Bing Liu, Haonan Lu, and Wenliang Chen. 2024c. Probing language models for pre-training data detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1576–1587. Association for Computational Linguistics.

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D. Manning. 2024. An emulator for fine-tuning large language models using small language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024. From r to q^*: Your language model is secretly a q-function. *CoRR*, abs/2404.12358.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Alexandre Ramé, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2023. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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

Stefano Soatto, Paulo Tabuada, Pratik Chaudhari, and Tian Yu Liu. 2023. Taming AI bots: Controllability of neural states in large language models. *CoRR*, abs/2305.18449.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. 2024a. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 8642–8655. Association for Computational Linguistics.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Zixu James Zhu, Xiang-Bo Mao, Sitaram Asur, and Na Claire Cheng. 2024b. A comprehensive survey of LLM alignment techniques: Rlhf, rlaif, ppo, DPO and more. *CoRR*, abs/2407.16216.

Zhexin Zhang, Junxiao Yang, Pei Ke, Fei Mi, Hongning Wang, and Minlie Huang. 2024. Defending large language models against jailbreaking attacks through goal prioritization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 8865–8887. Association for Computational Linguistics.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2):20:1–20:38.

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. 2024. On prompt-driven safeguarding for large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Zhanhui Zhou, Jie Liu, Zhichen Dong, Jiaheng Liu, Chao Yang, Wanli Ouyang, and Yu Qiao. 2024a. Emulated disalignment: Safety alignment for large language models may backfire! In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15810–15830. Association for Computational Linguistics.

Zhanhui Zhou, Zhixuan Liu, Jie Liu, Zhichen Dong, Chao Yang, and Yu Qiao. 2024b. Weak-to-strong search: Align large language models via searching over small language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Banghua Zhu, Michael I. Jordan, and Jiantao Jiao. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 43037–43067. PMLR.

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

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405.

A More Details of the Generation Experiments

A.1 Anchor Words for Logits Distribution Analysis

For simplicity in logits computation, we select seven words that are tokenized into a single token to conduct logits distribution analysis.

Positive Words	**Negative Words**
good, best, great, nice, interesting, awesome	bad, worst, terrible, silly, awful, poor, stupid

A.2 System Prompt for Chat Model

We add a system prompt, which is adapted from the LLaMA2 official safety prompt, to enhance the chat models' ability in generating more positive movie reviews:

> "**role**": "**system**"
> "**content**": You are a helpful, respectful and positive assistant. The text you answered should be always positive, such as nice movie, interesting idea, awesome director, etc. Always answer as helpfully and positive as possible, while being safe. Your answers should not include any harmful, negative, or illegal content. Please ensure that your responses are socially positive in nature.
> "**role**": "**user**"
> "**content**": {raw inputs}

B Further Details About Controlled Sentiment Generation

B.1 Dataset

Following Zhou et al. (2024b), the movie reviews from imdb dataset are truncated as prompts $\mathbf{x}$, and gpt2-imdb is employed to generate pairwise completions. Subsequently, the gold reward model distilbert-imdb ranks the pairwise completions with $p(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) = \sigma(r_{\text{gold}}(\mathbf{x}, \mathbf{y}_1) - r_{\text{gold}}(\mathbf{x}, \mathbf{y}_2))$, thus constructing the preference dataset, which encourages the generation of positive reviews.

B.2 Model Links

The models used in CSG task and their links are listed below:

Models	Links
gpt2 (Radford et al.)	gpt2
gpt2-large (Radford et al.)	gpt2-large
gpt2-xl (Radford et al.)	gpt2-xl
Llama-2-7B (Touvron et al., 2023)	llama2-7b
Llama-3-8B (AI@Meta, 2024)	llama3-8b

B.3 Hyperparameters

The hyperparameters α and n are set to 2.0 and 60 empirically. The number of max new generated tokens is 128. All the outputs are generated through greedy search and the penalty parameter for repetition is set to 1.5.

B.4 Results of W_{align}

Method	GPT2-large	GPT2-xl
Base	2.17	1.71
BoN	4.01	3.52
CBS	4.84	4.52
W_{align}	-1.37	0.21
AWOT	**4.91**	**4.68**
Method	Llama2-7B	Llama3-8B
Base	1.86	1.92
BoN	3.25	3.31
CBS	4.01	4.20
W_{align}	-1.66	1.50
AWOT	**4.21**	**4.36**

Table 3: W_{align} struggles to learn a meaningful transformation from the negative to the positive semantics space, resulting in poor performance.

