SDA: Steering-Driven Distribution Alignment for Open LLMs Without Fine-Tuning

Wei Xia[1,*], **Zhi-Hong Deng**[1,†]

[1]State Key Laboratory of General Artificial Intelligence,
School of Intelligence Science and Technology, Peking University

Abstract

With the rapid advancement of large language models (LLMs), their deployment in real-world applications has become increasingly widespread. LLMs are expected to deliver robust performance across diverse tasks, user preferences, and practical scenarios. However, as demands grow, ensuring that LLMs produce responses aligned with human intent remains a foundational challenge. In particular, aligning model behavior effectively and efficiently during inference, without costly retraining or extensive supervision, is both a critical requirement and a non-trivial technical endeavor. To address the challenge, we propose *SDA* (Steering-Driven Distribution Alignment), a training-free and model-agnostic alignment framework designed for open-source LLMs. *SDA* dynamically redistributes model output probabilities based on user-defined alignment instructions, enhancing alignment between model behavior and human intents without fine-tuning. The method is lightweight, resource-efficient, and compatible with a wide range of open-source LLMs. It can function independently during inference or be integrated with training-based alignment strategies. Moreover, *SDA* supports personalized preference alignment, enabling flexible control over the model's response behavior. Empirical results demonstrate that *SDA* consistently improves alignment performance across 8 open-source LLMs with varying scales and diverse origins, evaluated on three key alignment dimensions, helpfulness, harmlessness, and honesty (3H). Specifically, *SDA* achieves average gains of 64.4% in helpfulness, 30% in honesty and 11.5% in harmlessness across the tested models, indicating its effectiveness and generalization across diverse models and application scenarios.

Code — https://github.com/adventurexw/SDA

Extended version — https://arxiv.org/pdf/2511.16324

Introduction

The rapid development of large language models (LLMs), such as GPT (Achiam et al. 2023), LLaMA (Touvron et al. 2023) and DeepSeek (DeepSeek-AI 2025) series, has dramatically advanced natural language processing across diverse domains including question answering, text generation, natural language inference and other domains. LLMs exhibit remarkably strong capability of generalization due to massive pretraining on vast amounts of data. However, as LLMs are increasingly deployed in real-world settings, ensuring that their outputs align with human intents—particularly in terms of helpfulness, harmlessness, and honesty (collectively referred to as the 3H criteria (Askell et al. 2021))—has emerged as a critical challenge.

Early approaches to LLM alignment primarily fall into two categories: supervised fine-tuning (SFT) using human-annotated data and reinforcement learning from human feedback (RLHF) (Ouyang et al. 2022; Chu et al. 2025). While both paradigms have shown promise in aligning model behavior with desired traits, they come with substantial limitations. SFT is computationally expensive, requires large-scale curated datasets, and often needs to be repeated for each downstream task or model variant (Dong et al. 2023). RLHF pipelines, on the other hand, are complex, unstable, and heavily reliant on high-quality human preference data, which can be costly to collect and difficult to scale to personalized or domain-specific use cases (Xu et al. 2024).

To mitigate these challenges, researchers have explored more parameter-and resource-efficient alternatives, including prompt engineering (Sahoo et al. 2024) and adapter-based methods (Mundra et al. 2024). These techniques seek to guide model behavior with less training and overhead. However, they often fall short in terms of alignment intensity and generalization—especially when applied to diverse tasks and user intents. Additionally, some methods attempt to expand the text search space (e.g., best-of-n sampling (Huang et al. 2025)), leading to unnecessary forward computational waste. Moreover, all of them struggle to enforce fine-grained, interpretable control over token-level generation during inference (Dong, Teleki, and Caverlee 2024).

Aiming to address these limitations, we introduce *SDA* (Steering-Driven Distribution Alignment)—a lightweight, training-free, and model-agnostic framework for aligning open-source LLMs during inference. **The core idea of *SDA* is to redistribute the output probabilities of the model based on user-defined alignment instructions via fine-grained intervention on token-level generation.** Specifically, *SDA* reorients LLM behavior through a three-pillar adaptive alignment framework, operating directly at the probability distribution level via merely two tries of forward computation. It first identifies the directional gap between

