Entropy Controllable Direct Preference Optimization

Anonymous authors
Paper under double-blind review

Abstract

In the post-training of large language models (LLMs), Reinforcement Learning from Human Feedback (RLHF) is an effective approach to achieve generation aligned with human preferences. Direct Preference Optimization (DPO) allows for policy training with a simple binary cross-entropy loss without a reward model. The objective of DPO is regularized by reverse KL divergence that encourages mode-seeking fitting to the reference policy. Nonetheless, we indicate that minimizing reverse KL divergence could fail to capture a mode of the reference distribution, which may hurt the policy's performance. Based on this observation, we propose a simple modification to DPO, H-DPO, which allows for control over the entropy of the resulting policy, enhancing the distribution's sharpness and thereby enabling mode-seeking fitting more effectively. In our experiments, we show that H-DPO outperformed DPO across various tasks, demonstrating superior results in pass@k evaluations for mathematical tasks. Moreover, H-DPO is simple to implement, requiring only minor modifications to the loss calculation of DPO, which makes it highly practical and promising for wide-ranging applications in the training of LLMs.

1 Introduction

Large language models (LLMs) have exhibited remarkable performance across various tasks (OpenAI et al., 2023; Dubey et al., 2024). However, large datasets often include data created for various purposes, and the models trained on these datasets are not always suitable for users' specific needs. Additionally, some datasets include malicious text and code related to cyberattacks, posing risks of misuse by humans or the AI itself (Bender et al., 2021; Bai et al., 2022; Ji et al., 2023; Shevlane et al., 2023).

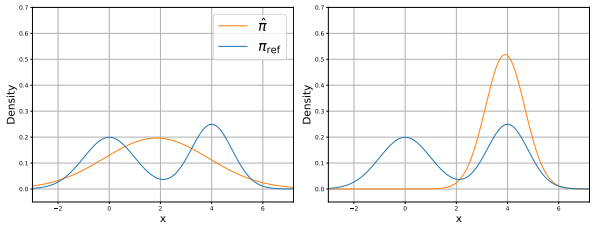
Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Bai et al., 2022) is an effective approach to make an LLM follow human instructions and suppressing undesired outputs. In RLHF, a reward model is trained based on data evaluated according to human preferences. The LLM then learns to maximize rewards, aligning its outputs with human preferences. To prevent significant deviation from the original model, regularization using reverse KL divergence is added to the reward maximization process, and RL algorithms such as PPO (Schulman et al., 2017) are employed.

However, RLHF has issues such as high computational costs, the reliance on a learned reward model, and the inherent instability and hyperparameter sensitivity of RL algorithms. To address these problems, Direct Policy Optimization (DPO) (Rafailov et al., 2023) has emerged and is now widely used. DPO proposes a loss function that directly optimizes the policy through a change of variables, eliminating the need for the reward model and allowing training with a simple binary cross-entropy loss. While more stable and lightweight than RLHF, DPO can optimize the same objective function as RLHF, which involves reward maximization and regularization with the reverse KL divergence. Other types of divergences have also been proposed to prevent deviation from the original model (Wang et al., 2024a), but reverse KL divergence, which enables mode-seeking estimation, is generally preferred for performance.

We point out that minimizing reverse KL divergence can cause the mode of the fitted distribution to fail to capture the mode of the target distribution. As shown in Figure 1, consider fitting a unimodal distribution to a multimodal distribution. We call the way of fitting a distribution *mode-seeking* when one of the modes of target distribution is captured by the fitted model as shown in the right side of Figure 1, and *mode-covering*

when all the modes are covered as shown in the left side of Figure 1. In the case of mode-seeking, the fitted distribution discards other modes of the target distribution, resulting in smaller variance than the target distribution. However, reverse KL minimization can fail at mode-seeking fitting due to its nature of preserving variance, as illustrated in the left side of Figure 1.

To enable variance reduction and encourage mode-seeking estimation, we generalize the loss function of DPO, named H-DPO, which allows for controlling the distribution's entropy $H(\pi)$ by modifying the regularization term. H-DPO can adjust the entropy of generations of the LLM during training using the hyperparameter α in Equation (9) introduced later. By setting α less than 1, it encourages the entropy to be reduced so that achieves mode-seeking fitting more successfully. The right side of Figure 1 demonstrates that our regularizer D_α, a modification to the reverse KL, enables mode-seeking fitting even in cases where reverse KL fails, as shown on the left.



$\min_\pi D_{\text{KL}}(\pi||\pi_{\text{ref}})$ $\min_\pi D_\alpha(\pi||\pi_{\text{ref}})$

Figure 1: For a Gaussian mixture model π_{ref}, $\hat{\pi}$ that minimizes D_{KL} (left) and $\hat{\pi}$ that minimizes $D_\alpha = -\alpha H(\pi) + H(\pi, \pi_{\text{ref}})$ with $\alpha = 0.6$ (right). Using D_α results in successful mode-seeking estimation.

Using our proposed loss with $\alpha < 1$, the estimated policy distribution is expected to be sharper or more deterministic, which we consider a beneficial feature rather than a problem. Traditional LLMs use a softmax function with a temperature parameter to represent distributions over raw outputs, where the temperature is set to 1 during training. When LLMs are evaluated, a lower value such as 0.6 often performs better (Xu et al., 2022; OpenAI et al., 2023; Zhu et al., 2024). This post-training sharpening lacks guarantees of optimality for the objective function. In contrast, our proposed method trains the language model using an objective function aimed at sharpening the distribution, ensuring that this sharper distribution aligns with the objective function.

Our main contribution is the alignment method H-DPO, which allows controlling entropy and encourages mode-seeking fitting more than DPO. The implementation of H-DPO is simple, requiring minimal modifications to DPO. Experiments included alignment based on Mistral-7B (Jiang et al., 2023) with the Zephyr framework (Tunstall et al.; 2023). Compared to DPO, our proposed method allows for more diverse generations without losing performance, and shows superior accuracy and coverage across various tasks.

