mathematics

MDPI

Article

Utility-Based Preference Training for Effective Synthetic Text Classification

Jiho Gwak [1] **and Yuchul Jung** [2,*]

[1] Department of Computer & AI Convergence Engineering, Kumoh National Institute of Technology, Gumi-si 39177, Republic of Korea; wlgh6709@kumoh.ac.kr
[2] Department of AI Engineering, Kumoh National Institute of Technology, Gumi-si 39177, Republic of Korea
* Correspondence: jyc@kumoh.ac.kr

Abstract

High-quality synthetic text can mitigate annotation scarcity in text classification. However, standard preference optimization often produces samples that are fluent but weakly label-specific. We present Utility-weighted Direct Preference Optimization (U-DPO), a preference-optimization framework for class-conditional synthetic data generation. In U-DPO, a task-specific classifier provides a margin-based external score for each candidate generation, which is combined with an embedding-based internal similarity score to form an overall utility. These utilities are used (i) to mine preference pairs from multiple candidates per class and (ii) to weigh each DPO update by the utility gap between preferred and dispreferred samples. This design encourages the generator to concentrate on learning informative, label-discriminative preference comparisons rather than treating all pairs equally. Across two multiclass scientific-abstract benchmarks (arXiv and WOS-11967), U-DPO consistently improves downstream SciBERT classification accuracy compared with both vanilla synthetic generation and standard DPO fine-tuning, with gains up to 0.88 percentage points on arXiv and 0.83 percentage points on WOS-11967 depending on the generator. An additional GPT-4.5-based evaluation also indicates a higher mean quality score for U-DPO samples with reduced variance.

Keywords: Direct Preference Optimization (DPO); synthetic data generation; text classification; Large Language Models (LLM); utility-based learning; margin-based optimization

MSC: 68T05; 68T50

1. Introduction

Text classification is a core task in natural language processing (NLP), underpinning applications from sentiment analysis to topic categorization. Recent Large Language Models (LLMs) have made it increasingly feasible to generate labeled synthetic text to supplement limited human annotations [1,2]. When effective, synthetic augmentation can reduce labeling costs and expand data coverage, which is particularly valuable in low-resource settings [3,4].

A central practical challenge; however, is label specificity [5]. Prompt-based generations can be fluent and coherent yet only weakly discriminative for the intended class. Such label-ambiguous samples may provide a limited learning signal and can even degrade downstream performance due to noise and distribution mismatch [6–8].

To improve the usefulness of synthetic data, prior work has explored stronger prompting strategies and filtering mechanisms. In parallel, preference optimization has emerged

Academic Editor: Shuai Liu

Received: 20 December 2025
Revised: 21 January 2026
Accepted: 28 January 2026
Published: 31 January 2026

as a powerful paradigm for aligning LLM outputs, including Reinforcement Learning from Human Feedback (RLHF) [9] and Direct Preference Optimization (DPO) [10]. DPO is particularly attractive because it offers a stable, reinforcement-learning-free objective and has been extended to better handle preference strength and robustness [11–13].

Despite these advances, an important gap remains for classification-oriented synthetic generation. Standard preference optimization typically treats all labeled preference pairs (preferred vs. dispreferred) as similarly informative. In class-conditional generation; however, preference pairs can differ substantially in downstream utility, which comprises both label discriminativeness and intrinsic text quality. A "dispreferred" sample is not necessarily useless; it may still contain strong class-discriminative cues and exhibit high linguistic fluency. In such cases, blindly forcing the model away from a valid dispreferred sample can be counterproductive. This suggests that preference learning for synthetic data should not apply uniform updates. Instead, it should account for the size of the utility gap (reflecting differences in semantic quality and classification confidence) to avoid penalizing high-quality candidates.

To address this gap, we propose Utility-weighted Direct Preference Optimization (U-DPO). U-DPO leverages utility in two complementary ways. First, it performs utility-guided pair construction to prioritize informative preference pairs. Second, it applies utility-gap weighting to reduce updates on ambiguous pairs, especially when the dispreferred candidate may still be beneficial for classification.

We focus on scientific abstract classification because it is a realistic setting where labeled data are costly and domain-specific terminology increases label ambiguity. Moreover, classes often exhibit distinct writing conventions (e.g., characteristic terminology and structure), so class-conditional generation must capture both topical content and label-specific cues. This makes scientific abstracts a strong stress test for utility-based preference training, where label-discriminativeness is critical. While our experiments focus on scientific corpora, the proposed utility-weighted preference mechanism is model-agnostic and can extend to other classification domains.

Accordingly, we evaluate U-DPO on two multiclass scientific-abstract classification benchmarks (arXiv and WOS-11967), comparing against prompt-based synthetic generation and standard DPO tuning. Across multiple open-source text generators, U-DPO consistently improves downstream SciBERT classification accuracy and produces synthetic samples with stronger label consistency. This demonstrates that incorporating task utility into preference optimization yields more effective synthetic training data.

Our main contributions are as follows:

1. Task-utility preference construction: We introduce a utility-based mechanism to mine class-conditional preference pairs from multiple candidate generations using classifier-margin confidence and embedding-based semantic quality signals.
2. Utility-gap–weighted preference optimization: We propose U-DPO, which scales DPO updates by the utility gap between preferred and dispreferred samples to focus learning on more informative comparisons and avoid counterproductive updates.
3. Empirical validation on scientific multiclass benchmarks: We demonstrate consistent downstream improvements on arXiv and WOS-11967 across multiple generators, supported by analyses showing enhanced label consistency of generated samples.

2. Related Work

2.1. LLMs for Text Classification

Recent studies have explored the capabilities of LLMs in text classification through zero-shot and few-shot prompting, as well as fine-tuning [14,15]. These works show that LLMs can often perform surprisingly well without task-specific training data, but

their effectiveness varies by task and setting [16]. LLMs have demonstrated competitive performance in specialized tasks such as scientific edit intent classification [17]. However, large-scale evaluations find that zero-shot prompting is most effective on simpler tasks like sentiment analysis. Fine-tuned models, even smaller ones, often remain stronger on more complex classification problems [18]. A recent multilingual study found that smaller fine-tuned transformers can even surpass few-shot LLMs in accuracy across most categories, suggesting that in-context learning alone is often insufficient for optimal performance [19].

2.2. Prompt-Based Synthetic Data Generation

Prompt-based synthetic data generation has emerged as a promising strategy for training text classifiers [3]. Instead of manually collecting or annotating data, researchers prompt LLMs to produce labeled examples, which can be used to augment or even replace human-labeled training sets [5]. Such approaches have shown growing effectiveness for domain adaptation and general-purpose classification [20]. For instance, recent studies show that LLMs can generate domain-general sentiment datasets [21], fully synthetic training corpora without human labels [22], and even code-mixed data for multilingual sentiment classification [23]. Despite these successes, important limitations have also been observed. In one health-related classification task, augmenting an unbalanced dataset with GPT-generated samples did not produce performance improvements [6]. This finding suggests that the utility of synthetic data depends on the target task. It also indicates that generation strategies must be carefully tailored to avoid issues such as bias inheritance or model collapse [24,25]. These findings suggest that synthetic data can be fluent yet not classifier-useful. This motivates utility-weighted preference learning that prioritizes label-discriminative samples and down-weights ambiguous generations.