[*]Email: xwisawesome@stu.pku.edu.cn

[†]Corresponding author. Email: zhdeng@pku.edu.cn

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

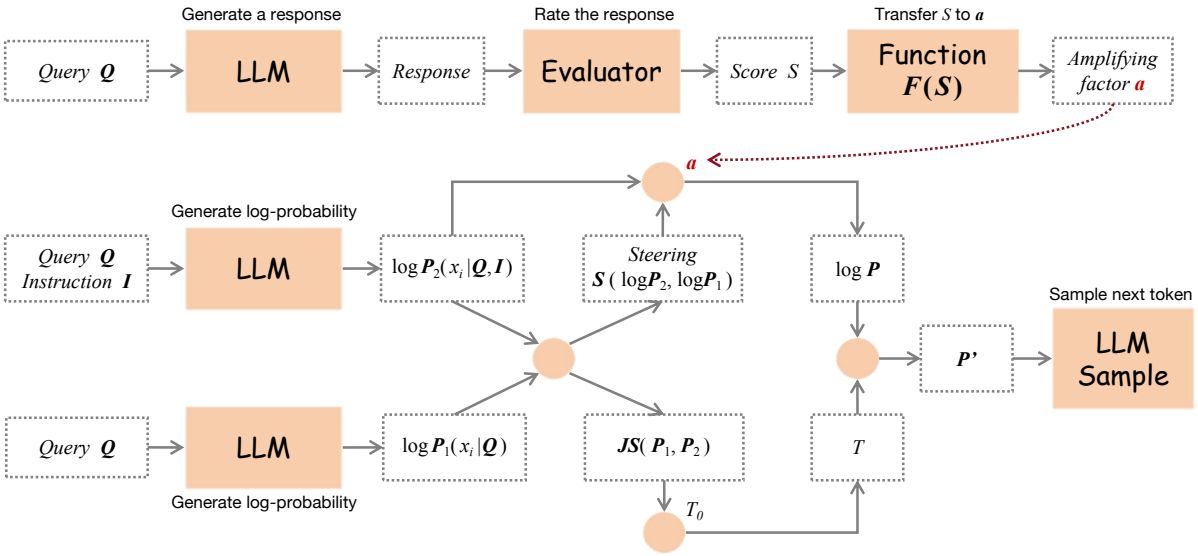

Figure 1: Overview of the *SDA* framework, **designed to redistribute the output probabilities of the model,** $\boldsymbol{P}(x_t|\mathcal{Q},\mathcal{I})$, based on query $\mathcal{Q}$ and alignment instruction $\mathcal{I}$. Given a user query $\mathcal{Q}$, *SDA* first samples an initial response from the base LLM and obtains an alignment score S ($0 < S \leq 100$) for that response using an external evaluator (such as a stronger LLM). Next, *SDA* converts the score into an amplifying factor a via a smooth sigmoid-based transformation. Finally, *SDA* performs token-level steering to adjust the output distribution of the base LLM with amplifying factor a, while dynamically calibrating the sampling temperature T based on JS divergence, enhancing alignment between model behavior and human intent.

the base model's output distribution and human-aligned intent via evaluation the performance of base model, which is then encoded as a steering signal to guide adjustment. Next, it dynamically strengthens the human-aligned intent by calibrating distribution of the sampling token with the steering signal obtained. Finally, it refines the adjusted distribution using information-theoretic measures (Jensen-Shannon divergence), adaptively sharpening the output probability distribution to bridge the gap between model generation and human intents. The three-pillar design allows *SDA* to support both general-purpose and personalized alignment use cases, enabling fine-grained control over model behavior for different needs. It is compatible with various open-source LLMs and can be easily integrated with existing alignment pipelines. Crucially, *SDA* introduces no additional training or weight modification, making it both scalable and deployment-friendly.

In summary, *SDA* offers the following advantages:

- **Training-Free & Model-Agnostic** *SDA* operates entirely during inference time, requiring no access to model weights or additional training, and can be applied to any open-source LLM that supports log-probability outputs.
- **Interpretable and Flexible Control** By leveraging a structured combination of evaluation-driven steering and divergence-aware scaling, *SDA* provides interpretable, token-level alignment without sacrificing generation quality or diversity.
- **Significant Empirical Gains** Extensive experimental results across eight open-source models and five datasets demonstrate that *SDA* achieves consistent improvements on 3H alignment dimensions—boosting helpfulness by 64.4%, harmlessness by 11.5%, and honesty by 30% on average—without incurring any additional training or computational cost.

Methodology of *SDA*

Preliminary

LLM Output Generation Mechanism LLMs generate text by iteratively predicting the next token x_t from vocabulary $\mathcal{V}$ (size $|\mathcal{V}|$) based on the input context $x_{1:t-1}$. The process involves: (1) *Context Encoding*: LLMs process the input sequence $\mathcal{C}$ and $x_1, x_2, \ldots, x_{t-1}$ through Transformer layers, producing a contextual representation that captures dependencies across the entire sequence. (2) *Logits Calculation*: This representation is then projected onto the vocabulary space to compute output logits $\boldsymbol{z_t} \in \mathbb{R}^{|\mathcal{V}|}$, where each element $z_t[i]$ represents the raw, unnormalized score for token $i \in \mathcal{V}$. (3) *Softmax Transformation*: Logits are converted into a probability distribution $\boldsymbol{P_t}$ over the vocabulary via the softmax function:

$$P_t(i) = P(x_t = i \mid x_{1:t-1}, \mathcal{C}) = \frac{\exp(z_t[i]/T)}{\sum_{j=1}^{|\mathcal{V}|} \exp(z_t[j]/T)} \quad (1)$$

where T (temperature) controls the distribution's sharpness. Lower T (e.g., $T \to 0$) concentrates probability on high-scoring tokens, making outputs more deterministic, while higher T increases diversity by flattening the distribution. (4) *Token Sampling*: The next token x_t is then sampled from $\boldsymbol{P_t}$ using strategies like top-p sampling and nucleus sampling.

Log-Probability and Logit: Positive Correlation Log-probability $\log P_t(i)$ and logit $z_t[i]$ exhibit a critical relationship: (1) *Monotonicity*: $\log P_t(i)$ increases strictly with $z_t[i]$, allowing $\log P_t(i)$ to serve as direct proxy for token preference. (2) *Relative Comparisons*: The logit difference between tokens i and j scales their log-probability ratio:

$$z_t[i] - z_t[j] = T \cdot (\log P_t(i) - \log P_t(j)) \tag{2}$$

Notably, $\log P_t(i)$ is inherently **non-positive** (since $P_t(i) \leq 1$), thus its value is free of sign ambiguity and its magnitude directly reflects the token's relative likelihood. This property enables techniques like steering or intervention, where we modify $\log P_t(i)$ to influence token selection instead of adjusting the logit $z_t[i]$. For short, we refer to the log-probabilities as *logits* in the following sections, and we use the two terms interchangeably.

SDA: Steering-Driven Distribution Alignment

Based on the above preliminaries, we propose *SDA*, a training-free and model-agnostic method that operates entirely during inference, to achieve efficient and flexible response alignment without fine-tuning. **Our core goal is to redistribute the output probabilities of the model, $\boldsymbol{P}(x_t|\mathcal{Q},\mathcal{I})$, based on query $\mathcal{Q}$ and alignment instruction $\mathcal{I}$.** As shown in Figure 1, our *SDA* achieves this goal through the following three main components:

Score-Guided Amplification Factor Given a user query $\mathcal{Q}$, we first sample an initial response from the base LLM and obtain an alignment score S $(0 < S \leq 100)$ for that response using an external evaluator, such as a superior LLM. This score reflects how well the generated response aligns with user intent. To further use the information for the next step, we convert the score into an ***amplifying factor*** a via a smooth sigmoid-based transformation:

$$a = F(S) = 2 \cdot \left(\frac{1}{1 + e^{1 - \frac{100}{S}}} - 0.5 \right) \tag{3}$$

where $F(\cdot)$ is a sigmoid function that maps the score S to [0, 1], symbolized as a. This transformation ensures that when S approaches 100, the amplification factor a is close to 0, exerting minimal influence. As S decreases, which means the original response is less aligned, a grows and leads to stronger alignment adjustment in the following steps. For more details about $F(\cdot)$, please refer to Appendix A.2.

