Data-efficient Online Training for Direct Alignment in LLMs

Chi Zhang
Beijing Institute of Technology
Beijing, China
zc315@bit.edu.cn

Jiacheng Wang
Beijing Institute of Technology
Beijing, China
wangjc@bit.edu.cn

Kun He*
Renmin University of China
Beijing, China
hekun2023@ruc.edu.cn

Chengliang Chai*
Beijing Institute of Technology
Beijing, China
ccl@bit.edu.cn

Yunpeng Zhang
Beijing Institute of Technology
Beijing, China
zyp123@bit.edu.cn

Yuping Wang
Beijing Institute of Technology
Beijing, China
wyp_cs@bit.edu.cn

Xu Zhou
Hunan University
Hunan, China
zhxu@hnu.edu.cn

Linan Zheng
University of Arizona
Tucson, AZ, USA
linanzheng@arizona.edu

Lijun Wu
Shanghai Artificial Intelligence Laboratory
Shanghai, China
wulijun3@pjlab.org.cn

Conghui He
Shanghai Artificial Intelligence Laboratory
Shanghai, China
heconghui@pjlab.org.cn

Lei Cao
Massachusetts Institute of Technology
Cambridge, MA, USA
lcao@csail.mit.edu

ABSTRACT

In recent years, online Direct Alignment from Preferences (DAP) has emerged as a popular alternative for Reinforcement Learning from Human Feedback (RLHF) due to its training stability and simplicity. In online DAP, training relies on preference data, each composed of a question and a pair of large language model (LLM) responses. However, annotating preference data, i.e., generating responses for questions, and using these data to train the RLHF model are computationally expensive. To address this, we propose DOTA, a data selection framework that minimizes the cost of generating preference data, while still ensuring the quality of training. First, we propose a theoretically grounded metric called Preference Perplexity (PFP) that enables us to design a low cost, gradient-based method to effectively estimate the contribution of each preference data point to model performance – critical to data selection. Second, rather than first generating responses for all candidate questions and then selecting preference data points by measuring their PFP, we design an iterative end-to-end framework that only has to generate responses for a small subset of questions, without missing valuable data points. Experiments on UltraChat-200k and HH-RLHF across 13 downstream tasks demonstrate that DOTA reduces computation cost by a factor of three on LLaMA-3-8B, Qwen-3-4B, and Qwen-3-1.7B, without compromising training effectiveness.

PVLDB Reference Format:
Chi Zhang, Jiacheng Wang, Kun He, Chengliang Chai, Yunpeng Zhang, Yuping Wang, Xu Zhou, Linan Zheng, Lijun Wu, Conghui He, and Lei Cao. Data-efficient Online Training for Direct Alignment in LLMs. PVLDB, 19(10): 2741-2754, 2026.

*Kun He and Chengliang Chai are the corresponding authors.

doi:10.14778/3828612.3828628

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/ZhangChi315/DOTA.

1 INTRODUCTION

A key challenge in developing reliable and effective large language models lies in aligning their behavior with human preference through Reinforcement Learning from Human Feedback (RLHF) [2, 30, 38]. The core of RLHF lies in the preference data, as the quality and diversity of these preference data points directly determine the accuracy of the reward signal and effectively establish the upper bound of the alignment performance [41, 48]. To be specific, a preference data point is in the form of $\{x, y^+, y^-\}$, where x denotes the input question, y^+ represents the chosen LLM response, and y^- represents the rejected response. An annotator is responsible for producing preference data [54, 57]. Most recent research has shown that online Direct Alignment from Preferences (DAP) [20, 21, 42] has emerged as a superior paradigm over offline methods. Instead of relying on a fixed set of preference data points pre-collected offline, online DAP leverages a target model to generate candidate responses for a given input x and then form fresh pairs of (y^+, y^-) during the *training process*. This online approach

Proceedings of the VLDB Endowment, Vol. 19, No. 10 ISSN 2150-8097.
doi:10.14778/3828612.3828628

ensures that the distribution of training data remains consistent with the evolving state of the target model [21]. Consequently, this on-policy generation effectively mitigates the distribution shift between the generation and training stages, allowing the model to learn through its own exploration and thus self-correcting more efficiently.

Motivation. Despite its advantages, online DAP incurs substantial computational overhead [44, 54, 62], primarily introduced by the following two main stages. (1) Preference data generation: it leverages the target model to generate multiple candidate responses y for each question x in a large candidate question pool $\mathcal{X}$, and then applies the reward model to identify the chosen (y^+) and rejected (y^-) responses for *all candidate questions*. Consequently, this generation process invokes multiple inference passes on each single data point, resulting in substantial computational overhead. (2) Training: The generated preference data is then utilized to optimize the target model, a process that entails significant computational overhead due to the massive volume of preference data points.

Intuitively, if there were a data efficient approach, which selectively generates training data (preference data) w.r.t. only a subset of candidate questions guaranteed to improve the performance of the model, it would greatly reduce the preference data generation cost and training cost of online DAP, while not sacrificing model performance.

Existing Solutions. Recently, `Less-is-More` [15] and SeRA [29] improve data efficiency by reducing the number of preference data points fed to the training stage. However, these approaches do not reduce the cost of generating preference data, because they have to first generate preference data points $\{x, y^+, y^-\}$ w.r.t. all candidate questions x before selecting the valuable ones for training. Therefore, although these approaches successfully reduce the training cost, they still incur substantial preference data generation cost, as depicted in Figure 1. In addition, these methods, simply selecting data points with a large score gap (evaluated by the reward model) between y^+ and y^-, suffer from significant performance degradation due to overlooking the state of the target model: even if a data point exhibits a clear preference score gap, it offers little value if the model has already aligned with the target preference, which ultimately limits the model performance.

Challenges. Overall, selectively generating reference points has to answer two challenging questions: (1) how to quantify the value of each preference data point to model performance and (2) how to identify the high value questions x without generating the responses and constructing the preference data points $\{x, y^+, y^-\}$.

Our Proposal. To solve the above challenges, we propose DOTA that effectively and efficiently identifies a subset of questions whose corresponding preference data points potentially improve the performance of the target model. DOTA features two key ideas: a theoretically grounded metric to measure the value of a preference data point, called Preference Perplexity (PFP) and an end-to-end selection-before-generation framework that leverages PFP to select valuable questions for generation. PFP addresses a key limitation in existing methods: the score gap, i.e., the margin between y^+ and y^- in reward model, produced by a *reward model*, does not necessarily reflect a data point's potential benefit to the target model.

Rather than assigning value to a preference data point using a static reward model, PFP measures its value by the potential impact

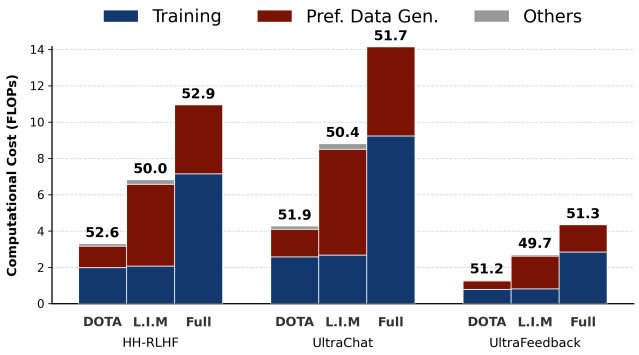

Figure 1: The computational cost of `DOTA`, `Less-Is-More` (`L.I.M`), and `Full`. Here, `Full` incorporates all preference data points generated from $\mathcal{X}$. Results are shown across HH-RLHF, UltraChat and UltraFeedback datasets. We break down the total cost into three phases: training, preference data generation, and others. The performance scores of each method are displayed at the top of their respective bars. Overall, `DOTA` saves many costs and achieves performance comparable to that of models trained with all preference data points generated from $\mathcal{X}$.

it can have on updating the current target model. Specifically, we estimate this value via the gradient magnitude induced by the data point: a larger gradient suggests that the target model has not yet fit the preference well, and such data points are therefore highly informative for refining the model's behavior. Directly computing gradient magnitudes, however, is computationally expensive, as it necessitates inference passes through both the current target and reference models. The reference model serves as an anchor during training, constraining updates so that the target model distribution does not drift excessively [44]. Fortunately, in the online setting, the divergence between the reference and target models can be tightly controlled. We can thus show theoretically that the reference-model term has a negligible effect on the gradient, allowing the gradient magnitude to be approximated using only a log-likelihood ratio under the target model. Guided by this insight, we introduce the PFP metric, which enables DOTA to estimate gradient magnitude with a single inference pass of the current target model, providing an effective and lightweight way to score the value of each preference data point.

Moreover, instead of selecting preference data points by first generating responses for all candidate questions and then computing their PFP, we propose an end-to-end selection-before-generation framework that only has to conduct this expensive process over a subset of carefully sampled questions. Specifically, DOTA starts with a lightweight warm-up step to align the feature space between questions x and generated preference data points $\{x, y^+, y^-\}$. Based on this alignment, DOTA clusters questions in $\mathcal{X}$ such that the preference data points generated within the same cluster exhibit similar deviation degrees. In this way, DOTA can effectively identify those high-quality clusters by first sampling a small subset of questions from each cluster and then generating preference data

points for these questions to save the costs, thereby transforming the expensive point-wise evaluation into an efficient cluster-wise estimation. While this cluster-level estimation significantly reduces computational overhead, a naive strategy of repeatedly sampling from a small number of high-PFP clusters tends to generate very similar preference data points. Such redundancy leads to diminishing returns during training. Therefore, we propose to balance the exploitation of high-PFP clusters with the exploration of the remaining clusters, by adopting an iterative strategy based on MAB. Specifically, we compute an upper confidence bound (UCB) score for each cluster, which down-weights high-PFP clusters when they have been sampled frequently.

Contributions. The key contributions of this work include:

- We propose DOTA, the first data selection framework that minimizes the cost of annotating reference data points for online DAP methods, while ensuring the quality of training.
- We propose a theoretically grounded Preference Perplexity (PFP) that effectively estimates the potential value of each preference data point with low cost.
- We design an end-to-end selection-before-generation framework that effectively leverages PFP to select a subset of valuable questions before data generation, so as to save the annotation cost. We also introduce a warm-up strategy to align the feature spaces of the questions and the generated preference data points.
- Extensive experiments on UltraChat-200k, UltraFeedback and HH-RLHF datasets as well as 13 popular downstream tasks demonstrate that DOTA significantly outperforms all baseline methods. Compared to training with a complete dataset, it saves 3× computation resources with comparable accuracy. Compared to the state-of-the-art data selection methods, it saves 2× computation resources, while improving accuracy by 2% (train/test on Llama-3-8B, Qwen-3-4B, and Qwen-3-1.7B models).