2.3. Preference Optimization for Text Generation

Aligning LLMs with human values is critical for their safe and effective deployment. The standard approach, Reinforcement Learning from Human Feedback (RLHF) [9], optimizes a model to produce outputs preferred by humans and has been widely used to train aligned language models [26,27]. However, RLHF can be complex and unstable, especially in classification settings where defining reliable reward functions is challenging [28]. To address these issues, Direct Preference Optimization (DPO) [10] was introduced as a more stable, RL-free alternative that learns directly from preference data. Building on this idea, subsequent work has adapted DPO to more complex alignment tasks. For instance, variants like IPO address overfitting [11], while KTO learns from binary good/bad labels instead of pairs [12]. Other approaches introduce adaptive reward margins or offsets to better reflect preference strength during optimization, such as AlphaDPO [13] and ODPO [29]. These advances show that preference optimization can be adapted to diverse task requirements through objective-level modifications, providing a practical and scalable framework for alignment. Our work complements this line by making the training emphasis task-specific for synthetic data generation. Unlike approaches that modify the optimization objective with static margins, we propose a dynamic reweighting mechanism based on a composite sample utility. Concretely, we define a task-specific utility by combining two factors, label discriminativeness and intrinsic text quality. We then use the utility gap between the preferred and dispreferred samples to scale the gradient of each update. This strategy amplifies learning from high-contrast pairs that significantly improve both class separability and text well-formedness, while effectively dampening the signal from ambiguous near-ties. By prioritizing informative comparisons over noisy ones, U-DPO tailors preference optimization specifically for downstream classification performance.

3. Theoretical Analysis

We briefly review DPO and its theoretical basis before introducing our utility-based extension.

3.1. RLHF Objective and Optimal Policy

The standard RLHF objective seeks a policy π_θ that maximizes the expected reward from a learned reward model $r_\phi(x, y)$ while remaining close to a reference policy π_{ref}, typically a supervised fine-tuned model. This is formulated as:

$$\max_\theta \ \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[r_\phi(x,y)] \ - \ \beta\, D_{KL}(\pi_\theta(y \mid x) \ \| \ \pi_{\text{ref}}(y \mid x)), \tag{1}$$

where $\beta > 0$ controls the strength of the KL penalty. Under mild assumptions, the optimal policy π^* has the closed form

$$\pi^*(y \mid x) = \frac{1}{Z(x)}\, \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r^*(x,y)\right), \tag{2}$$

where $Z(x)$ is a normalizing partition function. This relation connects the optimal policy π^* to an implicit reward function r^*.

3.2. The Bradley-Terry Model and the DPO Objective

DPO leverages this mapping by modeling human preferences via the Bradley–Terry model [30]. Given a pair of candidate responses (y_w, y_l) for the same input x, the probability that y_w is preferred over y_l is

$$p^*(y_w \succ y_l \mid x) = \sigma(r^*(x, y_w) - r^*(x, y_l)), \tag{3}$$

where $\sigma(\cdot)$ is the logistic sigmoid. Substituting the policy-based representation of r^* from (2), ref. [10] express this probability in terms of policies:

$$p^*(y_w \succ y_l \mid x) = \sigma(\beta \log \frac{\pi^*(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi^*(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}). \tag{4}$$

Using an implicit reward $\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$, the DPO loss is defined as the negative log-likelihood over a dataset of preference triplets $\mathcal{D}_{\text{pref}} = \{(x, y_w, y_l)\}$:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{pref}}}[\log \sigma(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l))]. \tag{5}$$

This objective directly optimizes π_θ to respect observed preferences, without training an explicit reward model or performing RL.

3.3. Utility-Weighted DPO

In our setting, preference pairs are induced by a proxy utility signal, and their reliability can vary substantially. Pairs with small utility gaps behave like near-ties and are more sensitive to noise or miscalibration in the utility scorer. We therefore model U-DPO as a weighted variant of DPO that assigns each pair a nonnegative weight w based on its utility gap. Here, w represents the confidence in the induced preference. Larger utility gaps indicate more reliable preferences, whereas near ties are likely to produce unstable preference labels and weak learning signals.

Concretely, we optimize a weighted Bradley–Terry negative log-likelihood:

$$\mathcal{L}_{\text{U-DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{pref}}}[w(x, y_w, y_l) \ * \ \log \sigma(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l))]. \tag{6}$$

When $w(x, y_w, y_l) = 1$ for all pairs, this reduces to standard DPO. Intuitively, weighting emphasizes more informative comparisons while down-weighting ambiguous pairs, which is desirable when preferences are automatically constructed from task utility.

4. Utility-Weighted DPO for Class-Conditional Synthetic Data

Standard DPO is agnostic to the downstream task: it aligns with whatever preferences are provided (e.g., helpful vs. unhelpful responses). For synthetic data generation in classification, we seek preferences that reflect how useful a sample is for training a classifier. Moreover, preference pairs are not equally informative in this setting; thus, applying uniform updates across pairs can be suboptimal. Unlike prior methods, we leverage a task-specific utility to adaptively adjust the update strength for each sample. This approach emphasizes comparisons that improve class separability while down-weighting ambiguous pairs. Crucially, the utility is composite: it dynamically reflects the relative difference between candidate samples while also accounting for intrinsic text quality, enabling U-DPO to favor both label-aligned and well-formed generations. An overview of the entire framework is illustrated in Figure 1.

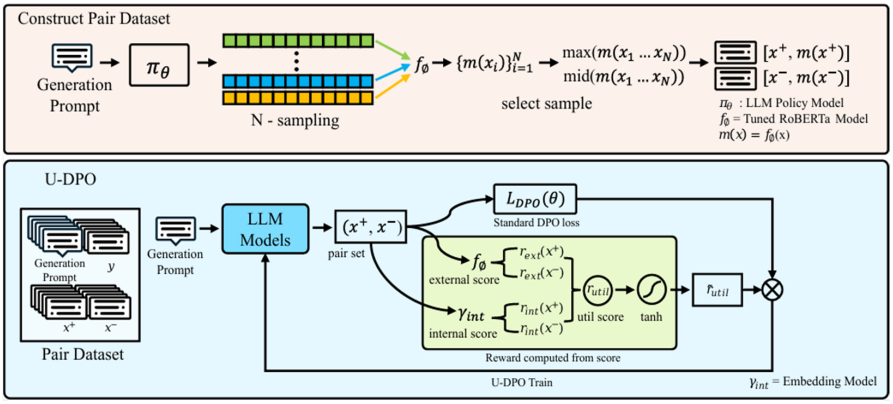

Figure 1. Overview of our Utility-weighted DPO (U-DPO) framework for class-conditional synthetic text generation.

4.1. Problem Setup

Let $\mathcal{C} = \{c_1, \dots, c_K\}$ denote the set of class labels. Our goal is to train a conditional language model $\pi_\theta(x \mid c)$ that generates synthetic texts x such that, for each label c, the resulting dataset is effective for training a downstream classifier.

We adopt a preference-learning view. For each label c, we consider pairs of candidate generations (x^+, x^-) intended to reflect relative quality with respect to the class semantics, where x^+ is the preferred (higher-utility) sample and x^- the less preferred one.

4.2. Utility-Based Preference Signal

Standard DPO typically relies on preference pairs, often human-annotated, that reflect generic notions of quality.

In contrast, our Utility-weighted DPO (U-DPO) redefines the preference signal in terms of a task-specific utility function r_{util} designed to approximate the value of a sample for classification. Instead of directly modeling a latent reward r^*, we construct preference pairs automatically based on utility differences.

