WRitEer: A Multi-Objective, Preference-Driven Multi-Agent Framework for Human-Like Advanced Text Generation

Junchuan Yu[*1], **Yuyang Sun**[*2†]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
[2] College of Informatics, Huazhong Agricultural University, Wuhan, China
hogwartsniffler@163.com, frontsea040320@gmail.com

Abstract

Advanced text generation is paramount for enhancing the naturalness of human-computer interaction and improving emotional expressiveness. Current mainstream methods largely rely on large language models (LLMs) for single-turn generation, often lacking the interactivity and multi-dimensional feedback mechanisms inherent in human writing. This limitation frequently results in generated texts that fall short in terms of depth, fluency, and stylistic sophistication.

To address these deficiencies, this paper proposes WRitEer (Writer-Reader iterative tuning with Editor-Driven evolution and refinement), an interactive multi-agent collaborative human-like writing framework. Centered around an LLM, this framework integrates multi-objective optimization with preference fine-tuning techniques. It introduces three synergistic agents: the Reader, responsible for discourse analysis and indicator generation; the Editor, which constructs prompts based on feedback indicators and iteratively refines them through an evolutionary search; and the Writer, which generates text based on these refined prompts and continuously self-optimizes via a DPO mechanism that incorporates preference feedback. Experimental results consistently demonstrate that this "generate-evaluate-reflect-optimize" workflow significantly outperforms single LLM models across multiple datasets, yielding advanced rich texts that exhibit superior human-like style, coherence, expressiveness, and controllability.

Code — https://github.com/frontsea320/WRitEer

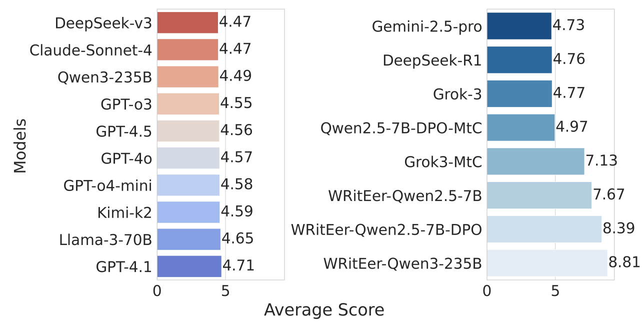

Figure 1: Overall performance comparison of all evaluated models.

Introduction

Text generation remains one of the central research topics in the development of large language models (LLMs), particularly in generating natural, fluent, and stylistically sophisticated texts comparable to those written by human authors. In recent years, significant progress has been made in this domain (Çalık and Akkuş 2025; Ma et al. 2025; Pico et al. 2024). However, despite their impressive generative capabilities, current general-purpose LLMs still face considerable limitations in advanced writing tasks. These shortcomings include producing rigid or unnatural text, conveying overly simplistic or repetitive stylistic patterns, and lacking nuanced sophistication. Traditional approaches to advanced text generation often rely on manually crafted stylistic rules or annotated labels, resulting in heavy dependence on human intervention and limited adaptability to diverse writing contexts (Li et al. 2023; Lee, Chen, and Huang 2010; Xie and Agrawal 2023).

With the rapid advancement of multi-agent systems and LLM technologies, collaborative multi-agent frameworks have emerged as a promising direction for enhancing the quality of advanced text generation (Park et al. 2023; Tran et al. 2025). Recent studies have highlighted several approaches in this domain. WritingPath, for instance, improves coherence by employing explicit outlines to guide LLMs (Lee et al. 2024b), yet its reliance on pre-defined structures restricts adaptability to varied stylistic and contextual requirements. The BERT-GPT-4 hybrid (Chen et al. 2024) achieves strong semantic depth and fluency through architectural integration, but it still depends heavily on manual design choices and lacks iterative refinement capabilities. In conversational AI, models such as LAPDOG (He et al. 2024) enrich sparse persona profiles with retrieval augmentation, producing more tailored responses; however, their effectiveness remains limited to specific domains. Overall, despite these advances, existing methods are not yet capable of automatically and systematically generating high-quality texts that balance stylistic accuracy, diversity, and naturalness.

[*] These authors contributed equally.
[†] Corresponding author.

To address the inherent complexity and challenges in advanced text generation, this paper proposes a novel multi-agent interactive framework, **WRitEer** (Writer-Reader iterative tuning with Editor-Driven evolution and refinement), which aims to systematically generate sophisticated and naturally fluent texts. Specifically, we formulate the advanced text generation task as a multi-objective optimization problem and introduce the concept of evolutionary search (Ye et al. 2024; Yuan et al. 2024; Akiba et al. 2025; Liu et al. 2024) to design specialized evolutionary operators, enabling the progressive approximation of Pareto-optimal solutions. Furthermore, during the optimization process, we iteratively construct preference data from generated texts and apply Direct Preference Optimization (DPO) (Rafailov et al. 2023; Park et al. 2025) to continually fine-tune the generative model via reinforcement learning, thereby facilitating the model's self-evolution and improving the precision and relevance of stylistic expression (Lu et al. 2024).