2 PRELIMINARY

Online DAP methods directly update the target model π_θ based on preference data points (i.e., $\{x, y^+, y^-\}$). The goal is to increase the probability of generating y^+, while decreasing that of y^-. This effectively aligns the target model π_θ with human preferences. Among all online DAP methods, online Direct Preference Optimization (DPO) [44] has emerged as a representative and widely used approach [21].

Online DPO. Adopting the Bradley–Terry model [3], online DPO models the probability that response y^+ is preferred to y^- (given a question x) as a logistic function of their score difference. Specifically, it defines the score of a response y under question x by the scaled log-likelihood ratio.

$$s_\theta(x, y) \;:=\; \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)},$$

where π_{ref} is a reference model (typically a copy of the target model which is frequently updated throughout the training process [38, 39, 45, 54]) that serves as an anchor by providing a baseline and by defining the KL penalty that discourages large deviations of π_θ from π_{ref} [44]. The resulting preference probability is

$$P_\theta(y^+ \succ y^- \mid x) \;=\; \sigma\big(s_\theta(x, y^+) - s_\theta(x, y^-)\big),$$

and the training loss is the negative log-likelihood of this preference:

$$\mathcal{L}_{\text{DPO}}(\{x, y^+, y^-\}) = -\log P_\theta(y^+ \succ y^- \mid x). \tag{1}$$

Here $\sigma(\cdot)$ denotes the sigmoid function. In practice, β controls the strength of the preference signal and the effective deviation from π_{ref}; it is often set to a small value (e.g., 0.1) for stable optimization [53]. During training, online DPO samples two responses $y_1, y_2 \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot \mid x)$ and uses a reward model $r_{\boldsymbol{\phi}}$ to rank them, yielding an online preference tuple $\{x, y^+, y^-\}$, which is then used to update the target model π_θ to reduce the deviation between π_θ and human preferences. For convenience, in the following we will use $\mathcal{L}_{\text{DPO}}(\{x, y^+, y^-\})$ to denote the loss function and omit the parameters π_θ, π_{ref} in $\mathcal{L}_{\text{DPO}}(\{x, y^+, y^-\}, \pi_\theta, \pi_{\text{ref}})$.

Other Online DAP Methods. Similar to online DPO, many other online DAP methods have also been proposed to directly align LLMs with human preferences, such as online Identity Policy Optimization (online IPO) [20] and online Sequence Likelihood Calibration with Human Feedback (online SLiC) [61].

Specifically, IPO replaces Bradley–Terry reward used in DPO with an objective that minimizes the squared log-probability gap between y^+ and y^-:

$$\mathcal{L}_{\text{IPO}}(\{x, y^+, y^-\}) = \left[\frac{1}{\beta}\left(s_\theta(x, y^+) - s_\theta(x, y^-)\right) - \frac{1}{2\beta}\right]^2. \tag{2}$$

SLiC directly maximizes the log-likelihood of y^+ while simultaneously minimizing that of y^- without the need for any separate reward function, and a clip margin averts over-confident shifts, so preference alignment is achieved with supervised learning:

$$\mathcal{L}_{\text{SLiC}}(\{x, y^+, y^-\}) = \max\left(0,\; 1 - \left(s_\theta(x, y^+) - s_\theta(x, y^-)\right)\right). \tag{3}$$

3 THE DOTA METHOD

3.1 Problem Definition

In this paper, we address the problem of selecting data from a large pool of candidate questions to generate preference data points used by online DAP as training data to align LLMs with human preferences. Specifically, given a candidate question pool $\mathcal{X}$, during each training iteration, we select a subset $\{x_i\}_{i=1}^n \subset \mathcal{X}$ to build a training set $\{x_i, y_i^+, y_i^-\}_{i=1}^n$ using the current target model π_θ. Then, in the next training iteration, we use this set to fine-tune the current target model with an online DAP method, where the objective is to minimize the loss of the updated model.

3.2 The Overall Framework of DOTA.

As shown in Figure 2, the DOTA framework can be divided into the following four steps. Firstly, DOTA incorporates a warm-up stage utilizing contrastive learning to align the feature spaces of question $x \in \mathcal{X}$ and generated preference data points $\{x, y^+, y^-\}$, ensuring that the subsequent clustering process is potentially based on the entire preference data points rather than purely the questions. Then, DOTA clusters all questions in $\mathcal{X}$, such that the preference data points $\{x, y^+, y^-\}$ generated within each cluster tend to exhibit similar deviation degrees (Step-1 in Figure 2), allowing us to estimate the potential value of a whole cluster jointly rather than evaluating every single data point individually. After that, in each training iteration, DOTA leverages the multi-armed bandit strategy

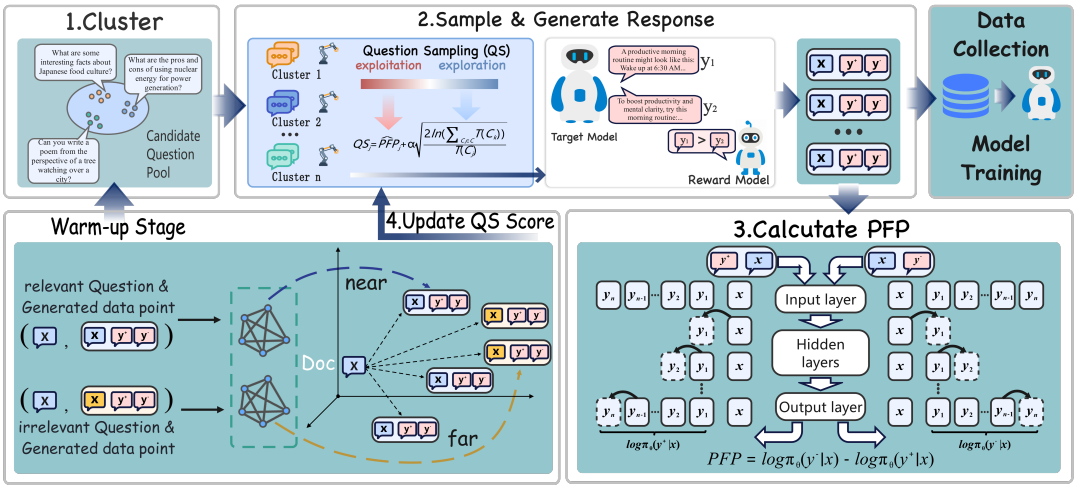

Figure 2: Overview of DOTA Framework

to select clusters and samples questions from these selected clusters to generate preference data points, which are subsequently added to the training data set (Step-2). Then it computes the PFP scores for these data points, which effectively represent their potential contribution to the target model by measuring the discrepancy between model outputs and human preferences (Step-3). Finally, taking into account the average PFP and the sampling frequency of each cluster, DOTA computes the upper confidence bound (UCB) and selects the next set of clusters to sample based on the UCB scores (Step-4). In this way, DOTA balances the exploitation of high-PFP clusters with the exploration of other rarely visited low-PFP clusters, thereby preventing overfitting and enhancing training efficiency.

The remainder of this section is organized as follows: We first introduce PFP in Sec 3.3, which is the key metric of our DOTA framework. We then detail the DOTA framework in Sec 3.4, followed by the warm-up stage in Sec 3.5.

3.3 Preference Perplexity (PFP)

The core challenge in online data selection lies in efficiently identifying the preference data points that maximize the model's learning progress. Theoretically, the magnitude of the gradient $\nabla_\theta \mathcal{L}_{\text{DAP}}$ serves as the ideal metric, as a large gradient indicates that the current model π_θ has not yet fitted the preference data point well, implying a high potential for optimization. Therefore, selecting data points with large DAP loss or gradients tends to better align the model with user preferences. However, computing the exact gradient or the raw DAP loss is computationally prohibitive in a high-throughput online setting. It requires two forward passes over both the target model π_θ and the reference model π_{ref} for every preference data point, doubling the inference cost. To address this bottleneck, we propose PFP and theoretically demonstrate that it serves as a reliable proxy for the DAP loss gradient while requiring *only a single forward pass* over the target model, drastically reducing the inference overhead.

For ease of exposition, we use online DPO as a representative example to illustrate PFP (derivations for other DAP methods are provided later in this section). By iteratively updating the reference model to the target model trained in the most recent iteration, online DPO effectively mitigates local performance bottlenecks and reward hacking, which are often associated with a static reference model [39, 45, 54]. Our proposed PFP further develops this idea: by initiating a new iteration at appropriate times, PFP precisely controls the divergence between the target model and the reference model. As a result, when assessing the influence of each data point on training, the effect induced by the target–reference mismatch becomes negligible, allowing us to focus primarily on the target model's behavior on that point.

In particular, to clarify how PFP captures the update of the target model, we compute the gradient of the DPO objective with respect to the model parameters θ, following [44]:

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\{x, y^+, y^-\}) = \\ -\beta\sigma\left(s_\theta(x, y^-) - s_\theta(x, y^+)\right) \cdot \nabla_\theta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_\theta(y^- \mid x)}, \tag{4}$$

where

$$s_\theta(x, y^-) - s_\theta(x, y^+) = \beta\left(\log \frac{\pi_\theta(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)} - \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)}\right). \tag{5}$$

A key observation behind PFP is that, in online DPO, the deviation of the current target model π_θ from the current reference model π_{ref} can be effectively controlled, ensuring that

$$s_\theta(x, y^-) - s_\theta(x, y^+) = O(\beta^c) \tag{6}$$

for some constant $c \in (0, 1)$ throughout training. Concretely, at the beginning of the i-th iteration, the current reference model is initialized to the current target model, i.e., $\pi_{\text{ref}^{(i)}} = \pi_{\theta^{(i)}}$. During the iteration, the DPO objective implicitly enforces a KL-type regularization that constrains the target model, thereby limiting its drift from π_{ref} within that iteration [8, 42, 46, 47]. Moreover, once the drift becomes too large—namely, when

$$\left|\log \frac{\pi_{\theta^{(i)}}(y^- \mid x)}{\pi_{\text{ref}^{(i)}}(y^- \mid x)} - \log \frac{\pi_{\theta^{(i)}}(y^+ \mid x)}{\pi_{\text{ref}^{(i)}}(y^+ \mid x)}\right| \geq \frac{1}{\beta^{1-c}}$$

for any group of $\{x, y^+, y^-\}$, we refresh the reference model by setting $\pi_{\text{ref}} = \pi_\theta$. We validate this strategy empirically in Section **??**. As a result, throughout online DPO training we maintain

$$\left|\log \frac{\pi_\theta(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)} - \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)}\right| < \frac{1}{\beta^{1-c}}.$$

Combined with Equation (5), we have Equation (6) immediately.