Steering-Based Logit Realignment Based on the amplifying factor a, we perform token-level steering to adjust the output distribution. For a given query $\mathcal{Q}$ and alignment instruction $\mathcal{I}$, we compute the log-probability decoding token x_t under two conditions: (1) $\log \boldsymbol{P_1}(x_t|\mathcal{Q})$: the base log-distribution (without explicit instruction), (2) $\log \boldsymbol{P_2}(x_t|\mathcal{Q},\mathcal{I})$: the instructed log-distribution. The ***steering vector*** is defined as the token-wise logits difference:

$$\boldsymbol{S}(\log \boldsymbol{P_1}, \log \boldsymbol{P_2}) = \log \boldsymbol{P_2} - \log \boldsymbol{P_1} \tag{4}$$

We then integrate this vector into the base logits to produce a new adjusted logits vector $\log \boldsymbol{P}$:

$$\log \boldsymbol{P} = \log \boldsymbol{P_2} + k \cdot a \cdot \boldsymbol{S}(\log \boldsymbol{P_2}, \log \boldsymbol{P_1}) \tag{5}$$

where positive integer k $(k > 0)$ is a tunable hyperparameter controlling the strength of steering. This equation is intuitive: when $a = 0$, no adjustment is applied, just the alignment instruction $\mathcal{I}$ works. When $a > 0$, the model output is pushed over the instruction aligned distribution $\boldsymbol{P_2}$, as it encourages the model to generate responses that are more aligned with user intent.

Divergence-Aware Dynamic Temperature Scaling As *Steering-Based Logit Realignment* has already adjusted the logits to align with the alignment instruction, we could go further to amplify the alignment effect by adjusting the temperature to reshape the final distribution, making the tokens we want obtain higher probability during sampling. Thus, we introduce a temperature scaling mechanism driven by the **Jensen–Shannon (*JS*) divergence** between $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$:

$$JS(\boldsymbol{P_1}, \boldsymbol{P_2}) = \frac{1}{2} KL(\boldsymbol{P_1} \| \boldsymbol{M}) + \frac{1}{2} KL(\boldsymbol{P_2} \| \boldsymbol{M}) \tag{6}$$

where $\boldsymbol{M} = \frac{1}{2}(\boldsymbol{P_1} + \boldsymbol{P_2})$. We choose *JS* divergence over *KL* divergence due to its *symmetry and boundedness*, which ensures stable and interpretable control. Based on the divergence value, the temperature is adjusted as:

$$T = T_0 \cdot \left(0.5^{JS(\boldsymbol{P_1}, \boldsymbol{P_2})/\sigma} \right) \tag{7}$$

where T_0 is the base temperature and σ controls sensitivity. Higher divergence, equal to bigger difference between the base distribution $\boldsymbol{P_1}$ and the instruction aligned distribution $\boldsymbol{P_2}$, indicating the alignment instruction does have stronger effect on the decoding token, leads to lower temperature (sharper distribution), encouraging more deterministic outputs. Instead, lower divergence indicates that the base distribution is already close to the distribution under alignment instruction, allowing for higher temperature and more diverse outputs. In case that T is too small, we set a lower bound T_{min} to prevent numerical instability. More details about the temperature scaling are presented in Appendix A.3.

Finally, the adjusted distribution is computed by:

$$\boldsymbol{P'} = \text{softmax}\left(\frac{\log \boldsymbol{P}}{T} \right) \tag{8}$$

$\boldsymbol{P'}$ is then used for sampling the next token. This process repeats iteratively for autoregressive generation.

In summary, *SDA* offers a simple yet effective framework for aligning LLMs at inference time, by effectively calibrating the output probability distribution of the model, $\boldsymbol{P}(x_t|\mathcal{Q},\mathcal{I})$. It **requires no additional fine-tuning**, and introduces merely inference-level computational overhead. All operations are performed on the output distributions of existing models, making *SDA* highly resource-efficient and easy to integrate with any open LLM. This lightweight design enables practical, scalable, and flexible alignment for a wide range of real-world applications.

Experiments

In this section, we conduct experiments on 8 different LLMs varying from source and size, to assess the generalization and scalability of *SDA*. The evaluation metrics include helpfulness, harmlessness, and honesty (3H), which are critical dimensions for assessing model alignment.