We view this as a form of reward shaping within the DPO framework: r_{util} is a proxy for the ideal reward, namely, downstream classification performance. The utility function combines external label confidence and internal semantic quality, described below.

However, employing a classifier-defined utility warrants careful consideration regarding feedback loops. Using such a utility may introduce bias propagation if the auxiliary

classifier is biased or miscalibrated. This could potentially amplify spurious cues in the generated samples.

A related risk is circular evaluation, which occurs when the utility scorer and downstream evaluator are identical or closely aligned. This overlap can cause the model to overfit to the evaluator's peculiarities, thereby exaggerating the apparent gains. To mitigate direct circularity, we distinguish the models used in our pipeline. Specifically, we use RoBERTa for utility scoring while reserving SciBERT solely for downstream evaluation.

4.3. Data Construction and Utility Function

Since collecting human-labeled preference pairs is expensive, we construct them automatically using an auxiliary classifier f_ϕ trained on available real data.

For each class $y \in \mathcal{C}$,

1. Sample $n = 8$ candidate texts $\tilde{x}_1, \ldots, \tilde{x}_n$ from $\pi_\theta(\cdot \mid y)$. Candidate texts are generated with a fixed 2-shot academic-abstract prompt to ensure a consistent scientific style across classes. The exact prompt template is provided in Appendix A.
2. Compute a margin score for each candidate using f_ϕ:

$$m(\tilde{x}) = f_\phi(\tilde{x})_y - \max_{j \neq y} f_\phi(\tilde{x})_j, \tag{7}$$

where $f_\phi(\tilde{x})_j$ is the predicted probability for class j. Candidates with higher $m(\tilde{x})$ are more confidently classified as a label y.

3. Select high-margin candidates as preferred x^+ and medium-margin candidates as less preferred x^-, forming automatic preference pairs. To mitigate bias, the utility classifier is trained only on the real training split and is kept fixed. It is never trained on synthetic samples, and it does not access the downstream test set.
4. Margin scores alone; however, they may be noisy or biased and do not capture aspects such as fluency or coherence. We therefore introduce an internal utility term $r_{int}(x)$ that measures the well-formedness of a candidate text independent of label confidence. Concretely, we implement $r_{int}(x)$ as a MiniLM-based text-quality scorer: a pretrained MiniLM (microsoft/MiniLM-L12-H384-uncased) encoder $E(\cdot)$ is kept frozen, and a lightweight linear head g(·) is trained to predict a continuous quality score in [0, 1]. Given a candidate text x, we obtain its representation $h(x) = E(x)$ (final-layer [CLS] embedding) and compute

$$r_{int}(x) = \sigma(g(h(x))), \tag{8}$$

where σ(·) denotes the sigmoid function. To train the quality head, we use the Ultrafeedback dataset, which provides supervision signals aimed at assessing intrinsic text quality independent of the downstream classification task. In other words, the head is optimized to score how good the text is as text, rather than how confidently it supports a particular label.

We define the external utility $r_{\text{ext}}(\tilde{x}, y)$ using the auxiliary classifier margin:

$$r_{ext}(\tilde{x}, y) = m(\tilde{x}), \tag{9}$$

Since the raw margin scale can vary across datasets and classifier calibrations, we apply min-max normalization over the candidate pool used for pair construction:

$$\widehat{r_{\text{ext}}}(\tilde{x}, y) = \hat{m}(\tilde{x}) = \frac{m(\tilde{x}) - m_{min}}{m_{max} - m_{min}} \tag{10}$$

Finally, we combine the internal text-quality score and the external label-confidence score into a single utility used for preference learning:

$$r_{\text{util}}(\widetilde{x}, y) = \lambda\, r_{\text{int}}(\widetilde{x}) + (1-\lambda)\, \widehat{r_{\text{ext}}}(\widetilde{x}, y), \tag{11}$$

where $0 \le \lambda \le 1$ balances semantic quality and label confidence. To focus optimization on informative pairs, we weight each pair by a modulation factor depending on the absolute utility gap. Additional details and minimal ablation studies on λ and the contribution of r_{int} and r_{ext} are deferred to Appendix A.

$$w(x^+, x^-) = \; r_{\text{util}}(x^+, y) - r_{\text{util}}(x^-, y). \tag{12}$$

The above equation down-weights training signals for pairs whose utility values are similar, as such pairs are likely to provide ambiguous supervision.

4.4. U-DPO Objective

Given preference pairs (x, x^+, x^-) constructed as above, we define the U-DPO loss by re-weighting the standard DPO loss with the modulation factor:

$$\mathscr{L}_{\text{U-DPO}}(\theta) = \mathbb{E}_{(x,x^+,x^-)}[w(x^+, x^-)\; \ell_{\text{DPO}}(x, x^+, x^-; \theta)], \tag{13}$$

where ℓ_{DPO} denotes the per-triplet contribution to the DPO loss in (5). In practice, we implement U-DPO by multiplying the standard DPO loss for each pair by $w(x^+, x^-)$, so that pairs with stronger utility differences exert stronger gradients.

Intuitively, U-DPO encourages the generator π_θ to produce outputs that are simultaneously (i) semantically plausible and (ii) highly discriminative under the classifier, thereby optimizing directly for the downstream classification task.

For clarity, Table 1 summarizes the key utility components used in U-DPO, along with the definition of the pair weight w.

Table 1. Summary of utility components in U-DPO.

Term	Definition	Range	Purpose
$\mathrm{r_{int}}$	$\sigma(\mathrm{g}(\mathrm{h}(\mathrm{x})))$	[0, 1]	Intrinsic text quality
r_{ext}	$m(\widetilde{x}) = f_\phi(\widetilde{x})_y - \max_{j \neq y} f_\phi(\widetilde{x})_j$	$\mathbb{R}$	Label confidence from auxiliary classifier
$\hat{r_{\text{ext}}}$	$\hat{m}(\widetilde{x}) = \frac{m(\widetilde{x}) - m_{min}}{m_{max} - m_{min}}$	[0, 1]	Min–max normalized margin to match scales with $\mathrm{r_{int}}$
r_{util}	$\lambda\, r_{\text{int}}(\widetilde{x}) + (1-\lambda)\, \hat{r_{\text{ext}}}(\widetilde{x}, y)$	[0, 1]	Overall utility combining quality + label evidence
λ	Mixing weight in r_{util}	[0, 1]	Trade-off: $\lambda\uparrow$ quality emphasis, $\lambda\downarrow$ label confidence emphasis
w	$r_{\text{util}}(x^+, y) - r_{\text{util}}(x^-, y)$	[0, 1]	Pair weight: down-weights ambiguous pairs and emphasizes informative utility gaps.

Trade-off: $\lambda\uparrow$ indicates quality emphasis, whereas $\lambda\downarrow$ indicates label-confidence emphasis.

5. Computational Complexity

We analyze the computational overhead of U-DPO relative to standard DPO and simple supervised fine-tuning.