2 Related Work

Alignment Language models trained through next-token prediction have rapidly advanced and show strong performance on many tasks in zero-shot or few-shot settings (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2023). Fine-tuning using human preferences and instructions, known as alignment, has proven effective in improving instruction following and reducing harmful outputs (Christiano et al., 2017; Bai et al., 2022; Touvron et al., 2023; Ouyang et al., 2022). A prominent method for alignment is RLHF; however, it encounters issues such as high computational costs, significant memory requirements, and the instability of reinforcement learning (Schulman et al., 2017; Engstrom et al., 2020; Ahmadian et al., 2024). To address these issues, DPO (Rafailov et al., 2023) has been proposed. DPO eliminates the need to model the reward function and employ reinforcement learning algorithms, evolving in various directions (Liu et al., 2024; Gheshlaghi Azar et al., 2024; Song et al., 2024). Wang et al. (2024a); Zeng et al. (2024) enable adjusting the diversity of generated responses by changing the regularization of reverse KL. Wang et al. (2024a) extends this to f-divergence other than reverse KL divergence, arguing that adjusting α in α-divergence allows for a trade-off between diversity and performance. As α-divergence interpolates between reverse KL and forward KL, using larger α makes the mode-seeking property diminish, which may increase diversity but deteriorate performance. Our study proposes a different method to balance diversity and performance while maintaining or strengthening the mode-seeking property of reverse KL.

Diversity in Language Models The importance of diversity in the responses generated by language models has been emphasized in numerous studies. Achieving more diverse text generation with high quality is crucial, despite the existing quality-diversity trade-off (Nenkova et al., 2007; Clarke et al., 2008; Hashimoto et al., 2019; Zhang et al., 2021). Diversity can be adjusted through various methods, such as sampling-based techniques like changing the temperature (Fan et al., 2018; Holtzman et al., 2020; Wang et al., 2024b), manipulating prompts (Arora et al., 2023; Li et al., 2023), or during DPO as mentioned in the previous section (Wang et al., 2024a; Zeng et al., 2024). Studies such as Wang et al. (2024a); Zeng et al. (2024) have examined changes in diversity due to objective functions in post-training, but have not considered the impact of temperature adjustments, which are commonly manipulated when using language models. Our study investigates the effects of both objective function modifications in post-training and temperature adjustments on diversity.

In recent LLMs, there has been a growing emphasis not only on the accuracy of a single response but also on the coverage — the fraction of problems solved by any generated sample (Kulal et al., 2019; Chen et al., 2021; Roziere et al., 2023; Brown et al., 2024). In such evaluations, diversity in the generated outputs contributes to improved coverage (Wang et al., 2024b). The importance of coverage is partly due to the presence of verifiers that can assess the correctness of generated answers, particularly in mathematical and coding tasks. These verifiers allow for selecting correct outputs from multiple candidates as the answer. Some studies (Kulal et al., 2019; Chen et al., 2021; Roziere et al., 2023) have demonstrated significant improvements in correctness through repeated sampling in coding tasks, while Brown et al. (2024) showed that even relatively lightweight models could outperform frontier models in coverage by increasing the number of generated samples in mathematical tasks. In tasks such as chat, where precise verification is challenging, the performance can still be enhanced through methods such as majority voting (Wang et al., 2023) or by using reward models and trained verifiers (Cobbe et al., 2021; Lightman et al., 2024; Hosseini et al., 2024; Wang et al., 2024c; Kang et al., 2024) to use repeated samples effectively.

Wang et al. (2024b) explored the relationship between diversity and coverage, demonstrating that greater diversity in generated outputs leads to a more significant improvement in coverage for larger values of k in pass@k, which denotes the probability that the correct answer is included in the k generated outputs. Our study shows that using the proposed objective can increase diversity while maintaining a certain level of accuracy, achieving favorable performance in pass@k evaluations.

Mode-Seeking and Mode-Covering When minimizing a certain divergence to bring two probability distributions closer, attention is often given to whether the fitting or divergence is mode-seeking or mode-covering (mass-covering) (Huszár, 2015; Shannon et al., 2020; Ke et al., 2021; Li & Farnia, 2023; Wang et al., 2024a). When fitting a distribution to a multimodal distribution, if the fitted distribution captures one of the modes, this fitting is called mode-seeking. If it covers all the modes, it is termed mode-covering. Accordingly, divergences facilitating such fittings when minimized are similarly referred to as mode-seeking and mode-covering divergences, respectively. The reverse KL divergence, which is used in RLHF and DPO training, is considered mode-seeking compared to forward KL and other f-divergence Shannon et al. (2020); Li & Farnia (2023). Policy learning using mode-seeking divergence often performs better than mode-covering divergence (Ke et al., 2021; Wang et al., 2024a). In this study, we propose a new regularizer to replace the minimization of reverse KL divergence in the objective function of DPO, aiming to achieve better performance through enhanced mode-seeking property.

3 Preliminaries

3.1 Reinforcement Learning from Human Feedbacks (RLHF)

In the context of LLM training, RLHF is a process of aligning an LLM to human preferences after pre-training, typically consisting of three steps: supervised fine-tuning (SFT), reward modeling, and RL fine-tuning.

Supervised Fine-Tuning (SFT) SFT is the process of adapting an already pre-trained LLM to specific tasks by optimizing the model parameters using a task-specific dataset. Using high-quality data related to the task, the model is optimized through supervised learning to obtain π_{SFT}.

Reward Modeling Next, a reward model is trained to reflect human preferences in RL. Let $r_\phi(x, y)$ be a reward model parameterized by ϕ, where x is a prompt and y is a completion. It is typically assumed that human preference for a pair of completions follows the Bradley-Terry (BT) model (Bradley & Terry, 1952), where the probability of preferring y_1 to y_2 is represented using a difference of rewards:

$$p(y_1 \succ y_2 \mid x) = \sigma(r(x, y_1) - r(x, y_2)), \tag{1}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is a sigmoid function. The larger the value of $r(x, y)$ is, the more preferable a completion y is to a prompt x.

Using a labeled dataset $\mathcal{D} = \{x^i, y_w^i, y_l^i\}_{i=0}^N$ with user preferences, where y_w^i is preferred to y_l^i for prompt x^i, the loss function for training the reward model is formulated by minimizing the negative log-likelihood:

$$L(r_\phi) = -\mathbb{E}_{(x,y_w,y_l\sim\mathcal{D})}[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))]. \tag{2}$$

RL Fine-Tuning Finally, the language model is fine-tuned, using the trained reward model, to maximize the following objective function:

$$J(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta}\left[r_\phi(x, y)\right] - \beta D_{\mathrm{KL}}(\pi_\theta(y \mid x)||\pi_{\mathrm{ref}}(y \mid x)), \tag{3}$$

where β is a hyperparameter that controls the deviation from π_{ref}. π_θ is trained to maximize the reward while being regularized by the reverse KL divergence to not deviate too much from π_{ref}. Typically, π_{ref} is fixed to π_{SFT} while π_θ is initialized with π_{SFT}.