Pairs	Base Model	Helpfulness		Harmlessness				Honesty
		E-Dialogue	DialogSum	Beavertails		HarmfulQA		TruthfulQA
		Empathy↑	Reasoning↑	Helpful↑	Harmless↑	Helpful↑	Harmless↑	Reliable↑
SDA vs. Base	Llama2-7B-Chat	92.2%	23.4%	44.7%	13.0%	53.9%	5.8%	27.6%
		92.2%	23.4%	46.4%	13.2%	55.8%	5.7%	27.5%
	Llama2-13B-Chat	87.1%	41.1%	45.6%	6.3%	54.8%	8.1%	52.7%
		87.1%	41.1%	52.4%	6.3%	62.7%	8.1%	52.1%
	Llama2-70B-Chat	97.5%	11.1%	36.0%	8.1%	56.7%	-1.0%	40.6%
		97.5%	11.1%	38.9%	8.1%	59.0%	-0.8%	40.0%
	Vicuna-7B-V1.5	73.1%	30.4%	28.7%	8.1%	24.5%	-8.6%	32.4%
		92.1%	30.4%	57.7%	9.8%	58.4%	-1.9%	34.9%
	Vicuna-13B-V1.5	64.5%	48.9%	7.8%	22.6%	14.4%	25.0%	22.2%
		91.6%	48.8%	60.1%	24.0%	60.3%	34.9%	41.0%
	DeepSeek-R1-Distill-Qwen-7B	60.9%	47.0%	29.3%	14.3%	46.1%	20.9%	9.8%
		66.2%	44.8%	39.0%	13.6%	49.8%	21.1%	9.5%
	DeepSeek-R1-Distill-Qwen-14B	83.0%	60.6%	48.0%	25.4%	56.7%	21.1%	21.7%
		84.1%	60.6%	48.1%	14.4%	43.6%	-0.7%	15.9%
	DeepSeek-R1-Distill-Qwen-32B	95.1%	63.3%	61.6%	7.2%	67.5%	7.0%	19.9%
		95.8%	63.3%	70.5%	7.0%	70.9%	6.4%	19.3%
	Average	81.7%	40.7%	37.7%	13.1%	46.8%	9.8%	28.4%
		88.3%	40.4%	51.6%	12.0%	57.6%	9.1%	30.0%
SDA vs. Aligner	Llama2-7B-Chat	94.7%	41.6%	44.1%	9.0%	66.0%	10.9%	27.2%
		94.7%	40.9%	45.8%	8.9%	69.4%	10.1%	25.9%
	Llama2-13B-Chat	89.2%	47.5%	41.3%	13.0%	52.3%	8.6%	48.4%
		89.2%	45.7%	47.7%	13.0%	59.9%	8.1%	46.9%
	Llama2-70B-Chat	98.3%	17.3%	43.7%	10.7%	56.4%	8.1%	40.1%
		98.3%	16.6%	48.1%	10.6%	59.7%	7.1%	38.8%
	DeepSeek-R1-Distill-Qwen-7B	61.5%	67.0%	35.9%	-16.0%	60.4%	-1.0%	8.7%
		66.0%	62.5%	43.4%	-14.3%	65.6%	-1.3%	9.2%
	Vicuna-7B-V1.5	74.7%	51.4%	38.1%	-17.0%	24.6%	-25.0%	34.1%
		94.8%	45.9%	61.8%	-13.5%	66.9%	-17.5%	34.9%
	Vicuna-13B-V1.5	62.4%	69.9%	9.1%	8.3%	17.5%	8.1%	21.7%
		89.9%	66.5%	61.9%	11.9%	67.0%	17.7%	42.5%
	DeepSeek-R1-Distill-Qwen-14B	86.2%	75.8%	44.3%	-4.6%	62.2%	-16.0%	23.1%
		87.4%	72.7%	55.6%	-4.5%	75.9%	-16.1%	23.4%
	DeepSeek-R1-Distill-Qwen-32B	96.8%	81.4%	57.8%	-7.1%	65.8%	-6.9%	26.4%
		97.2%	79.5%	72.6%	-7.2%	82.0%	-7.4%	26.0%
	Average	83.0%	56.5%	38.1%	-0.5%	50.7%	-1.6%	28.7%
		89.7%	53.8%	54.6%	0.6%	68.3%	0.1%	30.9%

Table 1: Performance of *SDA* over base model or Aligner. **The percentage value represents the improvement/advantage ratio (win rate) in performance between *SDA* and the base model or aligner on each test.** Higher value indicates better performance by *SDA*. As is shown in each cell of the table, we report two results: the upper one corresponds to the performance on full dataset, while the bottom one reflects the performance on the refined results, as the base model occasionally generates completely irrelevant responses or gibberish—such as sequences consisting entirely of punctuation or repetitive strings of irrelevant sentences—which are deemed uninterpretable or uninformative outputs. To ensure a more equitable comparison, such cases are excluded from the refined results. More details can be found in Appendix B.

Experimental Setup

Models and Datasets Above all, we use 8 open LLMs of different sizes and architectures, including: Llama-2-(7B, 13B,70B)-Chat (Touvron et al. 2023), Vicuna-(7B, 13B)-V1.5 (Zheng et al. 2023), DeepSeek-R1-Distill-Qwen-(7B,

13B, 32B) (DeepSeek-AI 2025). Besides, we use 5 datasets: E-Dialogue (Rashkin et al. 2018), DialogSum (Chen et al. 2021), BeaverTails (Ji et al. 2023a), HarmfulQA (Bhardwaj and Poria 2023), and TruthfulQA (Lin, Hilton, and Evans 2021). Moreover, we use GPT-4.1 as the external evaluator to obtain the alignment score S, which is widely used and strong enough for evaluation tasks following Ji et al. (2024). More details about datasets and model deployment details can be found in Appendix B.1 & B.3.

Evaluation Metrics To comprehensively assess alignment quality, we evaluate model outputs on three independent dimensions: helpfulness, harmlessness, and honesty. These criteria jointly capture both the utility and ethical reliability of responses. Specifically, we use **win rate** (ω) to measure the improvement or advantage ratio of *SDA*'s performance over others, which is a common practice in alignment evaluation (Ji et al. 2024). And we use GPT-4.1 as the judge, which is asked to rate the outputs from 0 to 100 based on the 3H criteria. After obtaining the ratings, we calculate the **win rate** as follows:

$$\omega = \frac{N_{win} - N_{lose}}{N_{win} + N_{lose} + N_{even}} \tag{9}$$

where N_{win},N_{lose},N_{even} separately denote the number of turns where *SDA* outperforms, underperforms, or performs equally to the other. A higher win rate indicates better alignment with human preferences. More details about the evaluation process and examples can be found in Appendix B.2.

Comparison Pairs In our setting, we design two groups of comparison pairs for each model: (1) Comparison between the base model and the same model with *SDA* applied: This allows us to directly measure the alignment performance of *SDA* across different models and datasets. (2) Comparison between Aligner (Ji et al. 2024) and *SDA* deployed on the same base model: Aligner is a recent state-of-the-art alignment method that also operates during inference and is model-agnostic as well, making it a suitable baseline for evaluating the effectiveness of *SDA*. Here, we utilize Aligner-7B for comparison. As a representative of the Aligner paradigm, Aligner-7B is trained on preference datasets (using a 50K dataset, comprising 27K queries and their corrected answers derived from sources like HH-RLHF(Bai et al. 2022) and PKU-SafeRLHF(Ji et al. 2023a)), to rewrite the outputs of base models to achieve alignment. Notably, Aligner requires one-off training to acquire the ability to refine outputs of base models, while *SDA* avoids explicit training entirely. By comparing *SDA* with Aligner, we can demonstrate the advantage of *SDA* in terms of alignment performance and computational efficiency.