5.1. Complexity of DPO Fine-Tuning

In standard DPO, each training step processes a batch of preference triplets (x, y_w, y_l). The main cost arises from computing log-probabilities under both the policy model π_θ and the frozen reference model π_{ref}, requiring two forward passes per candidate. For a Transformer-based model with sequence length N and hidden dimension d, a single forward pass has time complexity $\mathcal{O}(N^2 d)$ [31]. For batch size B, the per-step complexity is roughly:

$$\mathcal{O}(B \cdot 2 \cdot (T_{\pi_\theta} + T_{\pi_{\text{ref}}})) \approx \mathcal{O}(B N^2 d),$$

where T_{π_θ} and $T_{\pi_{\text{ref}}}$ denote single forward-pass costs, which are typically comparable as the architectures are matched. Memory usage is also substantial, since activations for both models must be stored [32].

5.2. Additional Complexity of U-DPO

U-DPO introduces an offline data-construction phase to build utility-filtered preference pairs. For each class $c \in \mathcal{C}$, this phase comprises:

1. Candidate Generation. For each prompt, we generate n candidate sequences from the generator π_θ, incurring complexity $\mathcal{O}(n N_{\text{gen}}^2 d)$, where N_{gen} is the length of the generated sequences.
2. Margin Scoring. Each candidate is fed to the classifier f_ϕ (e.g., SciBERT) of dimension d_{cls}, with complexity $\mathcal{O}(n N_{\text{cls}}^2 d_{\text{cls}})$.
3. Internal Utility Scoring. Each candidate is encoded by an embedding model (e.g., MiniLM) to compute r_{int}, with complexity $\mathcal{O}(n N_{\text{emb}}^2 d_{\text{emb}})$.

Thus, the total per-prompt cost is dominated by candidate generation and the two utility-scoring passes. Since preference-pair construction is performed once before U-DPO fine-tuning, this overhead is front-loaded and can be amortized across training runs. Moreover, in our implementation, we pre-compute offline to avoid keeping multiple scoring models on the GPU during fine-tuning; as a result, U-DPO training incurs only a negligible additional cost from simple scalar weighting. Table 2 reports the resulting wall-clock overhead measured in our setup.

Table 2. Runtime breakdown and additional overhead introduced by U-DPO. Units are minutes for training phases and milliseconds for inference and scoring steps.

Component	Time	Notes
External Scorer training	22 min	One-time
Internal Scorer training	25 min	One-time
Candidate generation (8 samples)	16,640 ms	Baseline
External Scoring	206 ms	Overhead
Internal Scoring	50 ms	Overhead
Total Overhead	257 ms	≈1.54%

6. Experiments

6.1. Experimental Setup

We evaluate our approach on two multiclass scientific classification datasets, arXiv [33], with 11 classes, and WOS-11967 [34], with 33 classes. Both datasets consist of scholarly abstracts.

The overall experimental configuration is summarized in Table 3. We use SciBERT (uncased) [35] as the classifier backbone and MiniLM-L12-H384-uncased [36] as the embedding model for computing semantic utility scores. Prior studies have shown that SciBERT consistently outperforms BERT [37] and RoBERTa [38] on scientific NLP benchmarks, highlighting its domain relevance.

Table 3. Summary of experimental configuration.

Dataset	**arXiv**	**WOS-11967**
Category	11	33
Train set	28,388	9473
Test set	2500	2394
Component	Details	
Classification Model	SciBERT	
Utility Function Model	MiniLM RoBERTa	
LLMs for Generation	LLaMA 3.2 1B, LLaMa 3.2 3B, Phi-4-mini 3.8B	
DPO Settings	$\beta = 1.0$ learning rate = 2×10^{-5} batch size = 2 epochs = 3	

Synthetic training data is generated using three open-source LLMs—LLaMA 3.2 1B, 3B [39], and Phi-4-mini [40]—with class-conditional prompts. DPO training is performed using HuggingFace TRL with custom modifications to incorporate our utility-aware pair selection. All experiments are conducted on a single NVIDIA A6000 GPU. Synthetic data are generated using a 2-shot prompting strategy, where two randomly selected examples with the same label are used as input to the LLM. For each prompt, we generate (n = 5) samples. For evaluation, we report accuracy.

6.2. Training-Time Preference Consistency

To evaluate how well the generator aligns with the preference signal during training, we compute the DPO reward accuracy. This metric is defined as the fraction of preference pairs in which the model assigns a higher log-probability to the preferred sample x^+. Figure 2 plots reward accuracy over training steps for standard DPO and U-DPO. U-DPO consistently achieves higher reward accuracy, indicating that utility-filtered pairs provide a cleaner, more informative training signal. In contrast, standard DPO exhibits more fluctuations, likely due to noise from unfiltered pair selection.

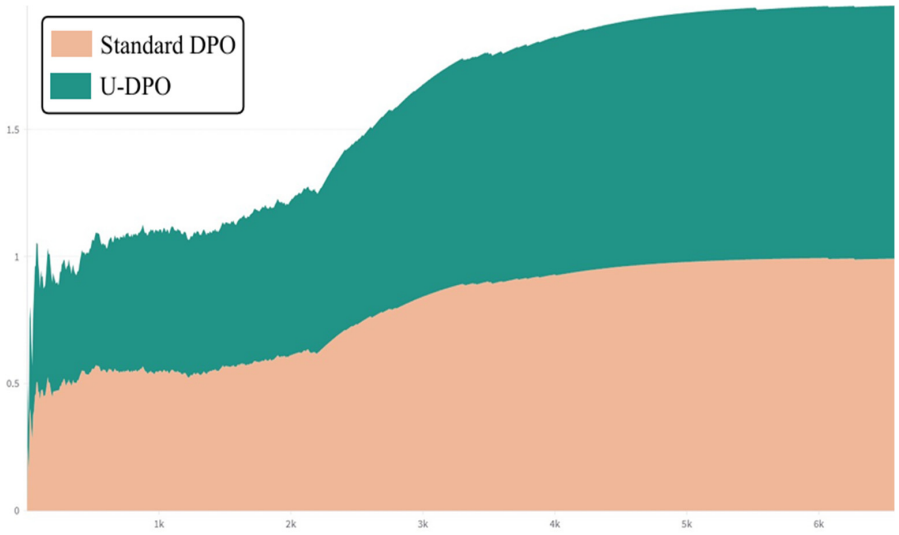

Figure 2. DPO reward accuracy over the course of training for standard DPO and U-DPO.

6.3. Margin-Based Quality Assessment

To verify whether U-DPO enhances the class consistency of generated text, we evaluate synthetic samples using the margin score defined in Section 4.3. We analyze this score using samples generated by the Phi-4-mini model on the ArXiv dataset. As shown in Figure 3, both standard DPO and U-DPO result in higher average and median margin scores compared to generation without preference optimization. Notably, U-DPO yields further gains, suggesting that incorporating utility signals strengthens the model's ability to generate label-consistent outputs.