Consequently, by the Taylor expansion of the sigmoid function $\sigma(\cdot)$ around 0 and Equation (6), we obtain

$$\begin{aligned} &\sigma\left(s_\theta(x, y^-) - s_\theta(x, y^+)\right) \\ &= \frac{1}{2} + \frac{s_\theta(x, y^-) - s_\theta(x, y^+)}{4} + O\left(\beta^{3c}\right) = \frac{1}{2} + O\left(\beta^c\right). \end{aligned} \tag{7}$$

Substituting this approximation into Equation (4) yields

$$\begin{aligned} \nabla_\theta \mathcal{L}_{DPO}(\{x, y^+, y^-\}) &= -\left(\frac{\beta}{2} + O\left(\beta^{1+c}\right)\right) \nabla_\theta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_\theta(y^- \mid x)} \\ &= \frac{\beta}{2}\left(1 + O\left(\beta^c\right)\right) \nabla_\theta \log \frac{\pi_\theta(y^- \mid x)}{\pi_\theta(y^+ \mid x)}. \end{aligned} \tag{8}$$

Here, the prefactor $\frac{\beta}{2}\left(1+O\left(\beta^c\right)\right)$ can be treated as an approximately constant scalar, given that β is a near-zero hyperparameter which typically remains invariant across different preference data points.

From Equation 8, we observe that $\nabla_\theta \mathcal{L}_{\text{DPO}}(x, y^+, y^-)$ is determined by $\nabla_\theta \log \frac{\pi_\theta(y^-|x)}{\pi_\theta(y^+|x)}$. Crucially, this shows that the intensity of the model update scales directly with the model's uncertainty: the larger the discrepancy between the assigned probabilities of the chosen response (y^+) and rejected response (y^-), the stronger the resulting gradient. This implies a monotonic relationship: data points with a larger log-probability gap in the target model will yield larger gradients. Consequently, the preference data points that confuse the model the most are mathematically guaranteed to provide the strongest learning signals. Therefore, using $\log \frac{\pi_\theta(y^-|x)}{\pi_\theta(y^+|x)}$ to approximate the DPO loss is still effective for DOTA to select preference data points, because ignoring the constant $\frac{\beta}{2} + o(\beta)$ will not materially alter the relative order of these data points, which determines their priorities. Compared to computing the original DPO loss $\mathcal{L}_{\text{DPO}}$ in Equation 1, which requires an additional, expensive forward pass through the reference model π_{ref}, this approximation offers a significant efficiency boost. By eliminating the need to load and query the reference model, our approach transforms the data selection process from a computational bottleneck into a lightweight filtering step, making it especially suitable for the high-throughput acquisition of online preference data where latency and resource constraints are critical.

Building on this, we propose PFP as an effective metric for data selection:

$$\text{PFP}(\{x, y^+, y^-\}) = \log \pi_\theta(y^- \mid x) - \log \pi_\theta(y^+ \mid x). \tag{9}$$

A larger $\text{PFP}(\{x, y^+, y^-\})$ is likely to indicate a high $\log \pi_\theta(y^- \mid x)$ and a low $\log \pi_\theta(y^+ \mid x)$, suggesting that the model assigns higher probability in generating the rejected response y^- than the chosen one y^+. By prioritizing these data points where the target model struggles the most currently, we can greatly enhance the training efficiency.

In this way, to measure the contribution of a data point $\{x, y^+, y^-\}$, we only need to compute the probabilities of the target model predicting y^+ and y^- given x based on the next-token prediction (i.e., $\log \pi_\theta(y^+ \mid x)$ and $\log \pi_\theta(y^- \mid x)$). It is worth noting that these probabilities have already been computed during the responses generation process by the target model π_θ. Thus, PFP can be computed with zero additional computational cost, effectively bypassing the need for reference model inference (as shown in Figure 2).

Bridging PFP with Perplexity (PPL) [4]. To further illustrate why the PFP in Equation 9 is effective in measuring the contribution of a preference data point to the target model during RLHF stage, we rearrange it into Equation 11 and establish a connection with PPL (see Equation 10), which is broadly used in LLM instruction-tuning stage to measure the model confidence in next-token prediction when generating the response y of a data point (x, y):

$$\text{PPL}(y \mid x) = \exp\Big(-\frac{1}{N}\sum_{j=1}^{N} \log p(y_j \mid x, y_1, \ldots, y_{j-1})\Big). \tag{10}$$

$$\begin{aligned} \text{PFP}(\{x, y^+, y^-\}) &= \log \pi_\theta(y^- \mid x) - \log \pi_\theta(y^+ \mid x) \\ &= |y^+| \cdot \log \text{PPL}(x, y^+) - |y^-| \cdot \log \text{PPL}(x, y^-). \end{aligned} \tag{11}$$

In Equation 10 (PPL), N denotes the length of the response y and y_j represents the j-th token in the response y. In Equation 11 (PFP), $|y^+|$ and $|y^-|$ denote the lengths of responses y^+ and y^-, respectively. Specifically, PFP is formulated as the difference between $|y^+|\text{PPL}(x, y^+)$ and $|y^-|\text{PPL}(x, y^-)$. By accounting for the model's output probability for a given response y, both PPL and PFP serve as effective metrics for quantifying the model's proficiency in generating y. Nevertheless, by incorporating the sequence length, PFP shifts the focus of PPL to the overall gap between the model's output and the human preference over the whole sequence.

This transition from token-level accuracy to sequence-level alignment reflects a shift in the training objective. To be specific, during instruction tuning, next-token prediction is used to optimize the training loss to enhance instruction following capabilities, while length-normalized PPL serves as a key metric for token-level data evaluation. In contrast, the RLHF stage shifts the training focus toward aligning model behaviors with human preferences. Consequently, evaluation should also shift from a per-token average to a sequence-level preference assessment. By reversing the length normalization, our approach captures the total accumulated likelihood, ensuring that the preference signal is determined by the whole sequence rather than being weakened by token-level averaging. Therefore, these data points, if used in training, are valuable in aligning the target model with human preference, which is the goal of RLHF.

PFP for A Group of Data Points. Given a group of preference data points $\mathcal{G} = \{x_i, y_i^+, y_i^-\}_{i=1}^n$, we measure its average informativeness by the *group preference perplexity* $\mathcal{PFP}(\mathcal{G})$, defined as the mean of the individual PFP scores:

$$\mathcal{PFP}(\mathcal{G}) = \frac{1}{n}\sum_{\{x, y^+, y^-\} \in \mathcal{G}} \Big[\log \pi_\theta(y^- \mid x) - \log \pi_\theta(y^+ \mid x)\Big]. \tag{12}$$

A larger $\mathcal{PFP}(\mathcal{G})$ indicates that the target model π_θ finds the group more challenging: on average, it assigns higher probability to the rejected response y^- than to the chosen response y^+. Consequently, $\mathcal{PFP}(\mathcal{G})$ provides a direct proxy for the average magnitude of the loss gradient induced by the data points in this group.

PFP for Other Online DAP Methods. PFP equally works for other online DAP methods, such as online IPO and online SLiC. We provide the detailed proof below, which is similar to that of DPO. Our experiments in Sec. 4 also confirm the effectiveness of PFP in other DAP methods.

Identity Policy Optimization. Specifically, following the approach of online DPO, we first compute the derivative of the IPO loss.

$$\nabla_\theta \mathcal{L}_{\text{IPO}}(\{x, y^+, y^-\}) = -\frac{2}{\beta}\left(\frac{1}{2} + s_\theta(x, y^-) - s_\theta(x, y^+)\right)\cdot\nabla_\theta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_\theta(y^- \mid x)}. \tag{13}$$

As discussed above, in the online IPO setting, $|s_\theta(x, y^-) - s_\theta(x, y^+)|$ can be kept below $\beta^c/2$, and is therefore negligible compared with the leading constant $1/2$. Since $1/2$ dominates, the coefficient $\frac{1}{2} + s_\theta(x, y^-) - s_\theta(x, y^+)$ can be approximated as a near-constant. In this regime, the effective gradient magnitude is primarily governed by the target model margin term $\log \frac{\pi_\theta(y^-|x)}{\pi_\theta(y^+|x)}$. This implies that preference data points with larger values of $\log \frac{\pi_\theta(y^-|x)}{\pi_\theta(y^+|x)}$ induce stronger corrective updates, whereas preference data points with smaller values lead to vanishing gradients and exert only limited influence on training. Consequently, prioritizing preference data points with larger values of $\log \frac{\pi_\theta(y^-|x)}{\pi_\theta(y^+|x)}$ inherently aligns with a hard-sample prioritization strategy, thereby enhancing the effectiveness of gradient optimization during training.

Sequence Likelihood Calibration. Similarly, for the SLiC method, we also take the derivative of its loss:

$$\nabla_\theta \mathcal{L}_{\text{SLiC}}(\{x, y^+, y^-\}) = -\beta\,\mathbf{1}\left[s_\theta(x, y^+) - s_\theta(x, y^-) < 1\right]\cdot\nabla_\theta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_\theta(y^- \mid x)}. \tag{14}$$

In this formulation, $\mathbf{1}[\cdot]$ serves as a critical gating indicator function. Its core function is to implement a filtering mechanism: a preference data point is activated and contributes to the gradient update only when its calibrated margin $s_\theta(x, y^+) - s_\theta(x, y^-)$ falls below a specific threshold (here set to 1). In online SLiC, the reference model is periodically updated to the latest target model at the beginning of each iteration. Accordingly, the condition $s_\theta(x, y^+) - s_\theta(x, y^-) < 1$ can be enforced by triggering a new iteration whenever the margin approaches or exceeds this threshold. In this case, the magnitude of the gradient is primarily governed by the term $\log \frac{\pi_\theta(y^-|x)}{\pi_\theta(y^+|x)}$. In practice, our PFP selects preference data points with larger values for this term, effectively isolating preference data points which the model struggles to distinguish between positive and negative responses (i.e., the probability of the positive response is significantly lower than that of the negative response). By concentrating computational resources on the data points that are currently the most difficult to differentiate, our approach significantly enhances the efficiency of the model's parameter updates.