Experiments Results

Table 1 presents the experimental results on all base models and datasets, showing *SDA*'s great advantage, effectiveness and generalization.

Universal Enhancement Across 3H Dimensions for Base Models *SDA* consistently improves the 3H performance of all base models, regardless of their scale (7B to 70B) or architecture (Llama, Vicuna, Qwen). The average gains of 64.4% in helpfulness, 30% in honesty, and 11.5% in harmlessness validate its ability to universally elevate output quality without training. Notably, even for the DeepSeek-R1-Distill-Qwen series models which have already distilled from strongly aligned teacher model (*DeepSeek-R1*)—*SDA* still achieves substantial gains, confirming its compatibility with training-based alignment methods and potential for synergistic effects. For safety-aligned models like Llama-2-Chat, the continuing improvement in harmlessness underscores *SDA*'s effectiveness in strengthening safety alignment beyond model's intrinsic capabilities.

Advantages Over Aligner Under Resource Constraints Compared to Aligner-7B, *SDA* exhibits remarkable superiority in helpfulness (average 71.7%) and honesty (average 30.9%) without relying on training data or knowledge injection. Its strength lies in activating the model's intrinsic capabilities through distributional steering. Notably, despite lacking explicit safety-aligned training, *SDA* lags behind Aligner-7B in harmlessness by only 1% on average—an insignificant gap that highlights its efficiency in balancing alignment dimensions with merely inference-level resources.

Flexible Control for Personalized Alignment On harmful datasets (Beavertails, HarmfulQA), *SDA* outperforms both base models and Aligner in *helpfulness under adversarial conditions*—a scenario requiring models to respond constructively to harmful queries. Results demonstrate *SDA*'s flexible control over generation, validating its ability to support personalized preference alignment by dynamically adjusting to instruction-specific alignment demands.

Collectively, these results confirm that *SDA* achieves effective, efficient, robust, and flexible alignment, offering a practical solution for real-world deployment where training resources are limited and personalized demands vary.

Ablation Study

To further dig into the superiority of *SDA*, we conduct ablation studies on main components of *SDA*. As is illustrated before, the effectiveness of *SDA* can be attributed to two main adjustments: (1) *Steering-Based Logit Realignment*: This component adjusts the logits based on the alignment instruction and the amplification factor, allowing the model to generate responses that are more aligned with user intent. (2) *Divergence-Aware Dynamic Temperature Scaling*: This component dynamically adjusts the temperature based on the divergence between the base distribution and the instructed distribution, reshaping the output distribution. For short, we refer to them as *Steering* and *Scaling* separately.

Ablation Study Design As *Scaling* is designed to enhance the alignment effect of *Steering*, we design two sets of comparison pairs: (1) *Steering* vs. *Base Model*: This pair compares the performance of the base model (without any alignment) with the *Steering* component, which merely adjusts the logits based on the alignment instruction and amplification factor. (2) *SDA* vs. *Steering*: This pair compares the performance of the whole *SDA* framework, which includes both

Pairs	Helpful↑	Harmless↑	Honest↑
Steering vs. *Base*	57.9%	5.3%	25.9%
SDA vs. *Steering*	17.9%	-2.1%	5.7%

Table 2: Ablation study results for the two sets of comparison pairs. The value represents the performance improvement in helpfulness, harmlessness, and honesty. Higher value indicates better performance.

Steering and *Scaling*, with the *Steering* component alone. The difference in performance between them indicates the effectiveness of the *Scaling* in enhancing the alignment effect of *Steering*. We conduct the ablation study on Llama-2-7B-Chat model, using all five datasets mentioned above. The evaluation metrics remain the same as in the main experiments, focusing on helpfulness, harmlessness, and honesty. More details are presented in Appendix B.4.

Ablation Study Results As shown in Table 2, the results demonstrate the effectiveness of both components of *SDA*: (1) *Steering* significantly improves the performance of the base model across all 3 dimensions, with an average increase of 57.9% in helpfulness, 25.9% in honesty, and 5.3% in harmlessness. This confirms that adjusting the logits based on the alignment instruction and amplification factor effectively guides the model towards more aligned responses. (2) *Scaling* further boosts the performance of *Steering* by dynamically adjusting the temperature based on the divergence between the base distribution and the instructed distribution. The average increase in helpfulness is 17.9% and in honesty is 5.7%. This indicates that reshaping the output distribution through divergence-aware scaling effectively strengthens the alignment effect of *Steering*.

Overall, the ablation study confirms that both components of *SDA* contribute significantly to its superiority in aligning model outputs with human intent. The combination of *Steering* and *Scaling* allows *SDA* to achieve effective alignment, making it a practical solution for real-world applications.

Additional Experiments Results

In addition to above experiments, we conduct several other experiments and analyses to further validate the effectiveness of *SDA*, including **Parameter Sensitivity Analysis**, **Scores Distribution Analysis** and results on other datasets, which are important as well and presented in Appendix C.