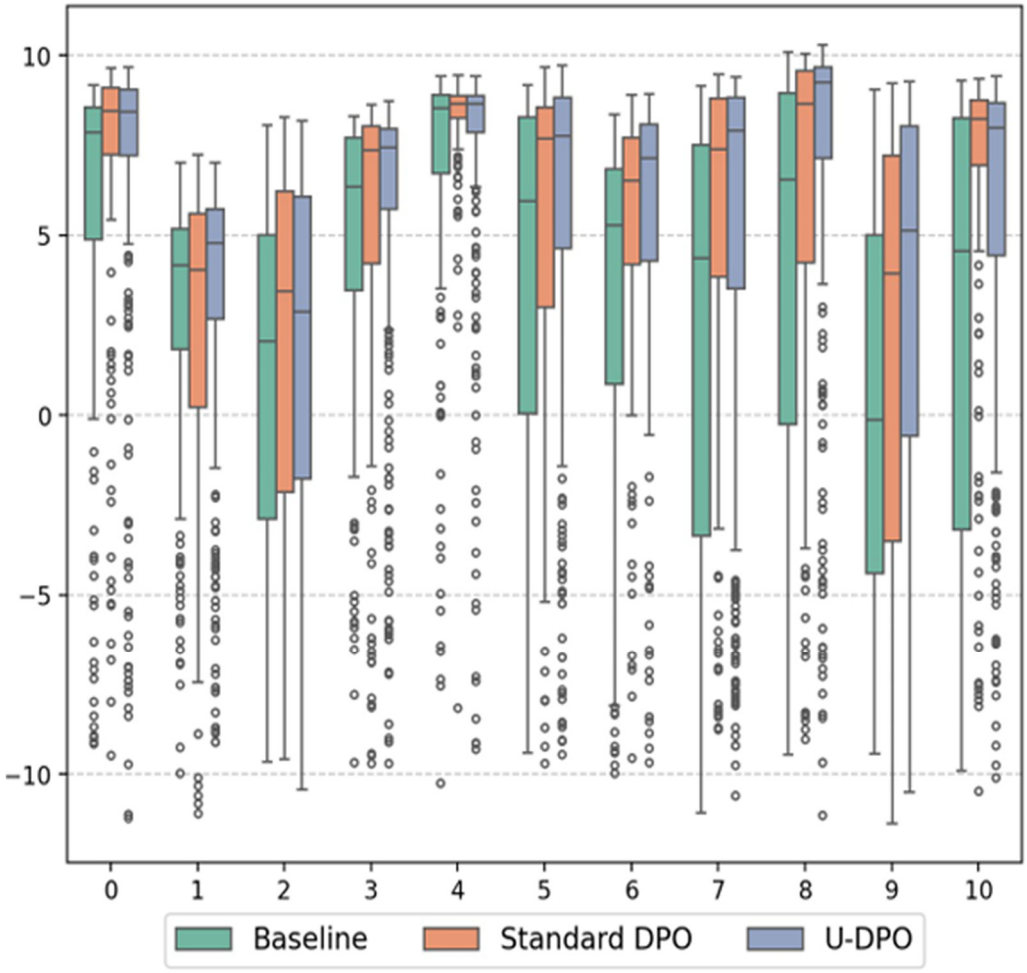

Figure 3. Margin score distributions for synthetic samples generated from identical prompts using the Phi-4-mini model on the arXiv dataset. We compare three training regimes: baseline (no preference optimization), standard DPO, and U-DPO.

6.4. Classification Performance with Synthetic Data

To directly assess the classification performance, we train models exclusively on generated samples from the arXiv dataset. We compare three regimes: (1) generation from the base LLM, (2) generation using standard DPO, and (3) generation via our U-DPO framework. As shown in Figure 4, classifiers trained using U-DPO samples consistently outperform those trained on data from the base LLM and standard DPO. This demonstrates that utility-based training leads to significant improvements in downstream task performance.

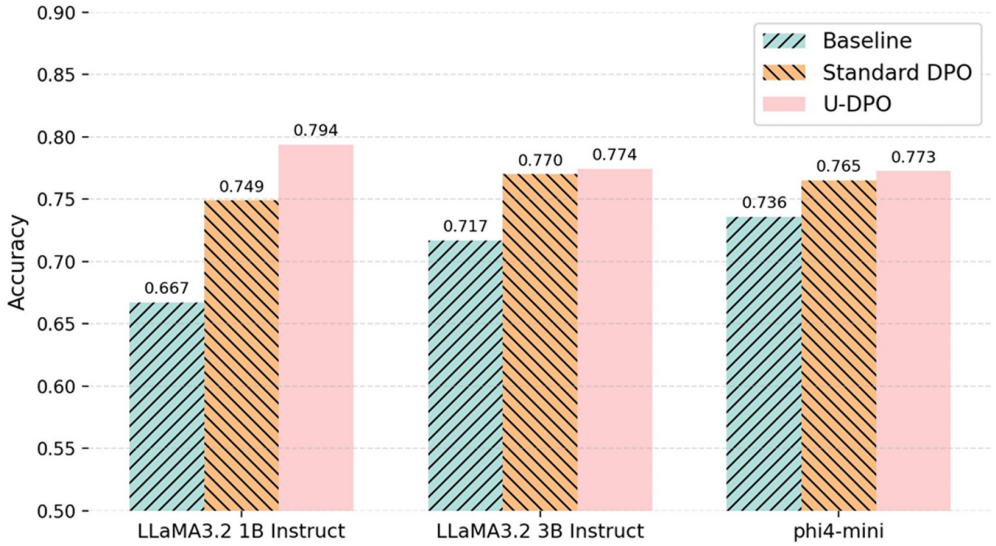

Figure 4. Downstream classification accuracy of models trained exclusively on synthetic data generated under three training regimes: baseline (no preference optimization), standard DPO, and U-DPO. All models are evaluated on the arXiv test set.

6.5. Evaluating Synthetic–Real Data Augmentation

To evaluate practical utility, we measure performance when combining synthetic samples with a fixed subset of real data. We vary the number of generated samples per class and found that 50 samples per class achieved the highest performance on average.

As shown in Table 4, augmenting real data with synthetic samples leads to consistent accuracy improvements across both datasets. U-DPO yields the most substantial gains, indicating its effectiveness in hybrid settings. We also compare against GPT-4o prompting baselines, and classifiers trained on U-DPO synthetic data outperform both zero-shot and few-shot setups.

Table 4. Classification accuracy of the fine-tuned SciBERT baseline, SciBERT prompting, and GPT-4o prompting on the arXiv and WOS-11967 datasets. Results of open-source LLMs—LLaMA 3.2 1B (denoted LLaMA-1B), LLaMA 3.2 3B (LLaMA-3B), and Phi-4-mini (Phi-4)—are also included for comparison. All values are means over 20 runs. *p*-values (two-sided paired *t*-tests over 20 matched runs) are reported below the table.

SciBERT Baseline Dataset		**Accuracy**		**Dataset**		**Accuracy**		
arXiv		0.8828		WOS-11967		0.9005		
Dataset	Method	LLaMA-1B	LLaMA-3B	Phi-4	Zero Shot	k = 2	k = 3	k = 5
arXiv	Prompt-base	-	-	-	0.78	0.85	0.86	0.88
	Base Synthetic	0.8864	0.8868	0.8824	-	-	-	-
	DPO Synthetic	0.8872	0.8876	0.8904	-	-	-	-
	U-DPO	0.8884	0.8896	**0.8912**	-	-	-	-
WOS	Prompt-based	-	-	-	0.64	0.78	0.82	0.85
	Base Synthetic	0.9076	0.8885	0.8824	-	-	-	-
	DPO Synthetic	0.9118	0.9112	0.8904	-	-	-	-
	U-DPO	0.9143	**0.9156**	0.9143	-	-	-	-

Furthermore, paired *t*-tests on 20 independent runs indicate that U-DPO consistently and significantly outperforms both the baseline ($p < 0.001$) and standard DPO ($p < 0.05$). As shown in Figure 5, U-DPO also produces more stable and higher-accuracy distributions between trials, strengthening the case for its robustness.