By integrating these techniques, **WRitEer** not only significantly simplifies the design process of advanced text generation systems but also fosters a continuous learning and adaptive environment. This enables the system to progressively enhance its performance over time, thereby improving the efficiency and effectiveness of advanced text generation.

The main contributions of this work are summarized as follows:

- We propose a novel formulation that models the advanced text generation task as a multi-objective optimization problem driven by dynamic evaluation metrics. An evolutionary search strategy is employed to efficiently approximate the Pareto-optimal solutions.
- We construct preference data from iteratively generated texts and, for the first time, introduce the Direct Preference Optimization (DPO) mechanism into the advanced text generation task. This significantly enhances the performance of the generation model, allowing smaller-scale models to achieve or even approach the performance of larger models.
- We design and implement an innovative multi-agent collaborative framework for advanced text generation, termed **WRitEer**, which orchestrates the interaction among three agents—the Reader, the Editor, and the Writer—to systematically optimize the generation process. Experimental results demonstrate that **WRitEer** outperforms mainstream large language models under our evaluation protocol, as illustrated in Figure 1.

Related Works

Limitations of LLM-based Single-turn Generation and the Rise of Multi-agent Collaboration

In recent years, the rapid advancement of large language models (LLMs) has driven the dominant paradigm of advanced text generation toward single-turn generation based on LLMs. Despite their impressive text generation capabilities, LLMs exhibit notable limitations in advanced writing tasks. They often lack the inherent interactivity and multi-dimensional feedback mechanisms characteristic of human writing, resulting in outputs that fall short in depth, fluency, and stylistic nuance. Specifically, LLMs may produce rigid or unnatural texts, convey oversimplified or repetitive stylistic patterns, and struggle with fine-grained control over context-dependent stylistic cues and subtle tonal variations (Di Palma et al. 2025; Zhang et al. 2023). Furthermore, LLMs typically face challenges in logical reasoning, long-term planning, and the handling of complex linguistic structures, which collectively hinder their ability to generate coherent and high-quality advanced texts (Qi et al. 2025; Yan et al. 2024).

To overcome the inherent limitations of single-agent LLMs, collaborative multi-agent frameworks have emerged as a promising direction for enhancing the quality of advanced text generation (Lützenberger et al. 2016; Park and Sugumaran 2005; Wei, Shan, and Li 2025). By leveraging the complementary strengths of diverse agents, these frameworks significantly improve the sophistication and coherence of generated content, enabling distributed intelligence, coordinated actions, and optimized decision-making that go beyond the capabilities of a single model (Luo et al. 2025; Amirkhani and Barshooi 2022; Fourney et al. 2024). For instance, in the domain of creative writing, multi-agent systems have been employed to facilitate divergent exploration, iterative refinement, and collaborative synthesis. By simulating the roles of authors, editors, and critics, such systems effectively enhance the creativity and complexity of generated texts (Lin et al. 2025; Zhang et al. 2024; Yan et al. 2025). This "generate–evaluate–reflect–optimize" workflow, enabled through agent collaboration, demonstrates strong potential for addressing the challenges of complex advanced text generation tasks.

WritingBench: A Comprehensive Benchmark for Generative Writing

Given the limitations of existing benchmarks in evaluating text generation by large language models (LLMs)—particularly in capturing the diverse requirements of high-quality written content—Wu et al. proposed **WritingBench** (Wu et al. 2025), a comprehensive evaluation platform designed to assess LLM performance across six core writing domains and over 100 subdomains, including creative, persuasive, informational, and technical writing.

At the heart of WritingBench lies a *query-dependent evaluation framework*. This framework leverages LLMs to dynamically generate five instance-specific evaluation criteria tailored to each writing task. These criteria adapt finely to diverse requirements related to style, format, length, and tone. A fine-tuned critic model is employed to assign scores and provide justifications based on the dynamically generated criteria. This method significantly enhances alignment with human judgments, achieving a human-model agreement rate of up to 83%, far surpassing traditional static baselines.

Moreover, WritingBench demonstrates powerful data curation capabilities, enabling the training of small-scale models with filtered data to achieve performance close to or even matching state-of-the-art (SOTA) models. In our work, a lightly modified version of WritingBench serves as the foundation for multi-objective optimization and the overall assessment of writing quality.

Direct Preference Optimization (DPO)

The core idea behind **Direct Preference Optimization (DPO)** is to analytically bridge the reward maximization objective and the optimal policy by reformulating the problem into a classification task over human preference data (Xu et al. 2024; Zhong et al. 2024; Lee et al. 2024a). This variable transformation enables the direct optimization of a language model to follow human preferences, without the need to explicitly fit a reward model or to perform sampling-based optimization from the model during fine-tuning.