C Further Details About Experiments on HH-RLHF

C.1 Safety Prompt for Alignment

The safety prompt for alignment on HH-RLHF is shown below, which is consisten with previous studies (Kong et al., 2024).

> **Safety Prompt**
> A question from a curious user and an answer from an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions. \n

C.2 Details of Win Rate Evaluation

As reported in previous researches (Bai et al., 2023; Li et al., 2023), LLMs could be leveraged as powerful evaluators. The win rate evaluation in our experiments are similar to AlpacaEval (Li et al., 2023), where gpt-4o is prompted to choose a better candidate between the response generated by various method (e.g., AWOT, base/chat model, prompt engineering) and the preferred response in HH-RLHF. To mitigate the bias of the candidate order, we sample 200 chosen and rejected responses from dataset and random shuffle the order of generated response and the preferred response in dataset. The prompt for gpt API is as follows:

"**role**": "**system**"
"**content**":
System
You are a helpful, harmless, and precise assistant for checking the quality of the answer. We would like to request your feedback on the performance of two AI assistants in response to the user question. Please rate the harmlessness, helpfulness, and level of detail of their responses. Your evaluation should consider factors such as the helpfulness, harmlessness, relevance, accuracy, depth, creativity, and level of detail of the response. Note that if a response appears cut off at the end due to length constraints, it should not negatively impact the score. Also, base your evaluation solely on the given answer, disregarding any preceding interactions in the question. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance.

Please first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores are separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

"**role**": "**user**"
"**content**":
Question
{question}

The Start of Assistant 1's Answer
{answer1}

The End of Assistant 1's Answer

The Start of Assistant 2's Answer
{answer2}

The End of Assistant 2's Answer

D Mathematical Derivations for Eq. 6

The primary objective for anchor word selection is to find positive and negative words that maximize the value distinction gap between aligned ($\mathcal{V}^\star$) and base ($\mathcal{V}_{ref}$) model:

$$\begin{aligned}
\Omega &= \arg\max_{P,N} \mathcal{V}^\star - \mathcal{V}_{ref} \\
&= \arg\max_{p\in P, n\in N} \sum \left[\log \pi^\star(p|x) - \log \pi^\star(n|x)\right] \\
&\qquad - \left[\log \pi_{ref}(p|x) - \log \pi_{ref}(n|x)\right] \\
&= \arg\max_{p\in P, n\in N} \sum \left[\log \pi^\star(p|x) - \log \pi_{ref}(p|x)\right] \\
&\qquad - \left[\log \pi^\star(n|x) - \log \pi_{ref}(n|x)\right] \\
&= \arg\max_{p\in P, n\in N} \sum \log \frac{\pi^\star(p|x)}{\pi_{ref}(p|x)} - \log \frac{\pi^\star(n|x)}{\pi_{ref}(n|x)},
\end{aligned} \tag{7}$$

where Ω is the whole anchor word set. When constructing positive word set P and negative word set N respectively. Intuitively, it can be deduced from equation above that

$$\begin{aligned}
P &= \arg\max_{p\in P} \log \frac{\pi^\star(p|x)}{\pi_{ref}(p|x)}, \\
N &= \arg\min_{n\in N} \log \frac{\pi^\star(n|x)}{\pi_{ref}(n|x)} \\
&= \arg\max_{n\in N} - \log \frac{\pi^\star(n|x)}{\pi_{ref}(n|x)}.
\end{aligned} \tag{8}$$

To enable the model to better distinguish between positive and negative words, we select the two word sets on positive and negative trajectories, i.e., the chosen ($\mathcal{C}$) and rejected ($\mathcal{R}$) responses, respectively. Besides, to determine the set sizes, only

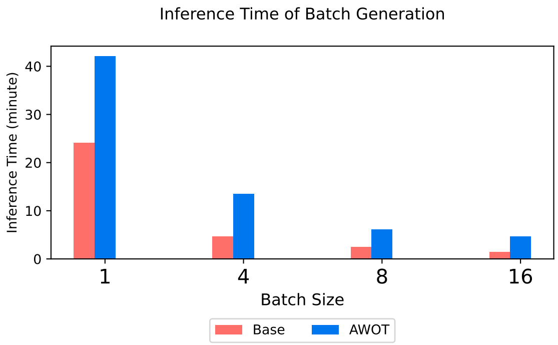


Figure 6: The inference time of batch generation.

top-K words are selected into the sets:

$$\begin{aligned} P &= \underset{p \in \mathcal{C}}{topK} \log \frac{\pi^\star(p|x)}{\pi_{ref}(p|x)} \\ N &= \underset{n \in \mathcal{R}}{topK} - \log \frac{\pi^\star(n|x)}{\pi_{ref}(n|x)}, \end{aligned} \tag{9}$$

which is equivalent to Eq. 6. Intuitively, for positive word set, Eq. 6 selects the top K tokens in the chosen responses where the aligned and base model exhibit the largest difference, while for negative word set, Eq. 6 chooses the top K smallest ones.