3.4 The DOTA Algorithm

In this section, we illustrate the details of the DOTA framework.

Multi-Armed Bandit (MAB) [50] serves as a classic decision-making framework that effectively characterizes the "exploration & exploitation" trade-off. In each round, an agent selects one of N candidate arms and receives a stochastic reward. The objective is to learn a selection strategy through interaction—under unknown reward distributions—to maximize accumulative rewards. The fundamental challenge in MAB is making decisions under uncertainty, which requires the agent to balance the benefit of reducing uncertainty by sampling new arms against the goal of maximizing expected reward by selecting the currently best arm.

Bridging MAB and Online DAP. Balancing the diversity and quality of training data presents a significant challenge during online DAP training. Prior approaches [15, 29, 40] typically focus exclusively on the quality of individual data points; this narrow focus often causes the model to overfit to a limited subset of data points, thereby degrading generalization. We identify a natural alignment between the exploration-exploitation trade-off in MAB frameworks and the diversity-quality equilibrium required in online DAP. By mapping the clusters of candidate questions to 'arms' and the PFP score to 'rewards', we transform the data generation problem of online DAP into a sequential decision-making process. Specifically, the exploration mechanism in MAB ensures data diversity, while the exploitation stage guarantees data quality.

Warm-up before Clustering. To ensure that the preference data points $\{x, y^+, y^-\}$ generated by the questions x within the same

Algorithm 1: DOTA Algorithm

Input: Candidate questions pool $\mathcal{X}$, sampled data ratio γ, number of training iterations M, target model π_θ, embedding model ϕ.
Output: Trained model $\pi_{\theta^{(M)}}$.

Sample a subset $\mathcal{X}_{warm} \subset \mathcal{X}$ randomly;
Generate warm-up data $\mathcal{D}_{warm} = \{(x, y^+, y^-) \mid x \in \mathcal{X}_{warm}\}$ using π_θ;
$\phi' \leftarrow$ Fine-tuned embedding model ϕ on $\mathcal{D}_{warm}$;
$\mathcal{C} \leftarrow$ Cluster($\mathcal{X}$) using aligned features from ϕ';
for C_i *in* $\mathcal{C}$ **do**
  $\mathcal{G}_i \leftarrow \emptyset$, $T(C_i) \leftarrow 0$;
end
for $m = 0, ..., M-1$ **do**
  Generated preference data $\mathcal{G} \leftarrow \emptyset$;
  while $|\mathcal{G}| < \gamma\,|\mathcal{X}|$ **do**
    Select cluster C_i with the highest QS score;
    Sample questions Q_i from C_i;
    Generate preference data points g_i from Q_i;
    $\mathcal{G} \leftarrow \mathcal{G} \cup g_i$, $\mathcal{G}_i \leftarrow \mathcal{G}_i \cup g_i$;
    $\widehat{\text{PFP}}_i \leftarrow \mathcal{PFP}(\mathcal{G}_i)$, $T(C_i) \leftarrow T(C_i) + 1$;
    for C_j *in* $\mathcal{C}$ **do**
      $QS_j \leftarrow \widehat{\text{PFP}}_j + \alpha\sqrt{\frac{2\ln \sum_{C_k \in \mathcal{C}} T(C_k)}{T(C_j)+1}}$;
    end
  end
  $\pi_{\theta^{(m+1)}} \leftarrow$ Train model $\pi_{\theta^{(m)}}$ with $\mathcal{G}$;
  $\pi_{\text{ref}^{(m+1)}} \leftarrow \pi_{\theta^{(m+1)}}$;
end
return $\pi_{\theta^{(M)}}$;

cluster are similar, we introduce a warm-up stage to fine-tune the embedding model ϕ before clustering, which is detailed in Sec 3.5.

QS Scoring with Upper Confidence Bound (UCB). As a classic strategy in the MAB problem, UCB [1] is broadly adopted to balance 'exploration' and 'exploitation' . Specifically, clusters with a higher $\widehat{\mathsf{PFP}}_i$ (see Section 3.3) have a higher chance to be selected, as it is more challenging for the model to correctly predict $\{x, y^+\}$ and $\{x, y^-\}$ associated with the question x. By prioritizing these challenging data points, DOTA directs the model to concentrate on resolving its most significant prediction errors, thereby ensuring substantial performance gains during training. At the same time, clusters that are less frequently visited are prioritized as well, promoting the exploration of a more diverse set of questions. Intuitively, the $\sqrt{\frac{2\ln\sum_{C_k\in C} T(C_k)}{T(C_j)+1}}$ term also serves as an uncertainty bonus: neglected clusters retain high uncertainty, prompting the algorithm to revisit them to verify if they contain overlooked high-value training signals. Based on the above insight, we define the Question Sampling (QS) score QS_j of a cluster C_j to effectively balance exploration (*i.e.*, data diversity) and exploitation (*i.e.*, data quality) as follows:

$$QS_j = \widehat{\mathsf{PFP}}_j + \alpha\sqrt{\frac{2\ln\sum_{C_k\in C} T(C_k)}{T(C_j)+1}}, \tag{15}$$

where $T(C_j)$ denotes the frequency of questions sampled from cluster C_j, $\sum_{C_k\in C} T(C_k)$ denotes the total number of samples from all clusters. The parameter α is set as $\frac{|C|}{1+\sum_{C_k\in C} T(C_k)}$ [23], which assigns a higher weight to exploration in early stages and exploitation in later stages (lines 16-18 in Algorithm 1). This adaptive strategy of α facilitates the data generation process from broadly covering most clusters to precisely focusing on specific clusters, preventing the target model from stabilizing too early on suboptimal solutions.

Update the QS Score. In each training iteration, the MAB strategy is employed to select clusters across multiple rounds based on the QS score. For each round, a subset of questions Q_i is sampled from the selected cluster C_i with the highest QS score (line 12); a set of preference data points g_i (i.e., $\{x, y^+, y^-\}$) is then generated from questions in Q_i (i.e., x). Then, the PFP score and sampling frequency $T(\cdot)$ of C_i will be updated as follows:

$$\widehat{\mathsf{PFP}}_i = \mathcal{PFP}(\mathcal{G}_i), \quad T(C_i) = T(C_i) + 1, \tag{16}$$

where $\mathcal{G}_i$ denotes all the preference data points generated from cluster C_i and $\mathcal{PFP}(\cdot)$ denotes the group preference perplexity defined in Equation 12. Then we update the QS score of all clusters. This dynamic update facilitates a strategic balance between exploration and exploitation in data generation, thereby enhancing data quality while simultaneously preserving the diversity.

Generated Data Collection. As shown in Figure 2, in each selection round, we add the generated preference data points g_i to the generated data pool $\mathcal{G}$ (Line 14). Then, in each training iteration m, we train model $\pi_{\theta^{(m)}}$ with the data points in $\mathcal{G}$ (line 20). This iterative process allows us to adaptively pinpoint preference data points where the target model deviates most from human preferences. By incorporating these data points, we significantly enhance the training efficacy. Concurrently, following the setting of online DAP [39, 45, 54], we employ the fine-tuned model $\pi_{\theta^{(m+1)}}$ as the new reference model $\pi_{\text{ref}^{(m+1)}}$ (line 21) at the end of each iteration. Finally, we apply the fine-tuned model $\pi_{\theta^{(m+1)}}$ and new reference model $\pi_{\text{ref}^{(m+1)}}$ to the next training iteration.

3.5 Warm-up Stage of DOTA

In this section, we detail the warm-up stage of DOTA.

Motivation. Given that the initial questions x are less informative than the final generated preference data points $\{x, y^+, y^-\}$, a discrepancy exists between the feature space of the original question embeddings and that of the generated preference data points. Ideally, we aim for similar preference data points to fall into the same cluster. However, clustering solely based on question feature embeddings may not reliably group similar preference data points, as semantic similarity between questions does not necessarily imply similarity in the final generated preference data points. For instance, two questions x_1, x_2 sharing semantic similarity might require different reasoning chains or domain knowledge to answer, which is only revealed in the generated answers $\{x_1, y_1^+, y_1^-\}$ and $\{x_2, y_2^+, y_2^-\}$. Without alignment, these questions might be erroneously clustered together. Ideally, we aim to cluster based on the full information of the final data points $\{x, y^+, y^-\}$; however, generating all preference data points $\{x, y^+, y^-\}$ for clustering is computationally expensive. Thus, we introduce a warm-up stage to align the two feature spaces via contrastive learning (bottom left corner of Figure 2), thereby enhancing overall clustering quality [27]. This alignment effectively constructs a shared semantic space where the lightweight question embeddings serve as reliable proxies for the computationally expensive full data points, bridging the gap between surface semantics and deep reasoning patterns.

Embedding Alignment. During the warm-up stage, we first randomly sample a small subset of the questions $\mathcal{X}_{warm}$ from the candidate question pool $\mathcal{X}$ (line 1 in Algorithm 1) and use the target model π_θ to generate preference data points D_{warm} (line 2). Then, we leverage $\mathcal{D}_{warm}$ to fine-tune the embedding model ϕ used for question embedding to capture the deep connection between the original questions and the generated preference data points (line 3). Specifically, we treat the sampled question x_i and the preference data point $\{x_i, y_i^+, y_i^-\}$ generated from this question as positive training example (i.e., $(x_i, (x_i, y_i^+, y_i^-))$) for contrastive learning. To generate negative examples $\{x_j, y_j^+, y_j^-\}$ for a question x_i, we conduct negative sampling from all current preference data points. Then we use the following loss function to fine-tune the embedding model ϕ:

$$L(\{x_i, y_i^+, y_i^-\}) = -\log\frac{e^{sim(x_i,(x_i,y_i^+,y_i^-))}}{e^{sim(x_i,(x_i,y_i^+,y_i^-))} + \sum_{j\neq i} e^{sim(x_i,(x_j,y_j^+,y_j^-))}},$$

where $sim(\cdot)$ denotes the cosine similarity between the embedding of a question x_i and the training data point (x_i, y_i^+, y_i^-). Intuitively, minimizing this loss injects the rich semantic information from the generated answers back into the question encoder, forcing it to look beyond lexical matches. In this way, questions that are potential to generate similar preference data points tend to be closer in the embedding space and are thereby grouped together in the same cluster.