3.2 Directed Preference Optimization (DPO)

In RLHF, the need to train the reward model and apply an online RL algorithm such as PPO imposes significant computational and memory costs. DPO suggests a method for directly learning to reflect human preferences in a supervised manner without using the reward model by mapping language model policies and reward functions. The objective function is equivalent to that of RLHF, and the optimal policy that maximizes Equation (3) when the reward model is optimal is derived as follows:

$$\pi^*(y \mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{r^*(x, y)}{\beta}\right), \tag{4}$$

where $Z(x)$ is the partition function. From this equation, the optimal reward can be expressed using the optimal policy:

$$r^*(x, y) = \beta \log \frac{\pi^*(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta \log Z(x). \tag{5}$$

Using this optimal reward function to calculate the probability distribution of the BT model, the computationally challenging partition function $Z(x)$ cancels out as follows:

$$p^*(y_1 \succ y_2 \mid x) = \sigma\left(\beta \log \frac{\pi^*(y_1 \mid x)}{\pi_{\mathrm{ref}}(y_1 \mid x)} - \beta \log \frac{\pi^*(y_2 \mid x)}{\pi_{\mathrm{ref}}(y_2 \mid x)}\right). \tag{6}$$

The loss function for π_θ is derived as the maximum likelihood estimation of the BT model from a human preference dataset $\mathcal{D}$:

$$L_{\mathrm{DPO}} = -\mathbb{E}_{x,y_w,y_l\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)} - \beta \log \frac{\pi_{\mathrm{ref}}(y_w \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right]. \tag{7}$$

Thus, DPO can align language models with human preferences without learning a reward model.

4 Entropy Controllable Directed Preference Optimization

In DPO, reverse KL divergence is used as a regularizer that controls the deviation from π_{ref}. The reverse KL divergence is defined as $D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}) = \int \pi_\theta(y \mid x) \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} dy$. Here, the integrand is zero for regions where $\pi_\theta(y \mid x) = 0$, meaning that only the regions supported by $\pi_\theta(y \mid x)$ affect the divergence. Consequently, fitting by minimizing the reverse KL divergence is known to be mode-seeking and generally performs better than other divergences such as forward KL (Ke et al., 2021; Wang et al., 2024a).

However, in this study, we discuss cases where even using reverse KL divergence can fail to achieve mode-seeking fitting with respect to the target distribution. We verify such cases through preliminary experiments and show that controlling the entropy of the distribution enables more effective mode-seeking fitting. To control the entropy of the output probability by language models in DPO, we propose H-DPO, which incorporates such entropy-controllable optimization into DPO.

4.1 Mode-seeking Property

As a preliminary experiment on the mode-seeking property of reverse KL divergence, we fit a Gaussian distribution to a mixture of two Gaussian components. Specifically, given a Gaussian mixture model π_{ref}, we compute the location and scale parameters of a Gaussian distribution π that minimize the reverse KL divergence $D_{\text{KL}}(\pi||\pi_{\text{ref}})$. If the fitting is mode-seeking, the estimated Gaussian distribution should capture one of the components of the mixture model. However, as shown in Figure 1, despite the reverse KL minimization, which is supposed to have the mode-seeking property, the fitting may look mode-covering, not mode-seeking. In this case, if π is a language model, it is likely to generate from valleys where π_{ref} has a low probability, possibly leading to degraded performance of π.

The cause of such mode-covering fitting could be the inherent property of reverse KL divergence minimization, which aims to preserve some variance. If π captures only one component, its variance should be smaller compared to π_{ref} as a whole because it must ignore the other component. As shown on the left side of Figure 1, however, reverse KL minimization does not take this into account, resulting in mode-covering estimation.

We consider an objective that can reduce variance or entropy as a remedy. To adjust the entropy of π, we note that the reverse KL divergence can be decomposed into entropy and cross-entropy components as follows:

$$\begin{aligned} D_{\text{KL}}(\pi||\pi_{\text{ref}}) &= \int (\pi(x) \log \pi(x) - \pi(x) \log \pi_{\text{ref}}(x)) dx \\ &= -H(\pi) + H(\pi, \pi_{\text{ref}}). \end{aligned} \tag{8}$$

By attaching a coefficient α to the entropy $H(\pi)$, we can derive another objective that can control entropy: $D_\alpha = -\alpha H(\pi) + H(\pi, \pi_{\text{ref}})$[†]. By making α less than 1, we can reduce the entropy while fitting between distributions. The right side of Figure 1 shows the distribution π that minimizes D_α as α decreases from 1. By reducing α from 1 to a smaller value, it can achieve the mode-seeking fitting. Details of the preliminary experiments related to Figure 1 are provided in Appendix A.1.

The effectiveness of the mode-seeking property has been verified in Wang et al. (2024a), and strengthening the mode-seeking property by reducing α is an attractive feature. However, even in cases where π and π_{ref} have the same number of modes (e.g., when both are unimodal distributions), allowing π to fit π_{ref} with D_α can result in π becoming a sharper distribution than π_{ref}. Although this might seem problematic, it could be beneficial in language model training. For better performance at inference time, the sampling temperature is often set below 1 (Xu et al., 2022; OpenAI et al., 2023; Zhu et al., 2024). This means the distribution learned at a temperature of 1 is sharpened by reducing the temperature. However, there is no guarantee that the sharpened distribution is optimal for the DPO objective function. The distribution learned by maximizing our objective function with a small α also becomes sharp, but unlike adjusting sampling temperature at inference time, it becomes sharp in a manner consistent with the objective function. The following section introduces how to incorporate such entropy adjustment using α into DPO.

[†]Note that, for $\alpha \neq 1$, $D_\alpha(p||q)$ is not a divergence because it is not zero even when $p = q$.