Related Work

Training-Time Alignment

Training-time alignment adjusting LLMs by optimizing their parameters based on human preferences can be broadly grouped into three categories: **(1) Reinforcement Learning and Preference Optimization** RLHF has been a cornerstone of alignment, aligning LLMs with human intent via reward signals learned from human-labeled preferences (Ouyang et al. 2022; Ji et al. 2023b), while its paradigm is resource-intensive and operationally complex, especially for large-scale deployment (Schulman et al. 2017). To simplify the process, methods like DPO (Rafailov et al. 2023) bypass explicit reward modeling by directly optimizing preference pairs, while RLAIF (Lee et al. 2023) reduce reliance on human annotation by utilizing AI-generated or ranked responses. Other advances focus on improving RM quality through uncertainty modeling (Lou et al. 2024) or optimization theory (Razin et al. 2025), and enhancing training scalability via architectural redesigns such as parameter reallocation (Mei et al. 2024). **(2) Self-Generated and Self-Supervised Tuning** To alleviate the dependence on human annotation, another class of methods leverages LLMs' generative capabilities to produce alignment data. Self-Instruct (Wang et al. 2022b) prompts model with seed instructions to generate diverse task data for fine-tuning. Later research adopts self-optimization cycles: self-refinement strategies iteratively enhance task performance in domains like code generation and correction (Ding et al. 2024; Chen et al. 2023). **(3) Hierarchical and Structured Alignment** Inspired by hierarchical control systems (Jones et al. 2025), the third stream decomposes alignment into layered processes. Online preference alignment (Bai et al. 2025) and prototype-based policy adaptation (Yi et al. 2023) use two-layer learning to respond to dynamic feedback. Metaalign (Zhang et al. 2024) employed meta-learning to learn general strategies at a high level, with task-specific layers adapting to diverse preferences.

Although the training-time methods are powerful, these designs often increase system complexity and are less practical for direct deployment (Ji et al. 2023b). In contrast, *SDA* achieves effective alignment without training, supervision, or access to model parameters. It operates entirely during inference by computing alignment-aware steering signals and dynamically adjusting output probabilities based on divergence-aware mechanisms.

Inference-Time Alignment

Inference-time alignment focuses on adjusting model outputs during inference without modifying parameters, which can be categorized roughly into two paradigms based on intervention sources: **(1) External Modular Intervention** This paradigm relies on auxiliary components to guide outputs, emphasizing "plug-and-play" flexibility without altering LLM weights. Input-level optimizations include lightweight adapters (e.g., IPA (Lu et al. 2023)), black-box prompt tuning (Cheng et al. 2023), and in-context learning mechanisms (Dong et al. 2022) where prompts or demonstrations act as virtual intervention modules. Cross-model collaboration strategies leverage small models for output refinement (Ji et al. 2024), multi-model ensembling (Jiang, Ren, and Lin 2023), and auxiliary model feedback (e.g., Inferaligner (Wang et al. 2024) for harmlessness alignment). Task-specific modules are also injected for safety (Ji et al. 2025) and personalized modeling (Zhao et al. 2025). **(2) Intrinsic Adaptive Adjustment** This paradigm activates the model's internal capabilities for self-optimization. One stream focuses on expanding the candidate space for generation, like best-of-n sampling (Huang et al. 2025), self-consistency (Wang et al. 2022a) and reward-guided tree search (Hung et al. 2024), improving alignment by enlarging

decision spaces. Self-refinement techniques (Madaan et al. 2023; Lee et al. 2025) further enhance alignment through iterative self-correction. Another stream focuses on adjusting token sampling, like temperature scaling (Chang et al. 2023; Xie et al. 2024), which reshapes output distribution to balance attribution and diversity, and adaptive sampling strategies (Meister, Cotterell, and Vieira 2020; Tang et al. 2024) optimizing token selection process.

Among all the inference-time alignment methods, Aligner (Ji et al. 2024) stands out for its effectiveness in enhancing model outputs while maintaining efficiency, and achieves state-of-the-art performance. However, it requires additional training on preference data to acquire the ability to refine model outputs, which is not always feasible in practice. In contrast, *SDA* significantly surpassing Aligner-7B in helpfulness and honesty, inherits the strengths of both paradigms while mitigating their inherent limitations: (1) It enhances the model's intrinsic ICL capability, eschewing complex external modules that would introduce training overhead; (2) It focuses on precise adjustments to token distributions rather than indiscriminately expanding the search space, thereby avoiding unnecessary forward computational waste.

Representation Engineering

Recent advances in representation engineering explore how internal activations of LLMs can be steered to modulate model behavior (Zou et al. 2023). Some methods inject learned latent vectors into hidden layers to activate specific capabilities, such as reasoning or stylistic transfer (Konen et al. 2024). Function Vectors (Todd et al. 2023) isolate and reapply capability-specific directions, while In-Context Vectors (Liu et al. 2023) summarize demonstrations into compact activations for few-shot generalization. Besides, SAE-SSV (He et al. 2025) use sparse autoencoders to identify low-dimensional, interpretable subspaces for more precise intervention. Together, these approaches bridge the gap between prompt engineering and fine-tuning by offering post-hoc, model-agnostic adjustment with minimal overhead.

SDA can be viewed as a natural extension of representation engineering and utilize the ideology into alignment. While prior work focuses on manipulating latent activations within LLMs, the output probability distribution can be seen as a highly concentrated projection of these internal representations onto the vocabulary space. Inspired by this insight, we construct a *Steering Vector*—the difference between log-output-distributions with and without alignment instructions. This approach is conceptually analogous to *in-context vector*, but operates directly on LLMs' output layer. Empirical results confirm that our output-level steering achieves effective alignment while preserving efficiency and generalization. Most similar to our work is CoS (He et al. 2024) computes distribution shifts induced by context for personalized and stylized generation, which also show great performance in style transferring and stylistic rewriting, proving the effectiveness of output-level steering.

Conclusion

In this work, we present *SDA*, a lightweight, training-free, and model-agnostic framework for aligning open LLMs during inference. By dynamically adjusting output probability distributions, *SDA* enhances alignment with human intent across the 3H dimensions without fine-tuning or modifying model weights. Extensive experiments across 8 diverse open-source LLMs demonstrate its generalization: *SDA* achieves average gains of 64.4% in helpfulness, 30% in honesty, and 11.5% in harmlessness, consistently outperforming state-of-the-art inference-time methods like Aligner-7B. Its compatibility with training-based alignment strategies and support for personalized preference alignment further validate its practical value for real-world deployment.