4 EXPERIMENT

In this section, we fine-tune three base models on three benchmark datasets and conduct sufficient ablation studies to demonstrate the efficiency and effectiveness of DOTA. In particular, we focus on answering the following questions:

Q1: How does DOTA compare with other baselines in accuracy and efficiency? (§ 4.2)
Q2: How does DOTA perform on different LLMs and datasets? (§ 4.2)
Q3: How efficient is the DOTA framework? (§ 4.3)
Q4: How does the warmup period affect DOTA performance? (§ 4.4)
Q5: How effective is the PFP metric in DOTA? (§ 4.5)

4.1 Experiment Setup

Training Settings. In both our warm-up SFT training stage and online DAP training stage, the batch size is set to 128. The learning rate is set to 5e-6 for both the Qwen-3 model and the Llama-3 model. For all DAP (i.e., DPO, IPO and SLiC) training, the loss parameter β is set to 0.1. We adopt the Adam optimizer with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 10^{-8}$. Following the typical setting [58], the target model is trained for three iterations in DOTA. In each iteration, the model is trained on the generated data points using eight A800 GPUs. During the warm-up stage, we randomly sample 3% of data from the candidate question pool to construct preference data points for fine-tuning the embedding model ϕ. For the clustering, we employ the fine-tuned BAAI/bge-large-en-v1.5 [55] model to generate embeddings for candidate questions. Then, all the questions in $\mathcal{X}$ are clustered using the k-means algorithm, with the number of clusters determined automatically (see Section 4.6). As for the reward model, we employ Skywork-Reward-V2-Llama-3.1-8B [33], which holds the top position on the RewardBench2 [36] leaderboard[1]. The setting is consistent with that of Less-is-More [15].

Dataset Preparation. We use the popular alignment datasets UltraChat-200k [49], UltraFeedback [13], and HH-RLHF [2] as our candidate datasets. Specifically, UltraChat-200k is a well-constructed and high-quality subset selected from UltraChat conversations. HH-RLHF contains human preference data collected by Anthropic, comprising two parts: helpfulness and harmlessness. UltraFeedback is annotated by GPT-4, comprising diverse instructions paired with fine-grained ratings and critiques.

Baselines. We compare our DOTA(MAB) against 6 data selection baselines, which are all initialized with the SFT model and conduct the further online DAP training: (1) SFT. We use LLaMA-3-8B, Qwen-3-1.7B and Qwen-3-4B as our SFT models, which do not conduct the later online DAP training process for this baseline. (2) Random. At each iteration, 30% of the questions are randomly sampled from the candidate question pool $\mathcal{X}$, and the generated preference data points are then used to train the target model. (3) Full. At each iteration, we train the target model using all preference data points generated from the question pool $\mathcal{X}$. (4) SeRA [29]. At each iteration, SeRA leverages implicit reward margins (IRM) to select the top 30% of preference data points from the candidate question pool $\mathcal{X}$ for training the target model. (5) Less-is-More [15] utilizes a dual-margin strategy that combines external and implicit reward scores to select the top 30% of generated online preference data points for model training. (6) Curry [40] sorts the generated preference data points in ascending order of the reward-score gap (as evaluated by a reward model), enabling the target model to learn progressively from easier to more difficult examples. (7) DOTA(Top$k$). At each iteration, the target model generates preference data points for all questions x in the candidate dataset. Then the top-30% online preference data points with the highest PFP scores are selected and used to train the target model. (8) DOTA(MAB) uses PFP scores as rewards for the MAB to select 30% of the questions for data generation in each iteration, which are then used to train the target model.

Evaluation Metrics. In the field of RLHF training [18], the effectiveness evaluation is always conducted over the two types of benchmarks:

(1) Objective Benchmarks. We evaluate the fine-tuned LLMs on well-known benchmarks with objective answers (e.g., multiple-choice and fill-in-the-blank questions), which can be divided into 3 major types: i). General Tasks: MMLU [24], MMLU-PRO [52], DROP [17], AGIEval [63], KorBench [35]; ii). Mathematical Tasks: GSM8K [11], MATH [25]; iii). Coding Tasks: HumanEval [10], LCB [12]. We conduct all these evaluations with the OpenCompass [12] framework.

(2) Open-ended Benchmarks. We also evaluate the LLMs under the open-ended benchmarks with detailed and subjective answers. For the datasets, we consider four popular test sets: AlpacaBench [19], Evol-Instruct [56], RewardBench [36] and UltraFeedback [13]. Details of this benchmark can be seen at [59].

For efficiency , we report the EFLOPs[2] consumed by each method to quantify their GPU cost.

4.2 Results

As shown in Table 1, DOTA outperforms all baseline methods on downstream tasks w.r.t. LLaMA-3-8B, Qwen-3-1.7B and Qwen-3-4B.

Overall Performance. As demonstrated in Table 1, our method DOTA(MAB) outperforms all baseline methods on all downstream tasks including both types of benchmarks. To be specific, on both tasks, DOTA(Top$k$) has an improvement of 3.5% and 4.5% respectively compared with Random on Llama-3-8B, confirming the effectiveness of PFP. DOTA outperforms Curry because it selects preference pairs where the chosen and rejected responses are farther apart only based on the reward model, without considering the impact on the target model. Although SeRA and Less-is-More consider the performance of the target model, they do not perform well because they select preference samples (x, y^+, y^-) with large differences in implicit reward (i.e., $y^+ \succ\succ y^-$). This is because implicit rewards fail to reflect whether the target model has already learned well on the given preference data points; consequently, training on these points results in negligible updates to model parameters. This also explains why DOTA outperforms Full in some cases; while the candidate set $\mathcal{X}$ contains questions that the model has already learned; using preference data derived from these questions can cause overfitting, which in turn leads to degraded performance. DOTA(Top$k$) and DOTA(MAB) achieve similar performance, while DOTA(MAB) is

[1] RewardBench2 is a leading benchmark for evaluating the performance of reward models, covering diverse task domains with high evaluation difficulty and strong correlation to downstream performance.

[2] FLOPs (Floating Point Operations) measure the total number of floating-point computations, serving as a standard metric for computational cost in LLMs. 1 EFLOPs corresponds to 10^{18} floating-point operations.

Table 1: Evaluation results of online DPO with 30% selection ratio. The highest scores are highlighted in bold and the second- or third-highest scores are underlined. We run each experiment three times and report the average score.

Models	General Tasks						Mathematical		Coding			Open-ended Tasks				
	EFLOPs	mmlu pro	drop	mmlu	agieval	korbench	gsm8k	math	humaneval	lcb	Average	Alpaca	Evol	Rrward	UltraFeed	Average
Llama3-8B																
SFT	-	31.84	62.80	54.17	34.17	28.56	40.78	14.04	44.10	19.72	36.69	-	-	-	-	-
Random	5.820	32.95	66.18	59.08	36.63	30.80	44.23	15.50	47.05	21.74	39.25	57.68	56.48	56.84	57.85	57.21
Curry	10.912	33.21	66.07	59.68	36.15	30.44	44.54	15.28	47.31	21.53	39.36	57.87	57.08	57.04	57.57	57.39
SeRA	10.886	33.43	66.58	58.95	36.94	28.28	45.43	16.20	48.17	22.28	39.58	58.64	58.88	56.73	59.04	58.32
Less-is-More	11.289	35.12	67.41	61.05	37.93	32.40	47.29	16.62	51.39	24.80	41.56	60.24	59.64	59.87	60.14	59.97
Full	19.800	35.78	68.84	61.47	38.52	32.43	48.17	17.42	52.90	**25.32**	42.43	62.40	61.46	60.85	**63.10**	**61.95**
DOTA(Top$k$)	10.796	36.22	**69.91**	**62.35**	**38.73**	**32.68**	49.20	17.50	**53.05**	25.04	42.74	62.24	**61.79**	60.53	62.51	61.77
DOTA(MAB)	5.970	**36.53**	69.60	62.11	38.45	32.64	**50.57**	**17.78**	52.63	24.83	**42.79**	**62.88**	61.29	**60.88**	62.47	61.88
Qwen3-4B																
SFT	-	44.75	79.58	61.47	38.25	49.84	28.98	19.56	58.51	17.31	44.28	-	-	-	-	-
Random	3.285	53.61	83.13	76.24	41.86	53.11	29.95	20.26	64.63	22.19	49.44	68.22	68.67	69.60	70.29	69.20
Curry	6.297	54.05	83.04	76.48	42.23	52.78	30.68	20.12	65.24	22.07	49.63	68.12	68.97	68.91	71.12	69.28
SeRA	6.428	54.61	83.21	76.83	42.67	52.80	30.33	20.42	64.54	24.33	49.97	69.12	68.54	69.34	71.83	69.71
Less-is-More	6.828	55.71	84.44	77.65	42.18	54.88	32.45	21.94	66.83	27.16	51.47	71.52	71.49	71.20	72.43	71.66
Full	10.950	56.28	**85.35**	78.02	**44.98**	55.04	32.84	23.02	69.80	**30.74**	**52.90**	**73.66**	72.24	70.40	74.43	72.68
DOTA(Top$k$)	6.467	**56.97**	85.18	**78.22**	44.69	55.28	**33.21**	22.92	**70.73**	28.14	52.82	72.55	**73.62**	**71.40**	74.33	72.98
DOTA(MAB)	3.415	56.53	84.96	77.93	44.55	**55.60**	32.95	**23.12**	68.68	29.10	52.60	72.73	**73.62**	71.00	**74.66**	**73.00**
Qwen3-1.7B																
SFT	-	35.65	68.19	60.75	35.35	38.10	30.46	24.96	57.93	17.31	40.97	-	-	-	-	-
Random	1.755	38.92	71.29	63.37	37.15	40.24	32.95	26.26	57.54	18.28	42.89	70.18	67.46	70.02	69.16	69.21
Curry	3.773	39.17	71.15	63.55	37.04	40.84	33.21	25.87	58.37	18.07	43.03	70.42	66.66	70.49	69.45	69.25
SeRA	3.881	39.46	71.85	63.63	37.65	40.40	33.34	26.96	58.39	18.84	43.39	71.44	68.46	71.70	70.16	70.44
Less-is-More	3.981	41.77	72.33	64.68	38.15	42.12	35.71	27.58	61.23	19.66	44.80	72.42	68.05	72.40	71.28	71.04
Full	6.312	41.98	**73.68**	**65.86**	**38.31**	**43.14**	36.73	28.16	**62.43**	**21.84**	**45.79**	**74.93**	**72.81**	74.00	**73.14**	**73.72**
DOTA(Top$k$)	3.516	**42.60**	73.39	65.80	38.29	43.04	36.77	**29.04**	62.20	20.86	45.78	74.68	71.24	**74.14**	72.95	73.25
DOTA(MAB)	1.875	42.47	72.84	65.37	38.24	42.64	**37.15**	28.28	61.98	20.52	45.57	74.04	70.91	73.75	73.02	72.93

more efficient (saving 4.7 EFLOPs on Llama3-8B) because it generates preference data only for questions sampled from the selected clusters, rather than processing the entire candidate pool $\mathcal{X}$.