4.2 H-DPO

As discussed in the previous section, by decomposing the reverse KL divergence into its entropy and cross-entropy components, we can adjust the entropy with α. The objective function for DPO with entropy adjustment is shown below:

$$\begin{aligned} J_{\text{H-DPO}} &= \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}\left[r(x,y)\right] - \beta D_{\alpha}(\pi||\pi_{\text{ref}}) \\ &= \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}\left[r(x,y)\right] + \alpha\beta H(\pi) - \beta H(\pi,\pi_{\text{ref}}). \end{aligned} \tag{9}$$

Here, when α equals 1, it becomes the same objective function as that of standard DPO. By setting α to be smaller than 1, the learning process aims to reduce the entropy. Similar to Wang et al. (2024a), we consider a constrained optimization. By applying Lagrange multipliers under the constraints that π is a probability distribution, i.e., $\sum_y \pi(y \mid x) = 1$ and $\forall y, \pi(y \mid x) \geq 0$, we obtain the following:

$$\mathcal{L}(\pi,\lambda,C) = \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}[r(x,y) - \alpha\beta\log\pi(y \mid x) + \beta\log\pi_{\text{ref}}(y \mid x)] - \lambda\left(\sum_y \pi(y \mid x) - 1\right) - \sum_y C(y)\pi(y \mid x), \tag{10}$$

where λ and C are the dual variables. Solving this problem, the optimal policy π^* can be derived as

$$\pi^*(y \mid x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y \mid x)^{1/\alpha}\exp\left(\frac{r^*(x,y)}{\alpha\beta}\right). \tag{11}$$

From this equation, the reward function can be expressed using the policy as follows:

$$r^*(x,y) = \alpha\beta\log\pi^*(y \mid x) - \beta\log\pi_{\text{ref}}(y \mid x) + \alpha\beta\log Z(x). \tag{12}$$

When applying this reward function to the BT model and performing the maximum likelihood estimation, the loss function using α is

$$L_{\text{H-DPO}} = -\mathbb{E}_{x,y_w,y_l\sim\mathcal{D}}\left[\log\sigma\left(\alpha\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)} - \beta\log\frac{\pi_{\text{ref}}(y_w \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]. \tag{13}$$

Comparing this equation to the DPO loss function in Equation (7), we can see that entropy adjustment using α can be implemented by simply replacing the coefficient for π_θ from β to $\alpha\beta$.

5 Experiments

In this section, we evaluate the performance of H-DPO in comparison to standard DPO using widely recognized metrics.

5.1 Experimental Setup

We conducted DPO training based on Zephyr-7B-Beta (Tunstall et al., 2023; Tunstall et al.). We started from zephyr-7b-sft-full, which is based on Mistral 7B (Jiang et al., 2023) and fine-tuned with UltraChat (Ding et al., 2023). We performed DPO training on it with UltraFeedback (Cui et al., 2023). We evaluated the performance when H-DPO was used instead of standard DPO. The hyperparameters during training were the same as those of Zephyr-7B-beta, except for the variable α. The α was varied in the range from 0.8 to 1.2.

The evaluation tasks included diverse grade school math word problems (GSM8K (Cobbe et al., 2021)), coding task (HumanEval (Chen et al., 2021)), multiple-choice question task (MMLU-Pro (Wang et al., 2024d)) and instruction-following task (IFEval (Zhou et al., 2023)). The training was conducted with three different seeds. Further experimental details are provided in Appendix A.3 and A.4.

5.2 Performance and Diversity

Table 1 shows the scores for each task when α was decreased. By reducing α by 0.05 to 0.1, performance improved on all tasks compared to the conventional DPO ($\alpha = 1$).

Table 2 presents diversity metrics when α was varied in H-DPO. When the temperature was set to 1, smaller α values resulted in lower diversity, while larger α values increased diversity. This indicates that diversity can be controlled through α. However, it should be noted that diversity changes with temperature, and the optimal temperature varies depending on the value of α. Hence, even with a smaller α, diversity could be increased if a higher temperature is used.

For MMLU-Pro, the scores and entropy with varying temperatures are shown in Figure 2. The left figure illustrates the relationship between temperature and score, highlighting that smaller α values exhibit less performance degradation and greater robustness to temperature selection. This is because entropy remains low even when a higher temperature is used. The right figure shows the relationship between entropy and score, where the entropy of the samples obtained at each temperature replaces the temperature shown in the left figure. At the same score point, the entropy is larger when α is smaller. In other words, with a smaller α, it is possible to achieve more diverse generations even with the same performance.

Table 1: Average scores of DPO and H-DPO with different α values on various tasks.

	GSM8K↑	HumanEval↑	MMLU-Pro↑	IFEval↑
DPO ($\alpha = 1$)	26.40 ± 1.76	28.77 ± 0.45	31.83 ± 0.17	59.63 ± 0.72
H-DPO ($\alpha = 0.95$)	27.77 ± 1.39	**30.70** ± 0.39	**32.37** ± 0.03	60.17 ± 0.34
H-DPO ($\alpha = 0.9$)	**28.83** ± 2.32	29.63 ± 0.45	32.30 ± 0.17	**60.93** ± 0.50
H-DPO ($\alpha = 0.8$)	28.66 ± 1.23	27.77 ± 0.67	31.93 ± 0.19	59.90 ± 0.59

Table 2: Comparison of DPO and H-DPO with various α values across different diversity metrics when temperature is 1.

	Entropy↑	Self-Bleu↓	Distinct-1↑	Distinct-2↑
H-DPO ($\alpha = 1.2$)	1.718	0.252	0.313	0.690
H-DPO ($\alpha = 1.1$)	1.483	0.293	0.296	0.652
DPO ($\alpha = 1$)	1.323	0.326	0.289	0.633
H-DPO ($\alpha = 0.95$)	1.223	0.339	0.277	0.611
H-DPO ($\alpha = 0.9$)	1.113	0.364	0.272	0.590
H-DPO ($\alpha = 0.8$)	0.977	0.391	0.268	0.574

5.3 Coverage Evaluation

As mentioned in the previous section, a smaller α enabled more diverse outputs at the same performance level. Wang et al. (2024b) demonstrated that high diversity positively impacts coverage performance, where coverage is evaluated using the pass@k metric. Coverage refers to the fraction of problems that can be solved using any generated sample, and pass@k is the coverage achieved by using k samples (Kulal et al., 2019; Chen et al., 2021). Chen et al. (2021) proposed an unbiased and stable calculation method for pass@k metric, which is employed in our study. Coverage is particularly significant in tasks where correctness evaluation is relatively straightforward, such as mathematical and coding tasks; hence, evaluations were conducted on the GSM8K (math task) and HumanEval (coding task).