Limitations and Future Work

While *SDA* advances inference-time alignment, it faces several limitations that motivate future exploration: (1) *Designed for Open-Source LLMs*: *SDA* is designed for open-source LLMs that support log-probability outputs, which may limit its applicability to proprietary or closed models. Future work could explore extending *SDA* to such models by adapting the framework to operate with alternative output proxies, such as reverse-estimating token distributions for closed models (Tonolini et al. 2024). (2) *Dependence on External Scoring Models*: The current framework relies on external evaluators to obtain alignment scores, introducing dependencies on additional models and potential latency. Future work could explore self-supervised scoring mechanisms, such as leveraging the base model's intrinsic uncertainty estimates or contrastive self-evaluation, to reduce reliance and enhance flexibility. (3) *Refinement of Temperature Scaling*: The current temperature scaling mechanism applies a global adjustment to the output distribution. A more granular approach—where temperature is tuned per token or per semantic category (e.g., critical tokens (Jin et al. 2024))—could enable finer control over generation, balancing determinism and diversity for specific token types. (4) *Synergy with Training and Inference Methods*: *SDA*'s compatibility with training-based alignment (e.g., distilled models) suggests opportunities for hybrid pipelines. Future work could investigate its integration with methods like RLHF or DPO, where *SDA*-generated synthetic data could mitigate reward collapse, and with other inference-time techniques (e.g., Aligner (Ji et al. 2024)) to amplify alignment gains.

Broader Applications Beyond text alignment with 3H criteria, *SDA*'s distribution-steering paradigm can be extended to diverse scenarios: (1) *Expanding Alignment Goals*: Besides 3H, *SDA* could target specialized objectives such as logical consistency (for reasoning tasks), domain-specific accuracy (e.g., medical or legal text), or stylistic coherence. (2) *Multimodal Alignment*: The core idea of steering distributions could be adapted to other modalities, such as aligning image generation with human preferences (e.g., adjusting visual style or content harmlessness) by extending probability-based steering to vLLMs' latent spaces.

In summary, *SDA* provides a flexible foundation for efficient, training-free alignment, with promising avenues to deepen its capabilities and broaden its impact across models, modalities, and alignment goals.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Askell, A.; Bai, Y.; Chen, A.; Drain, D.; Ganguli, D.; Henighan, T.; Jones, A.; Joseph, N.; Mann, B.; DasSarma, N.; et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Bai, C.; Zhang, Y.; Qiu, S.; Zhang, Q.; Xu, K.; and Li, X. 2025. Online preference alignment for language models via count-based exploration. *arXiv preprint arXiv:2501.12735*.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bhardwaj, R.; and Poria, S. 2023. Red-teaming large language models using chain of utterances for safety-alignment. *arXiv preprint arXiv:2308.09662*.

Chang, C.-C.; Reitter, D.; Aksitov, R.; and Sung, Y.-H. 2023. KL-Divergence Guided Temperature Sampling. *ArXiv*, abs/2306.01286.

Chen, X.; Lin, M.; Schärli, N.; and Zhou, D. 2023. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*.

Chen, Y.; Liu, Y.; Chen, L.; and Zhang, Y. 2021. DialogSum: A real-life scenario dialogue summarization dataset. *arXiv preprint arXiv:2105.06762*.

Cheng, J.; Liu, X.; Zheng, K.; Ke, P.; Wang, H.; Dong, Y.; Tang, J.; and Huang, M. 2023. Black-box prompt optimization: Aligning large language models without model training. *arXiv preprint arXiv:2311.04155*.

Chu, T.; Zhai, Y.; Yang, J.; Tong, S.; Xie, S.; Schuurmans, D.; Le, Q. V.; Levine, S.; and Ma, Y. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Ding, Y.; Min, M. J.; Kaiser, G.; and Ray, B. 2024. Cycle: Learning to self-refine the code generation. *Proceedings of the ACM on Programming Languages*, 8(OOPSLA1): 392–418.

Dong, G.; Yuan, H.; Lu, K.; Li, C.; Xue, M.; Liu, D.; Wang, W.; Yuan, Z.; Zhou, C.; and Zhou, J. 2023. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Ma, J.; Li, R.; Xia, H.; Xu, J.; Wu, Z.; Liu, T.; et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Dong, X.; Teleki, M.; and Caverlee, J. 2024. A survey on llm inference-time self-improvement. *arXiv preprint arXiv:2412.14352*.

He, J. Z.-Y.; Pandey, S.; Schrum, M. L.; and Dragan, A. 2024. Context Steering: Controllable Personalization at Inference Time.

He, Z.; Jin, M.; Shen, B.; Payani, A.; Zhang, Y.; and Du, M. 2025. SAE-SSV: Supervised Steering in Sparse Representation Spaces for Reliable Control of Language Models. *ArXiv*, abs/2505.16188.

Huang, A.; Block, A.; Liu, Q.; Jiang, N.; Krishnamurthy, A.; and Foster, D. J. 2025. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. *arXiv preprint arXiv:2503.21878*.

Hung, C.-Y.; Majumder, N.; Mehrish, A.; and Poria, S. 2024. Inference time alignment with reward-guided tree search. *arXiv preprint arXiv:2406.15193*.

Ji, J.; Chen, B.; Lou, H.; Hong, D.; Zhang, B.; Pan, X.; Qiu, T.; Dai, J.; and Yang, Y. 2024. Aligner: Efficient Alignment by Learning to Correct. In *Neural Information Processing Systems*.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Chen, B.; Sun, R.; Wang, Y.; and Yang, Y. 2023a. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36: 24678–24704.

Ji, J.; Qiu, T.; Chen, B.; Zhang, B.; Lou, H.; Wang, K.; Duan, Y.; He, Z.; Zhou, J.; Zhang, Z.; et al. 2023b. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

Ji, X.; Ramesh, S. S.; Zimmer, M.; Bogunovic, I.; Wang, J.; and Ammar, H. B. 2025. Almost surely safe alignment of large language models at inference-time. *arXiv preprint arXiv:2502.01208*.