In addition, we also report the performance improvement of all methods across three training iterations, as illustrated in Figure 3. On all three models, our proposed DOTA(MAB) achieves consistent and notable gains across the three iterations, further demonstrating its robustness and effectiveness. Furthermore, we also include UltraChat-200k [49] (UltraChat) and UltraFeedback [13] as the candidate question pool $\mathcal{X}$, from which 20%, 30% and 50% of the questions are selected to fine-tune Qwen-3-4B. As shown in Figure 4(a) and 4(b), by only selecting 30% questions, DOTA achieves competitive performance compared to using the entire dataset.

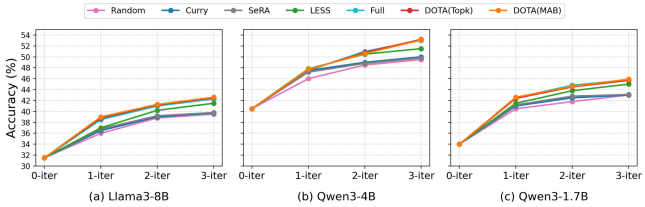

Figure 3: Performance of different models by varying the number of training iterations.

Applying DOTA to Other Online DAP Methods. To verify the general applicability of DOTA, we also conducted experiments on other widely used DAP methods such as IPO and SLiC. Specifically, we fine-tuned LLaMA-3-8B on 30% of the data selected by DOTA. As reported in Figure 7, DOTA consistently outperforms all baselines on downstream tasks. This demonstrates that DOTA is applicable to different online DAP methods that use different training strategies.

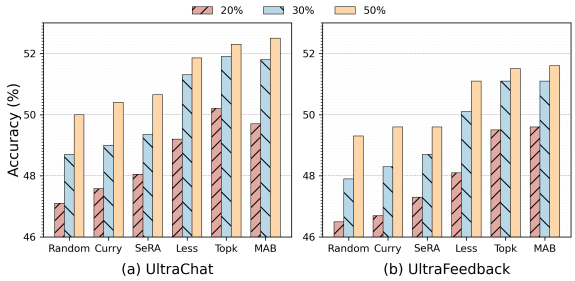

Figure 4: Evaluation of DOTA across two datasets.

4.3 Efficiency Analysis

To evaluate the practical feasibility of our approach, we compare the total computational costs (i.e., FLOPs) required to achieve the same accuracy across four stages: the warm-up stage, the question selection stage, the preference data points generation stage and the model training stage. As illustrated in Table 1, DOTA(MAB) achieves the lowest FLOPs consumption among all baselines, with the exception of simple heuristic-based methods like Random. Unlike Top-k, SeRA, and Less-is-More, which require extensive data synthesis, DOTA only generates preference pairs from a small subset of candidate questions. By eliminating the need to process the entire candidate pool $\mathcal{X}$, our approach substantially reduces data generation overhead while maintaining performance comparable to full-pool training, ultimately delivering a 3× reduction in total computational cost. Meanwhile, we also analyze the FLOPs breakdown for each step in DOTA and other methods. The results and detailed analysis can be seen at [59].

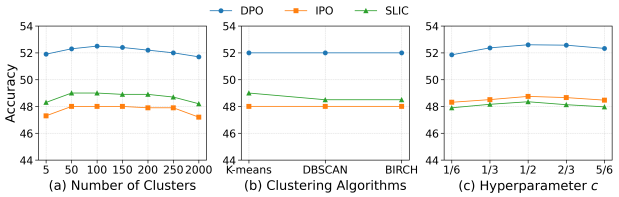

Figure 5: Ablation studies on clustering / hyperparameter c.

4.4 Robustness of the Warm-up Stage

Warm-up Data Ratio. To investigate the effectiveness of the warm-up stage in DOTA, we conduct experiments w.r.t. ϕ with different warm-up ratios on HH-RLHF and UltraChat-200k datasets. We vary the size of warm-up data ratio to be 0%, 3%, 10%, 25%. Specifically, as the warm-up ratio increases, the model's performance improves, while the overall training cost (i.e., FLOPs) also rises. Notably, with a warm-up ratio of 3%, comparable performance is achieved at a lower computational cost compared to using 25% of the data for warm-up. Therefore, we select the warm-up ratio of 3% as the default setting.

Table 2: Ablation Study on Warm-up Ratio.

Base Model	HH-RLHF				UltraChat-200k			
Warm-up Ratio	0%	3% (default)	10%	25%	0%	3% (default)	10%	25%
Llama-3.1-8B	60.38	61.88	61.97	62.11	53.30	55.87	56.09	56.28
Qwen-3-4B	71.76	73.00	73.17	73.31	60.15	61.28	61.64	61.71
Qwen-3-1.7B	70.19	72.93	73.02	73.16	58.65	60.15	60.67	60.79
Average Score	67.44	69.27	69.38	69.53	57.37	59.10	59.47	59.60

Number of Clusters and Clustering Algorithms. Figure 5(a) illustrates the performance of DOTA on HH-RLHF as the number of clusters varies from 5 to 2000. When the number of clusters is small (e.g., $k = 5$), DOTA performs poorly (0.5% lower accuracy for DPO) due to the high variance of preference data points generated from the documents in each cluster. In this case, the sampled questions do not represent the cluster well. Similarly, when the number is too high (e.g., $k = 2000$), there will be many clusters that generate similar preference data points. This undermines the diversity of the clusters that DOTA explores, leading to performance degradation – 0.7%, 0.8% and 0.7% lower accuracy. We can observe that DOTA keeps consistently good performance when k is within the range of 50 to 250. In our experiments, the Elbow method [26] recommends 125 as the appropriate number of clusters for the HH-RLHF dataset, which falls within this range and remains stable performance around this number. We also demonstrate the performance of DOTA when using other typical clustering algorithms including BIRCH [60] and DBSCAN [14] in Figure 5(b).

Effectiveness of the Warm-up Stage. We conduct following baseline comparisons to demonstrate the effectiveness of warm-up stage: (1) `No Adaptation` (`Off-the-shelf Embeddings`): we directly use an off-the-shelf embedding model (without fine-tuning) to compute embeddings for the questions in the candidate question pool and then perform clustering. (2) `Prompt Features`: we first select 3% of questions from the candidate question pool as the warm-up subset. Then, for each prompt x in the warm-up subset, we use the target model to generate a semantically equivalent paraphrase x', which is treated as a positive training example for contrastive learning. (3) `Length`: for each question x in the candidate pool, we generate sample response y using the target model and calculate the response length. (4) `Entropy`: we calculate the entropy $H(y|x)$ across the generated training data points to identify regions of the data pool where the target model lacks confidence. (5) `Reward-gap`: this strategy utilizes an external reward model to calculate the static quality margin $G_i = |r(y^+) - r(y^-)|$ for each preference pair.

Table 3: Comparison of warm-up alternatives.

	Llama-3-8B		Qwen-3-4B		Qwen-3-1.7B	
Method / Model	Cost	Acc	Cost	Acc	Cost	Acc
`No Adaptation`	5.82	39.25%	3.28	49.44%	1.76	42.89%
`Prompt Features`	5.95	39.67%	3.37	49.20%	1.89	43.73%
`Length`	7.76	40.38%	5.46	50.06%	2.67	43.89%
`Entropy`	8.16	41.19%	5.55	51.02%	2.76	44.47%
`Reward-gap`	10.37	40.56%	6.58	51.61%	3.59	43.72%
`DOTA(Ours)`	5.97	**42.79%**	3.41	**52.60%**	1.87	**45.57%**

In our experiment, we compare `MAB` (Ours) with all baselines in terms of training performance (`Acc`) and the computational cost of data selection and model training (`Cost`). As shown in Table 3, the warm-up method in DOTA outperforms all other methods by approximately 2% in final accuracy. To be specific, `Prompt Features` and `No Adaptation` perform clustering solely based on the semantic information of the prompts (questions). As a result, it cannot capture the full semantic information of the prompt and its generated responses. `Length` tends to group samples with similar output length but different semantic domains or alignment needs. Compared with our method, `Entropy` is weaker because it measures the model's self-confidence rather than the discrepancy between the preferred and rejected responses. `Reward-gap` underperforms DOTA because its data selection criterion is independent of the target model. Specifically, this method may repeatedly prioritize samples that are easy or already mastered by the target model but still exhibit clear reward separation, leading to redundant training data. Furthermore, except for `No Adaptation` and `Prompt Features`, which rely only on question-side information, all other methods incur substantially higher computational costs than the warm-up method used in DOTA. This is because they require generating complete answers for all questions in the candidate pool before clustering.

4.5 Evaluation of the Data Generation Stage

Effectiveness of MAB. We compare DOTA (MAB) against the following simpler baselines under the same compute budget: (1) `Stratified Sampling`: At each iteration, we sample the same number of questions from every cluster. (2) `Greedy Selection`: We first sample questions from all clusters, and then estimate the average PFP score of each cluster based on these training data points. We then continue selecting the cluster with the currently highest estimated average PFP until the budget is exhausted. (3) `Cluster-Top-k-Explore`: We sample questions from the top-k clusters with highest PFP scores while periodically performing random exploration over the remaining clusters.