Figure 3 presents the pass@k evaluation results for various k values in GSM8K. Overall, reducing α leads to better performance than standard DPO ($\alpha = 1$). In standard DPO, for most values of k, the best coverage when varying the temperature is achieved at a temperature of 0.5, which is smaller than the value of 1 used during training. However, for smaller α values (e.g., $\alpha = 0.8$), the best coverage is achieved with the same training temperature of 1 when k is large. This implies that decreasing α ($\alpha = 0.8$) and using a temperature close to that used during training provides better results than simply lowering the temperature in standard

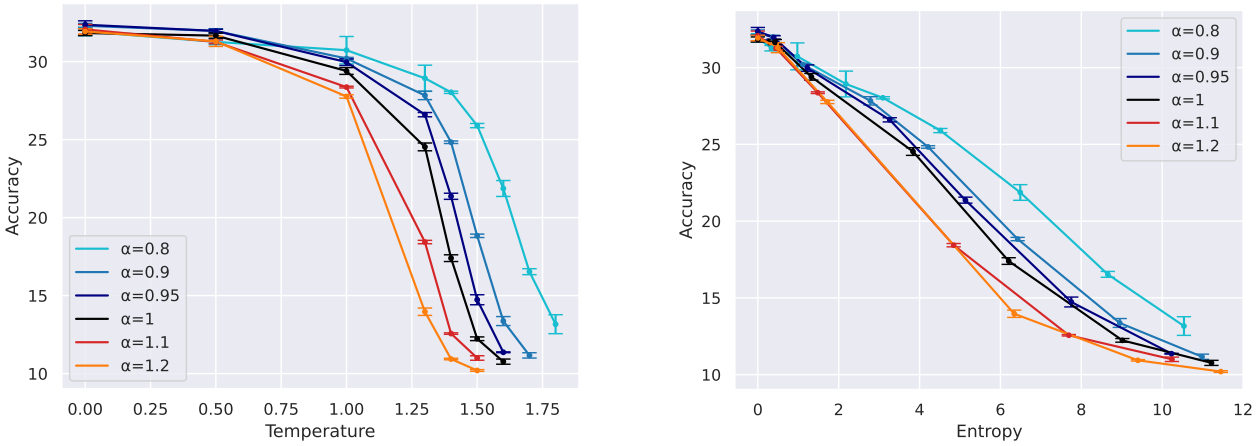
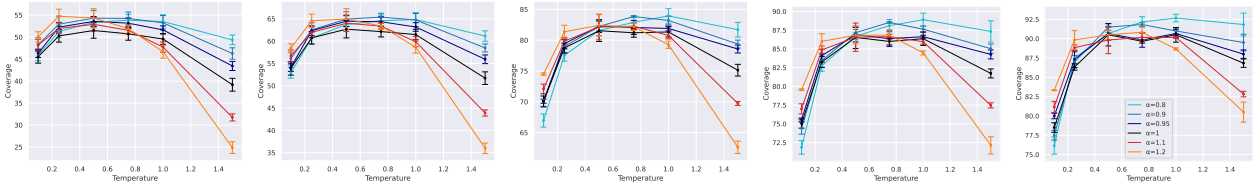


Figure 2: **Left**: Accuracy on **MMLU-Pro** at various temperatures. **Right**: Accuracy on **MMLU-Pro** at various entropy levels. The horizontal axis of the left figure is replaced with the entropy obtained from sampling at each corresponding temperature.



pass@5 pass@10 pass@50 pass@100 pass@200

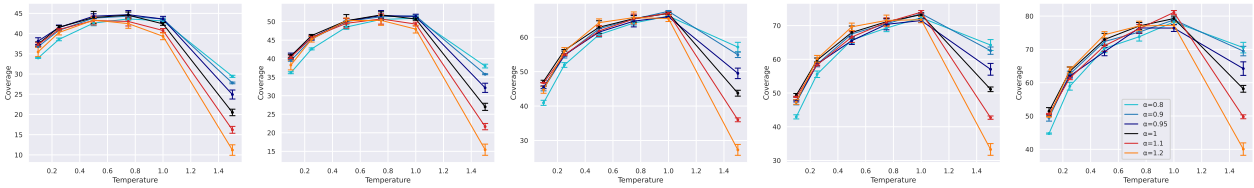
Figure 3: Coverage (pass@k) of H-DPO and DPO with various temperatures on **GSM8K**.

DPO. This suggests that H-DPO, which allows using a model closer to the one used during training even at test time, is superior to standard DPO in this setting.

Figure 4 presents the evaluation results of pass@k for various values of k on the HumanEval benchmark. On HumanEval, there is a negligible difference between models with a small α and standard DPO when k is large. However, interestingly, when k exceeds 100, the results improve for larger α values ($\alpha = 1.1$). This suggests that in the evaluation of coverage on HumanEval, the diversity of responses is more crucial than their quality. In standard DPO, as k increases, the best coverage is shown with a temperature of 1. When the temperature exceeds 1, which is higher than the temperature during training, performance drops significantly, indicating that the maximum effective temperature for increasing diversity is 1. As shown in Table 2, increasing α enhances diversity at a temperature of 1. Consequently, for tasks where diversity at a temperature of 1 is crucial, increasing α can improve coverage.

6 Conclusion

In this study, we proposed H-DPO, a generalization of DPO, and thoroughly examined its effectiveness. H-DPO allows for the adjustment of entropy during training through the hyperparameter α, enabling the control of distribution sharpness and achieving more effective mode-seeking fitting compared to standard DPO. This new method allows trained models to generate more accurate and diverse outputs, better aligning with their intended purposes. In the experiments, we aligned Mistral-7B-based models using the proposed method and compared them with standard DPO. H-DPO demonstrated superior performance compared to DPO across various tasks. In mathematical tasks, it showed excellent performance in pass@k evaluations. These results confirmed that the diversity and quality of the generated outputs improved, establishing H-DPO as a powerful method for improving the training process of LLMs. Moreover, H-DPO is extremely simple to implement, requiring only minor modifications to existing DPO, which adds to its practicality and potential



pass@5 pass@10 pass@50 pass@100 pass@200

Figure 4: Coverage (pass@k) of H-DPO and DPO with various temperatures on **HumanEval**.

for widespread application. The need to adjust α is a limitation of this method, and automating the search of appropriate α values for each task can be a focus of future research.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662.

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=bhUPJnS2g0X.

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

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pp. 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/2334029.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024. URL https://arxiv.org/abs/2407.21787.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. URL `http://jmlr.org/papers/v24/22-1144.html`.

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

Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pp. 659–666, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581644. doi: 10.1145/1390334.1390446. URL `https://doi.org/10.1145/1390334.1390446`.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations, 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=r1etN1rtPB`.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL `https://aclanthology.org/P18-1082`.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4447–4455. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/gheshlaghi-azar24a.html.

Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1689–1701, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1169. URL https://aclanthology.org/N19-1169.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STar: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=stmqBSW2dV.

Ferenc Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary?, 2015. URL https://arxiv.org/abs/1511.05101.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Jikun Kang, Xin Zhe Li, Xi Chen, Amirreza Kazemi, and Boxing Chen. Mindstar: Enhancing math reasoning in pre-trained llms at inference time. *arXiv preprint arXiv:2405.16265*, 2024.

Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. In Steven M. LaValle, Ming Lin, Timo Ojala, Dylan Shell, and Jingjin Yu (eds.), *Algorithmic Foundations of Robotics XIV*, pp. 313–329, Cham, 2021. Springer International Publishing.

Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy S Liang. Spoc: Search-based pseudocode to code. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7298332f04ac004a0ca44cc69ecf6f6b-Paper.pdf.

Cheuk Ting Li and Farzan Farnia. Mode-seeking divergences: Theory and applications to gans. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 8321–8350. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/ting-li23a.html.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://aclanthology.org/N16-1014.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5315–5333, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.291. URL https://aclanthology.org/2023.acl-long.291.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xbjSwwrQOe.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2):4–es, May 2007. ISSN 1550-4875. doi: 10.1145/1233912.1233913. URL https://doi.org/10.1145/1233912.1233913.

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

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

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

Matt Shannon, Ben Poole, Soroosh Mariooryad, Tom Bagby, Eric Battenberg, David Kao, Daisy Stanton, and RJ Skerry-Ryan. Non-saturating gan training as divergence minimization, 2020. URL https://arxiv.org/abs/2010.08029.

Toby Shevlane, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, et al. Model evaluation for extreme risks. *arXiv preprint arXiv:2305.15324*, 2023.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18990–18998, 2024.

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

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alvaro Bartolome, Alexander M. Rush, and Thomas Wolf. The Alignment Handbook. URL https://github.com/huggingface/alignment-handbook.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse KL: Generalizing direct preference optimization with diverse divergence constraints. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=2cRzmWXK9N.

Evan Wang, Federico Cassano, Catherine Wu, Yunfeng Bai, Will Song, Vaskar Nath, Ziwen Han, Sean Hendryx, Summer Yue, and Hugh Zhang. Planning in natural language improves llm search for code generation. *arXiv preprint arXiv:2409.03733*, 2024b.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024c.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024d.

Frank F Xu, Uri Alon, Graham Neubig, and Vincent Josua Hellendoorn. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, pp. 1–10, 2022.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58348–58365. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zeng24c.html.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. Trading off diversity and quality in natural language generation. In Anya Belz, Shubham Agarwal, Yvette Graham, Ehud Reiter, and Anastasia Shimorina (eds.), *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pp. 25–33, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.humeval-1.3.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pp. 1097–1100, New York, NY, USA,

2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.

Yuqi Zhu, Jia Li, Ge Li, YunFei Zhao, Zhi Jin, and Hong Mei. Hot or cold? adaptive temperature sampling for code generation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 437–445, 2024.

A Experimental Details

A.1 Preliminary Experiment with Gaussian Distribution

This section details the experiments shown in Figure 1. In this preliminary experiment, we use the proposed regularization $D_\alpha(\pi||\pi_{\text{ref}})$, where D_α is the same as the Kullback-Leibler divergence D_{KL} when $\alpha = 1$, to estimate the Gaussian distribution π that is closest to the Gaussian Mixture Model (GMM) π_{ref}. The experiments were conducted with GMMs π_{ref} consisting of 2, 3, and 4 Gaussian components, and the results are shown in Figure 5, 7 and 8, respectively. For any π_{ref}, the weights of the components are equal, and the standard deviations are 1 and 0.8 for the case of two components, 1, 0.8, and 0.5 for the case of three components, and 1, 0.8, 0.5, and 0.3 for the case of four components. In those figures, the results of varying the interval between the means of the Gaussian components are displayed in separate rows.

In the upper row of Figure 5, we observe that when $\alpha = 1$, i.e. using the KL divergence, the fitting becomes mode-covering. When α is reduced to 0.6, it successfully achieves mode-seeking fitting. In the middle row, where the interval between the means of the components is larger, making mode-seeking fitting more feasible, mode-seeking fitting is observed at $\alpha = 0.8$. In the bottom row, where the interval is even larger, mode-seeking fitting occurs even when minimizing the KL divergence, although the fitting targets the Gaussian with the larger variance on the left. As α decreases, the fitting shifts to the Gaussian on the right, which has smaller variance and higher probability.

In Figure 6, the values of $D_\alpha(\pi||\pi_{\text{ref}})$ are represented using color as the location and scale parameters of the Gaussian distribution $\hat{\pi}$ are varied. As α decreases, the $D_\alpha(\pi||\pi_{\text{ref}})$ values for mode-seeking $\hat{\pi}$ become smaller compared to those for mode-covering $\hat{\pi}$.

Similar results are observed in Figure 7 and 8 for cases with 3 and 4 Gaussian components. When minimizing the KL divergence, the fitting tends to be mode-covering or targets the component with larger variance. However, reducing α results in the fitting successfully targeting the region with the highest probability in all cases. As α decreases further, the variance of $\hat{\pi}$ also becomes smaller.

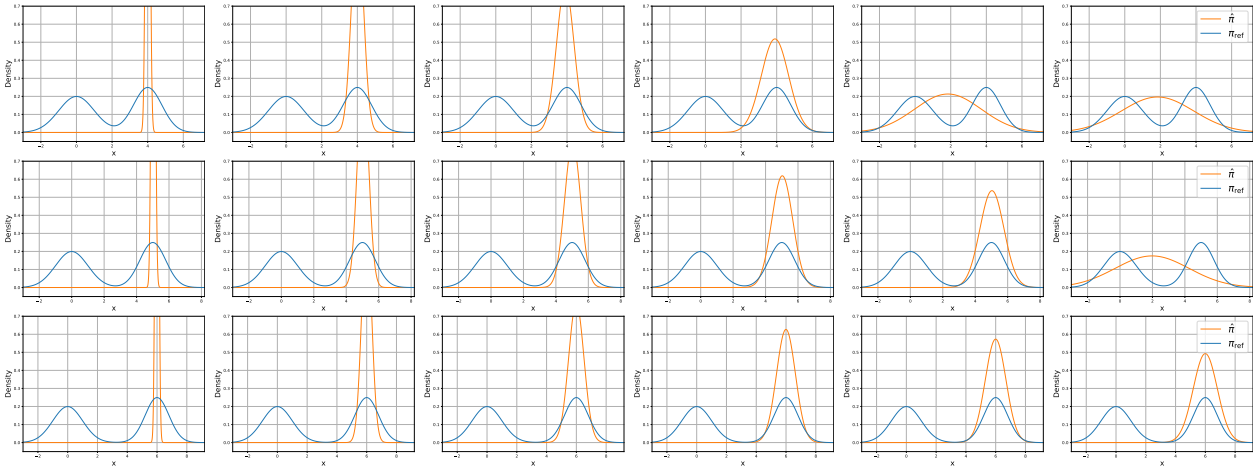
A.2 Comparison with β tuning

From Table 1, we observed that performance improves by decreasing the value of α. This raised the possibility that similar improvements might be achievable by tuning the β parameter in standard DPO. Therefore, we compared the performance of H-DPO, which showed promising results with parameters $(\alpha = 0.9, \beta = 0.01)$, against a DPO where β was similarly reduced $(\alpha = 1, \beta = 0.009)$. The results are presented in Table 3, showing that tuning β in DPO does not achieve the same level of improvement as H-DPO. The accuracy decreased in many tasks. The evaluation of coverage on the GSM8K dataset is shown in Figure 9, which indicates that tuning β in DPO does not improve coverage either. These results suggest that the performance enhancement obtained by tuning α to adjust the entropy cannot be replicated through β adjustment in DPO, thus demonstrating the effectiveness of H-DPO.