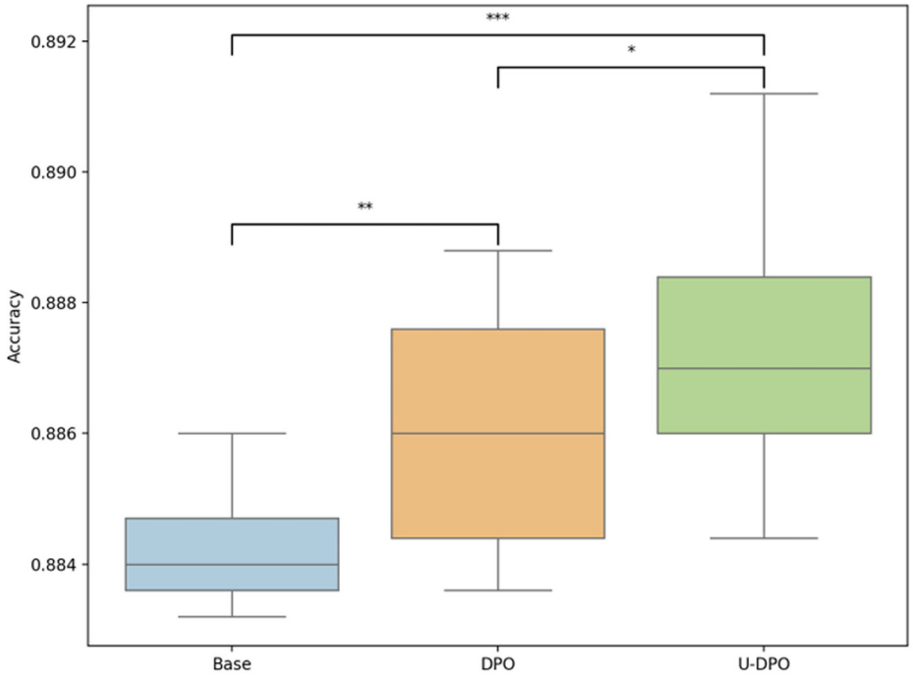

Figure 5. Each point represents one run with a different random seed (n = 20), and the box shows the median and interquartile range. "*" indicates statistical significance of U-DPO over the compared method under a paired *t*-test across the same seeds (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$).

6.6. LLM-Based Evaluation with GPT-4.5

To assess the quality of the generated synthetic samples, we employ GPT-4.5 as an automated evaluator. Each sample is rated on a 0–5 scale based on its relevance, fluency, and class alignment. A total of 132 synthetic samples were evaluated. Table 5 reports the statistics of scores assigned to generations from Standard DPO and U-DPO. U-DPO achieves a higher average score (4.14 vs. 4.05) and a lower standard deviation (0.89 vs. 1.09), suggesting it produces more consistent and higher-quality outputs.

To complement classifier-based metrics, we assess the intrinsic quality of generated samples using GPT-4.5 as an automatic evaluator. We sample 132 synthetic texts and ask GPT-4.5 to rate each on a 0–5 scale with respect to relevance, fluency, and alignment with the intended class. As in Table 3, U-DPO samples achieve a higher mean score (4.14 vs. 4.05) and lower standard deviation (0.89 vs. 1.09) than standard DPO, indicating more consistently high-quality outputs. Medians are comparable (4.5), suggesting that U-DPO primarily reduces variance and tail failures.

Table 5. GPT4.5-based Evaluation of Synthetic Samples from the Standard DPO and U-DPO.

Statistic	Standard DPO	U-DPO
Mean	4.05	4.14
Median	4.50	4.50
Std. Dev.	1.09	0.89

6.7. Discussion of Experimental Findings

Overall, the experiments show that U-DPO yields consistent improvements. The gains are observed across multiple generators and datasets. This suggests that utility-weighted preference training is robust to the choice of generator. Importantly, these trends are stable over 20 independent runs with different random seeds, indicating that the improvements are not driven by a favorable single run but reflect a reproducible effect.

Gains are observed in:

- Margin-based label consistency—measured by the auxiliary classifier's target-vs-competitor margin, indicating stronger class-discriminative cues in generated samples (Section 6.3);
- Downstream classification accuracy—evaluated under both the synthetic-only training regime and the hybrid real–synthetic augmentation setting (Sections 6.4 and 6.5);
- LLM-based quality assessments—ratings of fluency/relevance and class alignment that corroborate the quantitative results (Section 6.6).

We also note that smaller models, such as LLaMA 3.2 1B, generally benefit less than larger models, suggesting that model capacity influences the effectiveness of utility-based preference optimization. Nonetheless, even the smaller models show consistent trends in the same direction. Interestingly, the downstream gains in the full-data regime are modest for LLaMA 3.2 1B. However, U-DPO yields substantial improvements at the generation stage. It produces more label-discriminative and higher-utility samples, leading to clearer gains in the synthetic augmentation setting where synthetic data quality matters most.

7. Limitations

Despite its benefits, U-DPO has several limitations:

First, U-DPO depends on the quality of the auxiliary classifier f_ϕ and the resulting utility signal. Although we reduce direct circularity by using RoBERTa for utility scoring and SciBERT for downstream evaluation, the utility remains classifier-defined and may still encode bias or miscalibration. If f_ϕ is biased or poorly calibrated, U-DPO may over-optimize this proxy signal rather than true generalization performance [41].

Second, our empirical evaluation is restricted to scientific article classification on two benchmark datasets. It remains unclear how well U-DPO transfers to other types of tasks, such as short-form social media texts, multi-label classification, or domains with fuzzier label boundaries and higher stylistic variance.

Third, reliance on synthetic data—even when quality-controlled—raises concerns about model collapse and bias amplification [24,25]. Iteratively training on synthetic data generated by models that were themselves trained on synthetic or biased corpora can reduce diversity and distort the learned distribution. While our hybrid experiments, which mix real and synthetic samples, show promising results, the long-term impact of heavy synthetic data usage warrants further study.

Finally, U-DPO introduces computational overhead due to the offline generation and scoring of candidate samples. Although we restrict experiments to relatively lightweight models (up to 4B parameters), scaling U-DPO to larger models or to label spaces with many classes and fine-grained distinctions could be costly.

8. Conclusions

We presented Utility DPO (U-DPO), a utility-based preference optimization framework for class-conditional synthetic text generation in multiclass classification. U-DPO redefines the preference signal using a task-specific utility that combines classifier margin (label discriminativeness) with semantic quality, enabling LLMs to generate synthetic data that is more label-consistent and more useful for downstream training than baseline generation or standard DPO tuning.

Across arXiv and WOS-11967, classifiers trained on U-DPO-generated data consistently outperform those trained on base or DPO synthetic data in both fully synthetic and hybrid real–synthetic settings. Complementary analyses—including margin-based label consistency and LLM-based quality assessment—support the same conclusion: incorporating task utility into preference optimization steers generation toward classification-relevant

signal, not only surface fluency. Overall, our results highlight task-specific preference optimization as a practical route to more data-efficient model development with reduced reliance on large-scale manual annotation.

9. Future Work

Richer utility signals. Future work could extend γ_{util} beyond a single classifier's margin by incorporating ensemble agreement, uncertainty estimates, or more nuanced heuristics to better approximate true downstream utility and reduce noise [42].

Broader tasks and domains. Applying U-DPO to short-form text, multi-label or hierarchical classification, code classification, and domain-specific settings (e.g., medical or legal text) would further test generality and robustness [7]. Cross-modal extensions, such as generating text conditioned on images, are also promising [43].