Stability and Convergence Speed: In Figure 6, the x-axis denotes the percentage of sampled data, while the y-axis reports downstream accuracy. To be specific, the curve of MAB (Ours) rises more rapidly from 0% to 30% question sampling ratio and remains stable from 30% to 50% sampling ratio, indicating more efficient convergence and better stability during the selection process.

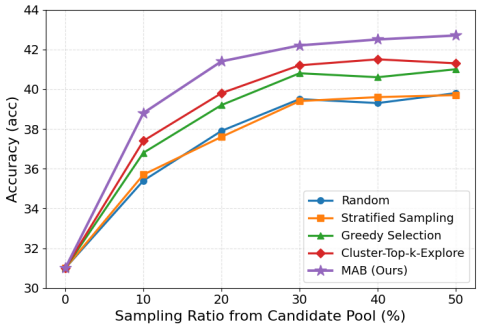

Figure 6: Convergence speed and stability of DOTA.

Final Performance: As shown in Table 4, MAB (Ours) achieves the highest accuracy among all methods on different model backbones. This is because it adaptively balances the exploitation of high-value clusters with the exploration of under-sampled clusters. In contrast, Stratified Sampling ensures uniform coverage but overlooks differences in data quality across clusters. Greedy Selection aggressively exploits the currently best cluster and is therefore reduce data diversity and degrade model generalization. Unlike Cluster-Top-k-Explore, which relies on a heuristic periodic schedule for random exploration, DOTA (MAB) adaptively updates the sampling probability of each cluster based on its historical sampling frequency. As a result, when the heuristic exploration frequency is set too high, Cluster-Top-k-Explore may waste computational resources sampling low-PFP data; when it is set too low, the selected data may be overly concentrated in the top-k clusters and reducing data diversity.

Effectiveness of Selection Metrics (PFP). To validate the effectiveness of the proposed PFP, we conduct an ablation study using two baselines: (1) *Full*, which serves as an upper bound, and (2) *DAP Loss*, which utilizes the exact DPO loss (requiring the reference model) as the oracle metric. Detailed results and explanation can be seen in [59].

Table 4: Comparison of different selection strategies.

Method/Model	Llama-3-8B	Qwen-3-4B	Qwen-3-1.7B
Random	39.25%	49.44%	42.89%
Stratified Sampling	39.65%	49.51%	43.27%
Greedy Selection	40.87%	50.73%	43.30%
Cluster-Top-k-Explore	41.05%	51.14%	44.23%
MAB(Ours)	**42.79%**	**52.60%**	**45.57%**

4.6 Ablation Study

Ablation of Refresh Frequency. In consideration that the typical training practice in online DAP is to refresh the reference model at the end of each epoch [39, 45, 54]. Under this training protocol, the divergence typically reaches its largest value near the end of each training period. Thus, in our experiments, we vary the frequency of refreshing the reference model: ① Per Epoch: refreshing the reference model at the end of each training epoch. ② DOTA (Ours): refreshing the reference model whenever the KL divergence exceeds the threshold c within each training epoch. To be specific, we conduct experiments on Llama-3-8B, Qwen-3-4B, and Qwen-3-1.7B, and evaluate model performance using accuracy on both objective and open-ended tasks, following the same experimental setting as in Section 4. As shown in Table 5, across all three target models, refreshing the reference model once per epoch achieves performance comparable to that of the DOTA refresh strategy. These results demonstrate that the PFP proxy remains effective even when the reference model is refreshed less frequently (resulting in a larger KL divergence), which indicates that the performance gains of PFP are not tied to a specific training regime.

Table 5: Stress test under different refresh frequency.

Update frequency	Per Epoch		DOTA (Ours)	
Model / Tasks	Objective	Open-ended	Objective	Open-ended
Llama-3-8B	42.71%	61.67%	42.79%	61.88%
Qwen-3-4B	52.73%	72.79%	52.60%	73.00%
Qwen-3-1.7B	45.76%	72.70%	45.57%	72.93%

Table 6: Robustness of DOTA with different reward models.

Method/Model	Llama-3-8B	Qwen-3-4B	Qwen-3-1.7B	Average
Random	56.28%	67.32%	67.03%	63.54%
Curry	56.87%	68.11%	67.38%	64.12%
SeRA	58.90%	69.55%	67.12%	65.19%
Less-is-More	58.47%	69.96%	68.23%	65.55%
DOTA (Top$k$)	59.56%	71.71%	71.21%	67.49%
DOTA (MAB)	**59.69%**	**71.76%**	**71.37%**	**67.61%**

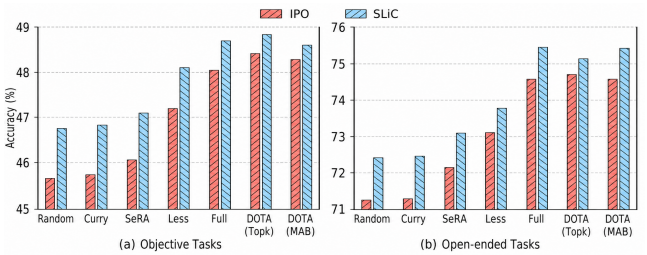

Figure 7: Evaluation results of online IPO and SLiC with 30% selection ratio on Llama-3-8B.

Ablation study of the reward model. To test the robustness of DOTA with the quality of reward signals, we replaced our default model with a weaker alternative (Starling-RM-7B-alpha),

which has a lower correlation with human preferences. We also use Llama-3-8B, Qwen-3-4B, and Qwen-3-1.7B as base models to compare the performance of DOTA with other baselines on open-ended tasks (same with Section 4). As shown in Table 6, DOTA consistently outperforms other methods on the accuracy downstream tasks. This confirms that the effectiveness of PFP is robust regardless of the capability of the reward model.

Data Selection Ratio. Figure 4 presents a comparison between DOTA and other baselines using various data generation ratios (i.e., 20%, 30% and 50%). Across all proportions, the performance of DOTA surpasses that of the other baseline methods. Interestingly, we find that for all models, generating only 30% of the data points in DOTA yields better results than using the full dataset $\mathcal{X}$. This demonstrates the effectiveness of DOTA. This is because not all data points generated from the questions in $\mathcal{X}$ are highly valuable for the model's training. The detailed results are shown as follows.

Ablation Study of c. In DOTA, we employ $1/\beta^{1-c}$ to dynamically regulate the update frequency of the reference model relative to the target model. As shown in Figure 5(b), the model achieves optimal performance when $c = 1/2$. A large c results in a high $1/\beta^{1-c}$ value, leading to lower update frequency of the reference model. This can cause performance bottleneck and reward hacking [39, 45, 54]. Conversely, a small c yields a low $1/\beta^{1-c}$ value, triggering excessively frequent updates to the reference model. This greatly compromises training stability and increases computational cost.

4.7 Effectiveness of LLM Evaluation

Metrics and Results between Multiple Judges. To mitigate the potential bias introduced by using a single judge model, we further conducted an additional evaluation with multiple judge models, including Gemini-3-Pro, Grok-4-Base, and GPT-5.1. Specifically, we use the open-ended tasks Alpaca, Evol, Reward and UltraFeed introduced in Section 4 to evaluate the effectiveness of DOTA and other baselines and report the average accuracy. As shown in Table 7, DOTA still outperforms other baselines under this new metric.

Table 7: Results of multiple LLM judges.

Method/Tasks	Alpaca	Evol	Reward	UltraFeed	Average
Random	56.17%	55.78%	56.96%	59.46%	57.09%
Curry	57.73%	56.91%	57.72%	56.32%	57.17%
SeRA	59.37%	59.76%	56.14%	59.01%	58.57%
Less-is-More	61.51%	59.64%	59.87%	60.90%	60.48%
Full	**62.78%**	61.59%	60.23%	**62.72%**	**61.83%**
DOTA(Top$k$)	62.77%	61.25%	**61.60%**	61.43%	61.76%
DOTA(MAB)	62.49%	**61.87%**	60.49%	62.27%	61.78%

Consistency between Human and LLM Judgments. We also conduct a small-scale human validation study on 300 randomly sampled response pairs from the open-ended benchmarks. As shown in Table **??**, the majority-vote human labels agree with the LLM judgments on 96.7%, 97.9%, 97.0%, and 97.3% of the 300 sampled pairs for all LLMs, which shows the reliability of LLM-based judgments (detailed results can be seen in [59]).

5 RELATED WORK

Data selection methods have drawn considerable attention from the database community, aiming to identify which candidate data points should be included for downstream tasks [7, 9, 22, 31, 34, 43, 51], which is driven by the fact that the quality of available data often varies significantly. Nowadays, with the development of LLMs, data selection for both the pretraining and instruction tuning stages of LLMs has been extensively explored, with research primarily focusing on improving data quality [32], enhancing diversity [28] and enhancing training efficiency [37]. However, these methods are inherently designed for the SFT and pretraining paradigms, making them difficult to directly adapt to RLHF, and data selection within the RLHF stage remains under-explored. Besides, among various data selection techniques, coreset selection [5, 6, 16, 32, 37] has emerged as a critical method for optimizing model training. Specifically, to mitigate the often prohibitive computational costs of large-scale training, coreset selection identifies a representative subset from the candidate pool, thereby facilitating efficiency through strategic data reduction.

6 CONCLUSION

In this work, we introduced DOTA, a novel data selection framework that addresses the critical computational bottleneck of data generation in online DAP. Central to our approach is PFP, a theoretically grounded metric that efficiently estimates the training value of data points by measuring their deviation from human preferences. To further eliminate the need for exhaustive generation, DOTA utilizes a warm-up stage to align question-response feature spaces, enabling an end-to-end selection-before generation strategy to dynamically select the most informative questions. Experiments on three datasets confirm that DOTA reduces computational costs by 3× while outperforming existing baselines, offering a scalable solution for efficient model alignment.

7 LIMITATIONS

Since the warm-up stage in DOTA is performed on only a subset of questions from the candidate pool, the learned alignment between the question feature space and the preference-data feature space is only an approximation of the true alignment over the entire candidate pool. Therefore, it may break down in cases involving highly out-of-distribution (OOD) questions, where the warm-up subset fails to capture the extreme variance of the entire pool.

Besides, our open-ended evaluation protocol also has several limitations due to the gap between LLM-based judgments and real human preferences. Specifically, if the judge model shares similar biases with the reward model, improvements in LLM-judge scores may partly reflect better alignment with the judge-model bias, rather than improvement in satisfying human preferences.