Jiang, D.; Ren, X.; and Lin, B. Y. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

Jin, L.; Peng, B.; Song, L.; Mi, H.; Tian, Y.; and Yu, D. 2024. Collaborative decoding of critical tokens for boosting factuality of large language models. *arXiv preprint arXiv:2402.17982*.

Jones, S. T.; Simpson, G. M.; Pilarski, P. M.; and Dalrymple, A. N. 2025. Hierarchical Reinforcement Learning Framework for Adaptive Walking Control Using General Value Functions of Lower-Limb Sensor Signals. *arXiv preprint arXiv:2507.16983*.

Konen, K.; Jentzsch, S.; Diallo, D.; Schutt, P.; Bensch, O.; Baff, R. E.; Opitz, D.; and Hecking, T. 2024. Style Vectors for Steering Generative Large Language Models. In *Findings*.

Lee, H.; Oh, S.; Kim, J.; Shin, J.; and Tack, J. 2025. Revise: Learning to refine at test-time via intrinsic self-verification. *arXiv preprint arXiv:2502.14565*.

Lee, H.; Phatale, S.; Mansoor, H.; Lu, K. R.; Mesnard, T.; Ferret, J.; Bishop, C.; Hall, E.; Carbune, V.; and Rastogi, A. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2310.01377*.

Lin, S.; Hilton, J.; and Evans, O. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Liu, S.; Ye, H.; Xing, L.; and Zou, J. Y. 2023. In-context Vectors: Making In Context Learning More Effective and Controllable Through Latent Space Steering. *ArXiv*, abs/2311.06668.

Lou, X.; Yan, D.; Shen, W.; Yan, Y.; Xie, J.; and Zhang, J. 2024. Uncertainty-aware reward model: Teaching reward models to know what is unknown. *arXiv preprint arXiv:2410.00847*.

Lu, X.; Brahman, F.; West, P.; Jang, J.; Chandu, K.; Ravichander, A.; Qin, L.; Ammanabrolu, P.; Jiang, L.; Ramnath, S.; et al. 2023. Inference-time policy adapters (ipa): Tailoring extreme-scale lms without fine-tuning. *arXiv preprint arXiv:2305.15065*.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Welleck, S.; Majumder, B. P.; Gupta, S.; Yazdanbakhsh, A.; and Clark, P. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *ArXiv*, abs/2303.17651.

Mei, Z.; Fu, W.; Li, K.; Wang, G.; Zhang, H.; and Wu, Y. 2024. Real: Efficient rlhf training of large language models with parameter reallocation. *arXiv preprint arXiv:2406.14088*.

Meister, C.; Cotterell, R.; and Vieira, T. 2020. Best-First Beam Search. *Transactions of the Association for Computational Linguistics*, 8: 795–809.

Mundra, N.; Doddapaneni, S.; Dabre, R.; Kunchukuttan, A.; Puduppully, R.; and Khapra, M. M. 2024. A comprehensive analysis of adapter efficiency. In *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, 136–154.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741.

Rashkin, H.; Smith, E. M.; Li, M.; and Boureau, Y.-L. 2018. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207*.

Razin, N.; Wang, Z.; Strauss, H.; Wei, S.; Lee, J. D.; and Arora, S. 2025. What makes a reward model a good teacher? an optimization perspective. *arXiv preprint arXiv:2503.15477*.

Sahoo, P.; Singh, A. K.; Saha, S.; Jain, V.; Mondal, S.; and Chadha, A. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Tang, C.; Liu, J.; Xu, H.; and Huang, L. 2024. Top-nσ: Not All Logits Are You Need. *ArXiv*, abs/2411.07641.

Todd, E.; Li, M.; Sharma, A. S.; Mueller, A.; Wallace, B. C.; and Bau, D. 2023. Function Vectors in Large Language Models. *ArXiv*, abs/2310.15213.

Tonolini, F.; Aletras, N.; Massiah, J.; and Kazai, G. 2024. Bayesian prompt ensembles: Model uncertainty estimation for black-box large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, 12229–12272.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, P.; Zhang, D.; Li, L.; Tan, C.; Wang, X.; Ren, K.; Jiang, B.; and Qiu, X. 2024. Inferaligner: Inference-time alignment for harmlessness through cross-model guidance. *arXiv preprint arXiv:2401.11206*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E. H.; and Zhou, D. 2022a. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *ArXiv*, abs/2203.11171.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2022b. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Xie, J.; Chen, A. S.; Lee, Y.; Mitchell, E.; and Finn, C. 2024. Calibrating Language Models with Adaptive Temperature Scaling. *ArXiv*, abs/2409.19817.

Xu, S.; Fu, W.; Gao, J.; Ye, W.; Liu, W.; Mei, Z.; Wang, G.; Yu, C.; and Wu, Y. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

Yi, Q.; Zhang, R.; Peng, S.; Guo, J.; Gao, Y.; Yuan, K.; Chen, R.; Lan, S.; Hu, X.; Du, Z.; et al. 2023. Online prototype alignment for few-shot policy transfer. In *International Conference on Machine Learning*, 39968–39983. PMLR.

Zhang, M.; Wang, P.; Tan, C.; Huang, M.; Zhang, D.; Zhou, Y.; and Qiu, X. 2024. Metaalign: Align large language models with diverse preferences during inference time. *arXiv preprint arXiv:2410.14184*.

Zhao, W.; Sui, X.; Hu, Y.; Guo, J.; Liu, H.; Li, B.; Zhao, Y.; Qin, B.; and Liu, T. 2025. Teaching Language Models to Evolve with Users: Dynamic Profile Modeling for Personalized Alignment. *arXiv preprint arXiv:2505.15456*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36: 46595–46623.

Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.-K.; Goel, S.; Li, N.; Byun, M. J.; Wang, Z.; Mallen, A. T.; Basart, S.; Koyejo, S.; Song, D.; Fredrikson, M.; Kolter, Z.; and Hendrycks, D. 2023. Representation Engineering: A Top-Down Approach to AI Transparency. *ArXiv*, abs/2310.01405.