Efficiency improvements. Since utility scoring is a major source of overhead, techniques such as ranking distillation, selective pair mining, caching, or joint training of the generator and auxiliary classifier could reduce computation while maintaining or improving generation quality.

Scalability across resource regimes. While this work demonstrates the efficacy of U-DPO in data-constrained settings, extending the analysis to moderate-to-high resource regimes remains an important direction to characterize the scaling behavior and potential saturation points of the proposed method.

Author Contributions: Conceptualization, J.G. and Y.J.; methodology, J.G.; software, J.G.; validation, J.G.; formal analysis, J.G.; investigation, J.G.; resources, J.G.; data curation, J.G.; writing—original draft preparation, J.G.; writing—review and editing, Y.J.; visualization, J.G.; supervision, Y.J.; project administration, Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the IITP (Institute of Information & Communications Technology Planning & Evaluation)-ICAN (ICT Challenge and Advanced Network of HRD) grant funded by the Korea government (Ministry of Science and ICT) (IITP-2026-RS-2022-00156394) and the Gyeongsangbuk-do RISE (Regional Innovation System & Education) project (Regional Growth Innovation LAB unit).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used OpenAI GPT-5.2 (December 2025 version) for proofreading suggestions. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

LLM	Large Language Model
U-DPO	Utility-weighted Direct Preference Optimization
RLHF	Reinforcement Learning from Human Feedback

Appendix A. Prompt Template

Appendix A.1

To enable reproducibility of our synthetic preference-pair construction pipeline, we provide the exact prompt template used to generate candidate texts. Since collecting human-labeled preference pairs is expensive, we construct them automatically by first generating multiple candidates per class and then scoring them with an auxiliary classifier.

For each class $y \in C$, we form a 2-shot prompt by randomly sampling two real training examples from class y and inserting them as in-context demonstrations. We then generate n candidate texts conditioned on these two examples.

Appendix A.2

Prompt Template: To control prompt length, each example is truncated to a maximum of 512 characters.

```
The following Examples are academic paper abstracts.
They are written in formal scientific style. The following Examples are academic paper abstracts.
Your task is to generate a new academic abstract in a similar style and topic.

Example 1:
{EXAMPLE_1_TRUNCATED}

Example 2:
{EXAMPLE_2_TRUNCATED}

Now, generate a single academic abstract paragraph in the same domain.
```

Figure A1. Prompt template used to generate candidate texts.

Appendix B. Utility Design and λ Ablation

U-DPO constructs automatic preference pairs using a task-specific utility that combines an internal text-quality signal and an external label-confidence signal.

Concretely, the internal utility $r_{int}(x)$ measures the intrinsic well-formedness of a generated text independent of the target label, implemented with a frozen MiniLM encoder and a lightweight head trained to predict a quality score in [0, 1].

The external utility $r_{ext}(x)$ is defined from an auxiliary classifier's target-vs-competitor margin and is min–max normalized over the candidate pool used for pair construction to ensure comparable scale.

$$u(x) = \lambda\, r_{int}(x) + (1 - \lambda)\, r_{ext}(x),\ 0 \le \lambda \le 1.$$

We combine the two signals into a single utility. Here, λ controls the trade-off: larger λ emphasizes intrinsic text quality, while smaller λ emphasizes label confidence.

$\lambda = 0$: utility reduces to external-only scoring, $u(x) = r_{ext}(x)$.

$\lambda = 1$: utility reduces to internal-only scoring, $u(x) = r_{int}(x)$.

To quantify the effect of the mixing weight and isolate the contribution of r_{int} and r_{ext}, we conduct a minimal sweep over:

$$\lambda \in \{0, 0.25, 0.5, 0.75, 1.0\}.$$

All other settings are kept identical to the main experiments (same prompts, candidate pool size n = 5, same auxiliary scorers, same DPO hyperparameters, and the same synthetic–real augmentation protocol).

We report downstream classification accuracy on arXiv, using the same evaluation procedure as Section 6.4.

We use $\lambda = 0.5$ as the default in the main paper, as it provides a balanced trade-off and strong performance across datasets.

Table A1. Ablation on λ: effect of mixing r_{ext} and r_{int} on arXiv accuracy.

λ	Utiltiy	arXiv ACC
0.0	r_{ext} only	0.7842
0.25	mix	0.7864
0.5	mix(default)	0.790
0.75	mix	0.7752
1.0	r_{int} only	0.7544

Appendix C. Generalizability Beyond Topic Classification

We acknowledge a limitation: we do not directly evaluate U-DPO on other task families such as sentiment analysis, intent detection, toxicity moderation, or broader instruction-following benchmarks. As a result, our empirical claims are currently bound to the class-conditional augmentation setting studied in the main paper.

In addition, our main experiments target scientific-document topic classification and use SciBERT as the downstream classifier. While this choice is well-motivated for scientific text, it limits the scope of our empirical validation: the observed gains may partially depend on domain-specific properties and on the particular inductive biases of SciBERT. We therefore view cross-task and cross-model generalization as an important limitation of the current study.

To address this concern, we include additional experiments beyond the main topic classification setting to provide evidence that U-DPO is not restricted to scientific domain classification.

Table A2 reports across-domain generalization results on SST-5 sentiment classification, where we evaluate downstream performance using BERT classifiers. Table A3 reports the across-backbone results by replacing SciBERT with BERT in the downstream classifier. The experimental setup is identical to Section 6.5 (same prompting, candidate pool size, utility computation, and training protocol), with only the target dataset/task or classifier backbone changed as specified above.

Table A2. Reports the cross-task generalization results on SST-5 (mean over 20 runs).

Model	Accuracy	Delta
$\text{BERT}_{\text{base}}$	0.538	-
U-DPO ($\lambda = 0.5$)	0.544	+0.006

Table A3. Cross-backbone results on arXiv topic classification by replacing SciBERT with BERT as the downstream classifier (mean over 20 runs).

Model	Accuracy	Delta
$\text{BERT}_{\text{base}}$	0.8672	-
$\text{DPO}_{\text{augmentation}}$	0.8676	+0.0004
U-DPO ($\lambda = 0.5$)	0.8756	+0.0084

References

1. Wang, Z.; Pang, Y.; Lin, Y.; Zhu, X. Adaptable and reliable text classification using large language models. *arXiv* **2024**, arXiv:2405.10523. [CrossRef]
2. Kostina, A.; Dikaiakos, M.D.; Stefanidis, D.; Pallis, G. Large language models for text classification: Case study and comprehensive review. *arXiv* **2025**, arXiv:2501.08457. [CrossRef]
3. Yoo, K.M.; Park, D.; Kang, J.; Lee, S.-W.; Park, W. GPT3Mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021. [CrossRef]