ACKNOWLEDGMENTS

Chengliang Chai is supported by the NSFC (U25B2019, 62472031, 62577010), the National Key Research and Development Program of China (2024YFC3308200), Beijing Nova Program, and Huawei. Yuping Wang is supported by the NSFC (U23A20297, U23A20317). Lei Cao is supported by the NSF (DBI-2327954) and Amazon Research Awards. Kun He supported by the National NSFC (62472430).

REFERENCES

[1] Peter Auer. 2000. Using upper confidence bounds for online learning. In *Proceedings 41st annual symposium on foundations of computer science*. IEEE, 270–279.

[2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2025. Training a helpful and harmless assistant with reinforcement learning from human feedback. *13th International Conference on Learning Representations (ICLR)* (2025).

[3] Ralph Allan Bradley and Milton E. Terry. [n.d.]. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika* ([n. d.]), 324. https://doi.org/10.2307/2334029

[4] Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18, 4 (1992), 467–480.

[5] Chengliang Chai, Jiabin Liu, Nan Tang, Ju Fan, Dongjing Miao, Jiayi Wang, Yuyu Luo, and Guoliang Li. 2023. Goodcore: Data-effective and data-efficient machine learning through coreset selection over incomplete data. *SIGMOD* 1, 2 (2023), 1–27.

[6] Chengliang Chai, Jiabin Liu, Nan Tang, Guoliang Li, and Yuyu Luo. 2022. Selective data acquisition in the wild for model charging. *Proceedings of the VLDB Endowment* 15, 7 (2022), 1466–1478.

[7] Chengliang Chai, Nan Tang, Ju Fan, and Yuyu Luo. 2023. Demystifying artificial intelligence for data preparation. In *SIGMOD*. 13–20.

[8] Changyu Chen, Zichen Liu, Chao Du, Tianyu Pang, Qian Liu, Arunesh Sinha, Pradeep Varakantham, and Min Lin. 2024. Bootstrapping language models with dpo implicit rewards. *arXiv preprint arXiv:2406.09760* (2024).

[9] Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, et al. 2024. Data-juicer: A one-stop data processing system for large language models. In *SIGMOD*. 120–134.

[10] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

[11] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168* (2021).

[12] OpenCompass Contributors. 2023. OpenCompass: A Universal Evaluation Platform for Foundation Models. https://github.com/open-compass/opencompass.

[13] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. *41st International Conference on Machine Learning (ICML)* (2024).

[14] Dingsheng Deng. 2020. DBSCAN Clustering Algorithm Based on Density. In *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*. 949–953. https://doi.org/10.1109/IFEEA51475.2020.00199

[15] Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. 2026. Less is More: Improving LLM Alignment via Preference Data Selection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=R2ZJSjLDJC

[16] Yuhao Deng, Chengliang Chai, Kaisen Jin, Linan Zheng, Lei Cao, Ye Yuan, and Guoren Wang. 2025. Two birds with one stone: Efficient deep learning over mislabeled data through subset selection. *SIGMOD* 3, 3 (2025), 1–28.

[17] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *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)*. 2368–2378.

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

[19] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori Hashimoto. 2023. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*. https://openreview.net/forum?id=4hturzLcKX

[20] Shivank Garg, Ayush Singh, Shweta Singh, and Paras Chopra. 2025. IPO: Your Language Model is Secretly a Preference Classifier. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)* (2025).

[21] Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. 2024. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792* (2024).

[22] LIU Hanmo, DI Shimin, LI Haoyang, LI Shuangyin, CHEN Lei, and ZHOU Xiaofang. 2024. Effective Data Selection and Replay for Unsupervised Continual Learning. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 1449–1463.

[23] Botao Hao, YasinAbbasi Yadkori, Zheng Wen, and Guang Cheng. 2019. Bootstrapping Upper Confidence Bound. *Neural Information Processing Systems,Neural Information Processing Systems* (Jan 2019).

[24] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

[25] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. [n.d.]. Measuring Mathematical Problem Solving With the MATH Dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[26] Indra Herdiana, M Alfin Kamal, Triyani, Mutia Nur Estri, and Renny. 2025. A More Precise Elbow Method for Optimum K-means Clustering. arXiv:2502.00851 [stat.ME] https://arxiv.org/abs/2502.00851

[27] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems* 33 (2020), 18661–18673.

[28] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of rlhf on llm generalisation and diversity. *The Twelfth International Conference on Learning Representations (ICLR)* (2024).

[29] Jongwoo Ko, Saket Dingliwal, Bhavana Ganesh, Sailik Sengupta, Sravan Babu Bodapati, and Aram Galstyan. 2025. SeRA: Self-Reviewing and Alignment of LLMs using Implicit Reward Margins. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=uIGnuyDSB9

[30] Fengxin Li, Yi Li, Yue Liu, Chao Zhou, Yuan Wang, Xiaoxiang Deng, Wei Xue, Dapeng Liu, Lei Xiao, Haijie Gu, et al. 2025. LEADRE: Multi-Faceted Knowledge Enhanced LLM Empowered Display Advertisement Recommender System. *Proceedings of the VLDB Endowment* (2025).

[31] Kaiyu Li, Zhongxin Hu, Yuxin Gao, and Yuyang Wu. [n.d.]. DeepSearch: LLM-powered Data Acquisition for Machine Learning. *Proceedings of the VLDB Endowment. ISSN* 2150 ([n. d.]), 8097.

[32] Xiaotian Lin, Yanlin Qi, Yizhang Zhu, Themis Palpanas, Chengliang Chai, Nan Tang, and Yuyu Luo. 2026. Lead: Iterative data selection for efficient llm instruction tuning. *Proceedings of the VLDB Endowment* (2026).

[33] Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. 2025. Skywork-Reward-V2: Scaling Preference Data Curation via Human-AI Synergy. *arXiv preprint arXiv:2507.01352* (2025).

[34] Yuyu Luo, Yihui Zhou, Nan Tang, Guoliang Li, Chengliang Chai, and Leixian Shen. 2023. Learned Data-aware Image Representations of Line Charts for Similarity Search. *Proc. ACM Manag. Data* 1, 1 (2023), 88:1–88:29.

[35] Kaijing Ma, Xeron Du, Yunran Wang, Haoran Zhang, ZhoufutuWen, Xingwei Qu, Jian Yang, Jiaheng Liu, minghao liu, Xiang Yue, Wenhao Huang, and Ge Zhang. 2025. KOR-Bench: Benchmarking Language Models on Knowledge-Orthogonal Reasoning Tasks. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=SVRRQ8goQo

[36] Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. 2025. RewardBench 2: Advancing Reward Model Evaluation. https://huggingface.co/spaces/allenai/reward-bench.

[37] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 645, 11 pages.

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

[39] Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems* 37 (2024), 116617–116637.

[40] Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. 2024. Curry-DPO: Enhancing Alignment using Curriculum Learning & Ranked Preferences. In *the Association for Computational Linguistics: EMNLP 2024*.

[41] Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. 2024. Personalizing reinforcement learning from human feedback with variational preference learning. *Advances in Neural Information Processing Systems* 37 (2024), 52516–52544.

[42] Biqing Qi, Pengfei Li, Fangyuan Li, Junqi Gao, Kaiyan Zhang, and Bowen Zhou. 2024. Online dpo: Online direct preference optimization with fast-slow chasing. *arXiv preprint arXiv:2406.05534* (2024).

[43] Yulei Qin, Yuncheng Yang, Pengcheng Guo, Gang Li, Hang Shao, Yuchen Shi, Zihan Xu, Yun Gu, Ke Li, and Xing Sun. 2025. Unleashing the Power of Data Tsunami: A Comprehensive Survey on Data Assessment and Selection for Instruction Tuning of Language Models. *Transactions on Machine Learning Research* (2025). https://openreview.net/forum?id=RJT1baPhdV Survey Certification.

[44] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems* 36 (2023), 53728–53741.

[45] Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. 2024. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715* (2024).

[46] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[47] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG] https://arxiv.org/abs/1707.06347

[48] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems* 33 (2020), 3008–3021.

[49] 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. 2023. Zephyr: Direct Distillation of LM Alignment. arXiv:2310.16944 [cs.LG] https://arxiv.org/abs/2310.16944

[50] Joannes Vermorel and Mehryar Mohri. 2005. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*. Springer, 437–448.

[51] Tingting Wang, Shixun Huang, Zhifeng Bao, J Shane Culpepper, Volkan Dedeoglu, and Reza Arablouei. 2024. Optimizing data acquisition to enhance machine learning performance. *Proceedings of the VLDB Endowment* 17, 6 (2024), 1310–1323.

[52] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems* 37 (2024), 95266–95290.

[53] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024. β-DPO: Direct Preference Optimization with Dynamic β. *Advances in Neural Information Processing Systems* 37 (2024), 129944–129966.

[54] Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2025. Self-play preference optimization for language model alignment. *The Thirteenth International Conference on Learning Representations (ICLR)* (2025).

[55] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2024. C-Pack: Packaged Resources To Advance General Chinese Embedding. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieve (SIGIR)*.

[56] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2025. WizardLM: Empowering large pre-trained language models to follow complex instructions. arXiv:2304.12244 [cs.CL] https://arxiv.org/abs/2304.12244

[57] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems* 36 (2023), 10935–10950.

[58] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*.

[59] Chi Zhang, Jiacheng Wang, Kun He, Chengliang Chai, Yunpeng Zhang, Yuping Wang, Xu Zhou Linan Zheng, and Lei Cao Lijun Wu, Conghui He. 2026. Data-efficient Online Training for Direct Alignment in LLMs. https://github.com/ZhangChi315/DOTA/Tech_Report.pdf

[60] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (Montreal, Quebec, Canada) *(SIGMOD '96)*. Association for Computing Machinery, New York, NY, USA, 103–114. https://doi.org/10.1145/233269.233324

[61] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. SLiC-HF: Sequence Likelihood Calibration with Human Feedback. *arXiv preprint arXiv:2305.10425* (2023).

[62] Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. 2025. Dpo meets ppo: Reinforced token optimization for rlhf. *Proceedings of the 42nd International Conference on Machine Learning (ICML)* (2025).

[63] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. Agieval: A human-centric benchmark for evaluating foundation models. *the Association for Computational Linguistics: NAACL* (2024).