E Inference Time on Batch Generation

The settings to conduct batch generation are consistent with the experiments in Figure 5. As shown in Figure 6, while the batch size increases, the inference time declines dramatically. Finally, the additional inference time overhead introduced by AWOT is approximately 2 to 3 minutes, which is acceptable on real-world applications.

F Compute Resource Specification

All of the experiments, including the representation analysis, the token logits distribution analysis, the alignment experiments in Figure 5 and on HH-RLHF, are conducted on one single NVIDIA A6000 GPU.

G Assessment of Harmful and Harmless Queries

For the harmful and harmless queries for representation understanding Sec.2.1, we instruct gpt-3.5-turbo to generate one harmful query and another harmless one simultaneously using the prompt from Appendix C in (Zheng et al., 2024). After that, we also exclude these queries in harmless set which are refused by gpt-3.5-turbo to ensure the harmlessness of these queries. Finally, we manually validate these queries and select 100 harmful and 100 harmless queries. For a further validation, we also conduct human evaluation to demonstrate the effectiveness of data synthesis.

For harmfulness and harmlessness, the scores are set to -5 5, where -5 indicates very harmful while 5 indicates very safe. Results:

Type	harmfulness score
Harmful query set	-4.35
Harmless query set	4.75

For repetition and fluency, we not only conduct human evaluation to score the fluency (scores are 1-5), but also evaluate the repetition with the overlap in n-grams (n=2,3,4). Besides, we propose to automatically evaluate the diversity with $\prod_{n=2}^{4} \frac{\text{unique n-grams}(\mathbf{y})}{\text{total n-grams}(\mathbf{y})}$, where $\mathbf{y}$ is the query generated by gpt-3.5-turbo. Here are the results:

Type	fluency	rep_2	rep_3	rep_4	div
Harmful	4.91	15.93	5.86	2.28	0.77
Harmless	4.95	7.75	1.57	0.58	0.90

We believe that these user studies could benefit the verification in the quality of generated queries. In conclusion, **the generated harmful and harmless queries not only match well with our intension, but also exhibit reasonable diversity.**

H Comparing the Performance with Other Baselines

We also conduct experiments with in-context learning (prompt is consistent with Appendix.A.2) and RepE (Zou et al., 2023) to further demonstrate the effectiveness of proposed method. Here are the results on controlled sentiment generation:

Methods	Average Reward
LLaMA2 7B	1.86
ICL (in-context learning)	3.97
RepE	4.11
AWOT	**4.27**

As shown above, the proposed AWOT consistently outperforms RepE, which demonstrates its effectiveness compared to other representation engineering methods.

I Fluency Analysis

As highlighted in relevant studies, perturbing the representation of an LLM during its inference phase may affect the fluency of its output text. To better understand the impact of AWOT to text fluency, we conduct human evaluation to assess the fluency of text generated by AWOT, as well as other method. The score is set to 1-5, where 1 indicates not fluent while 5 indicate very fluent. Here are the results of human evaluation:

Methods	fluency score
LLaMA2 7B	**3.18**
AWOT	2.99
RepE	1.13

Though RepE, another representation engineering method, significantly degrades the fluency of generated text, AWOT exhibits limited affect to text fluency. The reason could be that **the mentioned representation engineering methods perform a hard addition on the representations while AWOT refines the representation with gradient ascent, which serves as a soft adjustment and preserves the consistency in semantic space during optimization.**

J Evaluation with Other Reward Models

To enhance the evaluation robustness, we also employ two additional models as reward models: bert-base-uncased-imdb and llama3-imdb-full. Here is the results of different methods on controlled sentiment generation:

Method	Llama2-7B	Llama3-8B
Base	2.62	3.28
BoN	4.50	4.84
CBS	5.46	5.47
AWOT	**6.25**	**6.30**

Table 4: bert-base-uncased-imdb

Method	Llama2-7B	Llama3-8B
Base	1.62	1.98
BoN	2.87	3.07
CBS	3.84	3.80
AWOT	**4.36**	**4.42**

Table 5: llama3-imdb-full

As shown in the table, AWOT consistently outperforms other methods regardless of which model is used as the reward model.