A.3 Other Details

A.4 Evaluation of Diversity

For the evaluation of diversity in Table 2, we used entropy, Self-BLEU (Zhu et al., 2018), and Distinct-1, -2 (Li et al., 2016). Regarding the measurement of entropy, we used 200 prompts from the UltraFeedback



$\alpha = 0$ $\alpha = 0.2$ $\alpha = 0.4$ $\alpha = 0.6$ $\alpha = 0.8$ $\alpha = 1$

Figure 5: π_{ref} is a GMM composed of two normal distributions, and $\hat{\pi}$ represents the normal distribution that minimizes $D_\alpha(\pi||\pi_{\text{ref}})$. The upper, middle, and bottom rows correspond to cases where the mean intervals between components are 4, 5, and 6, respectively. The standard deviations of each component are 1 and 0.8 from left to right.

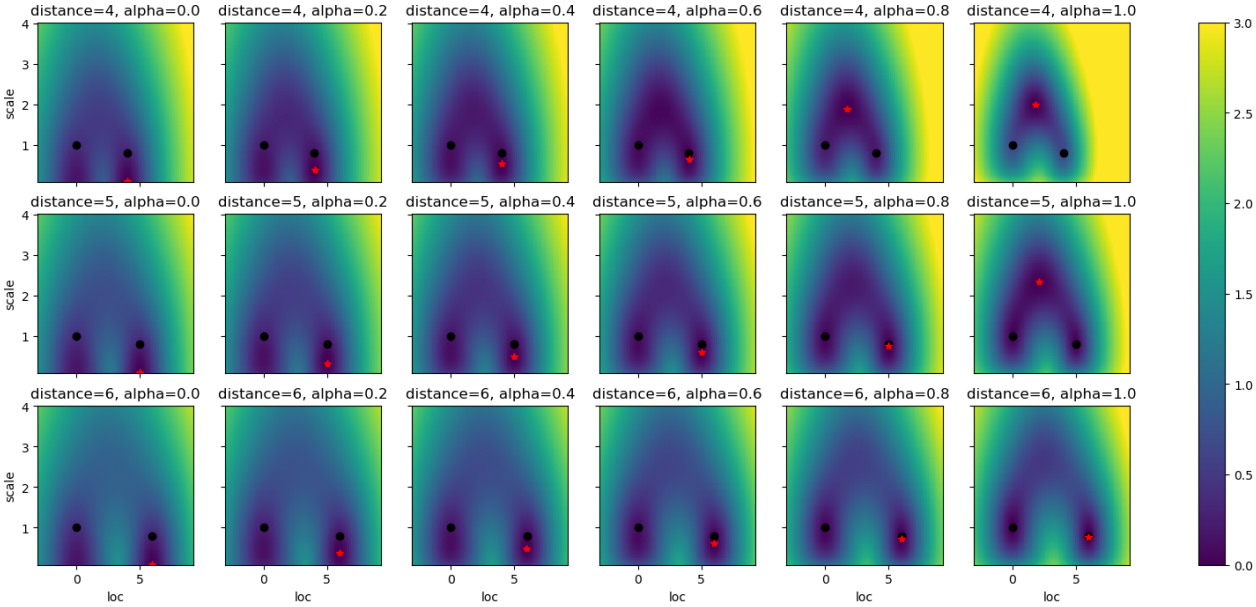
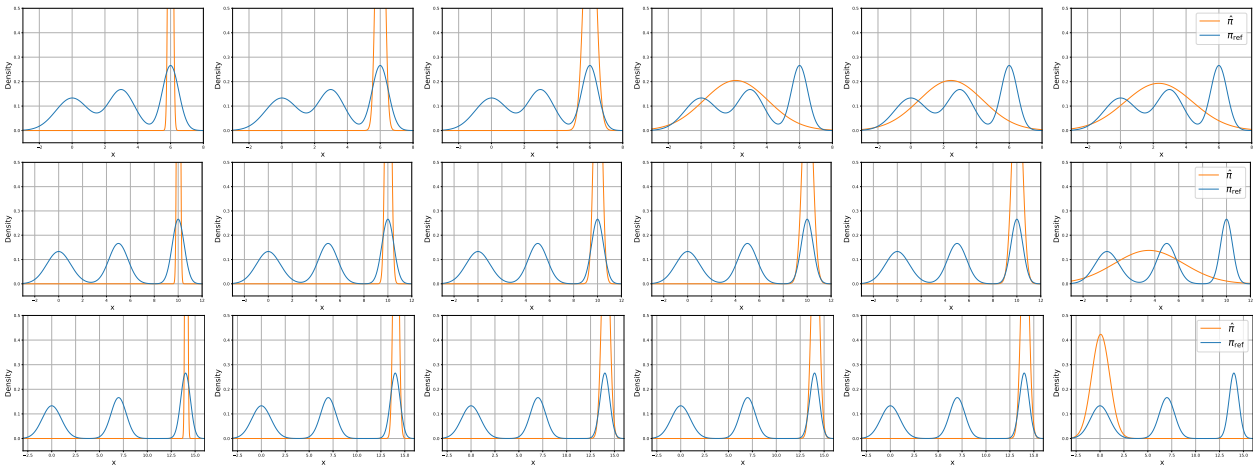
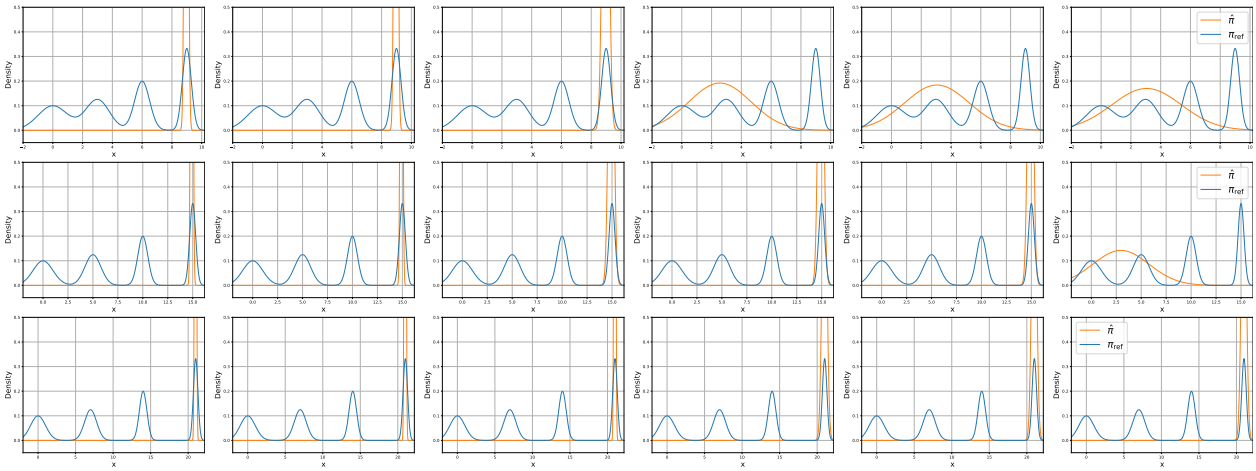