4. Kruschwitz, U.; Schmidhuber, M. LLM-based synthetic datasets: Applications and limitations in toxicity detection. In Proceedings of the Fourth Workshop on Threat, Aggression & Cyberbullying, Torino, Italy, 20 May 2024.
5. Li, Z.; Zhu, H.; Lu, Z.; Yin, M. Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023. [CrossRef]
6. Yamagishi, Y.; Nakamura, Y. UTRadNLP at #SMM4H 2024: Why LLM-generated texts fail to improve text classification models. In Proceedings of the 9th Social Media Mining for Health Research and Applications (SMM4H 2024) Workshop, Bangkok, Thailand, 15 August 2024.
7. Nadas, M.; Diosan, L.; Tomescu, A. Synthetic Data Generation Using Large Language Models: Advances in Text and Code. *arXiv* **2025**, arXiv:2503.14023. [CrossRef]
8. Gan, Z.; Liu, Y. Towards a theoretical understanding of synthetic data in llm post-training: A reverse-bottleneck perspective. *arXiv* **2025**, arXiv:2410.01720. [CrossRef]
9. Christiano, P.F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; Amodei, D. Deep reinforcement learning from human preferences. *arXiv* **2023**, arXiv:1706.03741. [CrossRef]
10. Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C.D.; Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv* **2024**, arXiv:2305.18290. [CrossRef]
11. Azar, M.G.; Guo, Z.D.; Piot, B.; Munos, R.; Rowland, M.; Valko, M.; Calandriello, D. A General Theoretical Paradigm to Understand Learning from Human Preferences. *arXiv* **2024**, arXiv:2310.12036. [CrossRef]
12. Ethayarajh, K.; Xu, W.; Muennighoff, N.; Jurafsky, D.; Kiela, D. KTO: Model Alignment as Prospect Theoretic Optimization. *arXiv* **2024**, arXiv:2402.01306. [CrossRef]
13. Wu, J.; Wang, X.; Yang, Z.; Wu, J.; Gao, J.; Ding, B.; Wang, X.; He, X. AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization. *arXiv* **2024**, arXiv:2410.10148. [CrossRef]
14. Wang, Z.; Pang, Y.; Lin, Y. Large language models are zero-shot text classifiers. *arXiv* **2023**, arXiv:2312.01044. [CrossRef]
15. Meshkin, H.; Zirkle, J.; Arabidarrehdor, G.; Chaturbedi, A.; Chakravartula, S.; Mann, J.; Thrasher, B.; Li, Z. Harnessing large language models' zero-shot and few-shot learning capabilities for regulatory research. *Brief. Bioinform.* **2024**, *25*, bbae354. [CrossRef] [PubMed]
16. Bucher, M.J.J.; Martini, M. Fine-tuned 'small' LLMs (still) significantly outperform zero-shot generative AI models in text classification. *arXiv* **2024**, arXiv:2406.08660. [CrossRef]
17. Ruan, Q.; Kuznetsov, I.; Gurevych, I. Are large language models good classifiers? A study on edit intent classification in scientific document revisions. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024. [CrossRef]
18. Vajjala, S.; Shimangaud, S. Text classification in the llm era—Where do we stand? *arXiv* **2025**, arXiv:2502.11830. [CrossRef]
19. Edwards, A.; Camacho-Collados, J. Language models for text classification: Is in-context learning enough? *arXiv* **2024**, arXiv:2403.17661. [CrossRef]
20. Tan, Z.; Li, D.; Wang, S.; Beigi, A.; Jiang, B.; Bhattacharjee, A.; Karami, M.; Li, J.; Cheng, L.; Liu, H. Large language models for data annotation and synthesis: A survey. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024. [CrossRef]
21. Choi, J.; Kim, Y.; Yu, S.; Yun, J.M.; Kim, Y.B. UniGen: Universal domain generalization for sentiment classification via zero-shot dataset generation. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024. [CrossRef]
22. Peng, L.; Wang, Z.; Shang, J. Incubating text classifiers following user instruction with nothing but LLM. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024. [CrossRef]
23. Zeng, L. Leveraging large language models for code-mixed data augmentation in sentiment analysis. In Proceedings of the Second Workshop on Social Influence in Conversations (SICon 2024), Miami, FL, USA, 15–16 November 2024. [CrossRef]
24. Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Papernot, N.; Anderson, R.; Gal, Y. AI models collapse when trained on recursively generated data. *Nature* **2024**, *631*, 755–759. [CrossRef]
25. Li, M.; Chen, H.; Wang, Y.; Zhu, T.; Zhang, W.; Zhu, K.; Wong, K.F.; Wang, J. Understanding and Mitigating the Bias Inheritance in LLM-based Data Augmentation. *arXiv* **2025**, arXiv:2502.04419. [CrossRef]
26. Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; Christiano, P.F. Learning to summarize from human feedback. *arXiv* **2022**, arXiv:2009.01325. [CrossRef]
27. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *arXiv* **2022**, arXiv:2203.02155. [CrossRef]
28. Kaufmann, T.; Weng, P.; Bengs, V.; Hüllermeier, E. A survey of reinforcement learning from human feedback. *arXiv* **2024**, arXiv:2312.14925. [CrossRef]

29. Amini, A.; Vieira, T.; Cotterell, R. Direct Preference Optimization with an Offset. In *Findings of the Association for Computational Linguistics*; ACL: Stroudsburg, PA, USA, 2024. [CrossRef]
30. Bradley, R.A.; Terry, M.E. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* **1952**, *39*, 324–345. [CrossRef]
31. Keles, F.D.; Wijewardena, P.M.; Hegde, C. On The Computational Complexity of Self-Attention. *arXiv* **2022**, arXiv:2209.04881. [CrossRef]
32. Wolfe, C.R. Writing an LLM from Scratch, Part 14—The Complexity of Self-Attention at Scale. Deep (Learning) Focus. 2025. Available online: https://www.gilesthomas.com/2025/05/llm-from-scratch-14-taking-stock-part-2-the-complexity-of-self-attention-at-scale (accessed on 27 January 2026).
33. Clement, C.B.; Bierbaum, M.; O'Keeffe, K.P.; Alemi, A.A. On the use of arxiv as a dataset. *arXiv* **2019**, arXiv:1905.00075. [CrossRef]
34. Kowsari, K.; Brown, D.E.; Heidarysafa, M.; Meimandi, K.J.; Gerber, M.S.; Barnes, L.E. HDLTex: Hierarchical deep learning for text classification. In Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017. [CrossRef]
35. Beltagy, I.; Lo, K.; Cohan, A. SciBERT: A Pretrained Language Model for Scientific Text. *arXiv* **2019**, arXiv:1903.10676. [CrossRef]
36. Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; Zhou, M. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-trained Transformers. *arXiv* **2020**, arXiv:2002.10957. [CrossRef]
37. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019. [CrossRef]
38. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692. [CrossRef]
39. Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The Llama 3 Herd of Models. *arXiv* **2024**, arXiv:2407.21783. [CrossRef]
40. Abouelenin, A.; Ashfaq, A.; Atkinson, A.; Awadalla, H.; Bach, N.; Bao, J.; Benhaim, A.; Cai, M.; Chaudhary, V.; Chen, C.; et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv* **2025**, arXiv:2503.01743. [CrossRef]
41. Casper, S.; Davies, X.; Shi, C.; Gilbert, T.K.; Scheurer, J.; Rando, J.; Freedman, R.; Korbak, T.; Lindner, D.; Freire, P.; et al. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback. *arXiv* **2023**, arXiv:2307.15217. [CrossRef]
42. Shi, W.; Yuan, M.; Wu, J.; Wang, Q.; Feng, F. Direct multi-turn preference optimization for language agents. *arXiv* **2025**, arXiv:2406.14868. [CrossRef]
43. Hu, Z.; Rostami, M.; Thomason, J. Multimodal Synthetic Data Finetuning and Model Collapse. *arXiv* **2025**, arXiv:2505.08803. [CrossRef]