Figure 6: Values of $D_\alpha(\pi||\pi_{\text{ref}})$ for the normal distribution $\hat{\pi}$ with various location and scale parameters in the experiment shown in Figure 5. The red star indicates the parameters of $\hat{\pi}$ that minimize $D_\alpha(\pi||\pi_{\text{ref}})$, and these values are used to plot Figure 5.



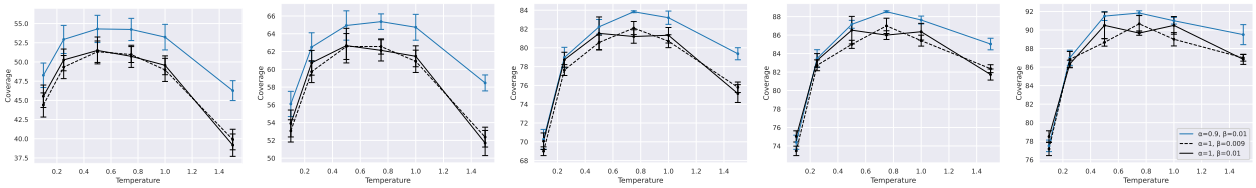
$\alpha = 0$ $\alpha = 0.2$ $\alpha = 0.4$ $\alpha = 0.6$ $\alpha = 0.8$ $\alpha = 1$

Figure 7: π_{ref} is a GMM composed of two normal distributions, and $\hat{\pi}$ represents the normal distribution that minimizes $D_\alpha(\pi||\pi_{\text{ref}})$. The upper, middle, and bottom rows correspond to cases where the mean intervals between components are 3, 5, and 7, respectively. The standard deviations of each component are 1, 0.8 and 0.5 from left to right.



$\alpha = 0$ $\alpha = 0.2$ $\alpha = 0.4$ $\alpha = 0.6$ $\alpha = 0.8$ $\alpha = 1$

Figure 8: π_{ref} is a GMM composed of two normal distributions, and $\hat{\pi}$ represents the normal distribution that minimizes $D_\alpha(\pi||\pi_{\text{ref}})$. The upper, middle, and bottom rows correspond to cases where the mean intervals between components are 3, 5, and 7, respectively. The standard deviations of each component are 1, 0.8, 0.5 and 0.3 from left to right.



pass@5 pass@10 pass@50 pass@100 pass@200

Figure 9: Coverage (pass@k) of H-DPO and DPO with various temperatures on **GSM8K**.

Table 3: Average scores of DPO and H-DPO on various tasks.

	GSM8K↑	HumanEval↑	MMLU-Pro↑	IFEval↑
DPO ($\alpha = 1, \beta = 0.01$)	26.40 ± 1.76	28.77 ± 0.45	31.83 ± 0.17	59.63 ± 0.72
DPO ($\alpha = 1, \beta = 0.009$)	25.13 ± 1.22	26.37 ± 1.34	31.93 ± 0.10	59.53 ± 0.35
H-DPO ($\alpha = 0.9, \beta = 0.01$)	**28.83** ± 2.32	**29.63** ± 0.45	**32.30** ± 0.17	**60.93** ± 0.50

(Cui et al., 2023) test dataset, which was used in the training of DPO, and generated 25 responses for each prompt. The maximum length of the responses was limited to 512, and the entropy was calculated using the log probability of each response, normalized by the response length. Self-BLEU and Distinct-1, -2 were also calculated using the same responses based on Zhu et al. (2018) and Li et al. (2016).

MMLU-Pro was evaluated using the official implementation from Wang et al. (2024d). IFEval and GSM8K were implemented using Gao et al. (2024), where IFEval was evaluated in a 0-shot setting, and GSM8K was evaluated in an 8-shot setting. HumanEval was evaluated using the official implementation from Chen et al. (2021). MMLU-Pro and IFEval were evaluated using one sampling for all test data, and the average accuracy and standard error at a temperature of 0 are shown in Table 1. For GSM8K, 200 test data were used, and for HumanEval, all test data were used, generating 200 responses for each prompt to calculate pass@k based on Chen et al. (2021), as shown in Figure 3 and 4. The results for pass@1 at a temperature of 0.1 are shown in Table 1. The results for varying temperatures in MMLU-Pro, GSM8K, HumanEval, and IFEval are shown in Figure 2 to 4 and 10.

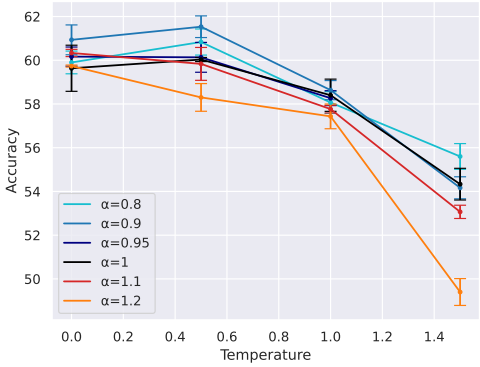


Figure 10: Accuracy on IFEval with various temperatures.