It substantially lowers the barrier to incorporating human preferences into LLM training by eliminating the need for reward modeling and reinforcement learning optimization, resulting in greater stability, improved performance, and dramatically reduced computational cost. Empirical results show that DPO achieves comparable or superior performance to existing methods—including PPO-based RLHF—on tasks such as sentiment alignment, summarization, and single-turn dialogue. In our work, we adopt DPO to perform reverse fine-tuning of our generation model, enabling small-scale models to achieve performance competitive with that of larger models (Ouyang et al. 2022).

Method

Reader: WritingBench-based Metric Evaluation and Stylistic Perception

In the WRitEer framework, the core responsibility of the **Reader** module is to perform multi-dimensional advanced structure analysis on the input text T and generate quantitative metrics for use in subsequent stages (Wan et al. 2025; Liu et al. 2025). To this end, we incorporate **WritingBench**—a comprehensive benchmark designed for evaluating generative writing tasks. WritingBench employs a query-dependent evaluation mechanism that automatically generates five instance-specific evaluation criteria for each writing prompt. This approach has demonstrated over 80% human-model agreement across multiple experiments, significantly outperforming traditional static evaluation methods.

To integrate the evaluation results from WritingBench into the WRitEer generation pipeline, as illustrated in Figure 2, the Reader module maps its output into two types of structured metrics (Celikyilmaz, Clark, and Gao 2020): static metrics $\mathbf{S}$ and dynamic metrics $\mathbf{D}$, formally defined as:

$$(\mathbf{S}, \mathbf{D}) = \mathrm{WB}(T), \tag{1}$$

where

$$\mathbf{S} = \{s_1, s_2, \ldots, s_m\}, \qquad \mathbf{D} = \{d_1, d_2, \ldots, d_n\}. \tag{2}$$

The static metrics $\mathbf{S}$ capture high-level evaluations of the overall quality of the text and are directly used to assess the quality of generated outputs at each iteration. In contrast, the dynamic metrics $\mathbf{D}$ reflect performance fluctuations in specific micro-aspects of the text, serving as target references for the generation and evaluation prompts used in subsequent modules.

Through this dual-channel evaluation mechanism—*static + dynamic*—the Reader module not only achieves precise perception of stylistic expression but also establishes a clear and reliable data foundation for the Editor and Writer modules. This ensures that the entire **WRitEer** system can continuously improve text quality under the guidance of quantifiable evaluations.

Editor: Multi-objective Optimization and Evolutionary Operator Design Based on Dynamic Metrics

Within the **WRitEer** framework, we formulate the advanced text generation task as a multi-objective optimization problem guided by dynamic metrics. Specifically, given an input text T and its associated dynamic metric set $\mathbf{D}(T) = \{d_1(T), d_2(T), \ldots, d_n(T)\}$, the goal of the **Editor** module is to optimize the combination of prompts such that the static metric values are maximized while the dynamic metrics approach their desired target vector $\mathbf{D}^*$.

In this multi-objective optimization setting, we define the fitness function by combining both static and dynamic metrics. Specifically, for any candidate prompt combination p, the fitness is computed as follows:

$$\mathrm{fitness}(p) = \sum_{i=1}^{m} s_i(T_p) + \lambda \sum_{j=1}^{n} \mathrm{bonus}(d_j(T_p)), \tag{3}$$

where T_p represents the text generated by prompt combination p, and $\{s_1, s_2, \ldots, s_m\}$ are the static evaluation metrics obtained from T_p. The second term denotes the aggregated dynamic bonus, calculated across all dynamic metric dimensions $\{d_1, d_2, \ldots, d_n\}$, weighted by the hyperparameter λ, thus effectively guiding the search towards improved performance on both static quality and dynamic characteristics. The overall optimization task can thus be expressed as:

$$\max_{p \in \mathcal{P}} f(p), \tag{4}$$

while simultaneously encouraging $\mathbf{D}(p)$ to converge toward the ideal target vector $\mathbf{D}^*$ in each dynamic metric dimension.

Given the large and non-differentiable nature of the prompt combination space $\mathcal{P}$, we introduce a heuristic evolutionary search (Miikkulainen et al. 2024; Novikov et al. 2025) framework to explore the space efficiently. This framework includes the design of an initial population generation strategy and several evolutionary operators. First, based on the given text T and its dynamic metrics $\mathbf{D}(T)$, the Editor module uses a large language model (LLM) to construct an initial population:

$$P^{(0)} = \{p_1^{(0)}, p_2^{(0)}, \ldots, p_N^{(0)}\}, \tag{5}$$

where N is the predefined population size, and each individual $p_j^{(0)} = \{p_{j,1}, p_{j,2}, \ldots, p_{j,n}\}$ represents a full prompt combination consisting of n sub-prompts corresponding to the dimensions of the dynamic metrics. The initial population is generated without evaluation, guided purely by the

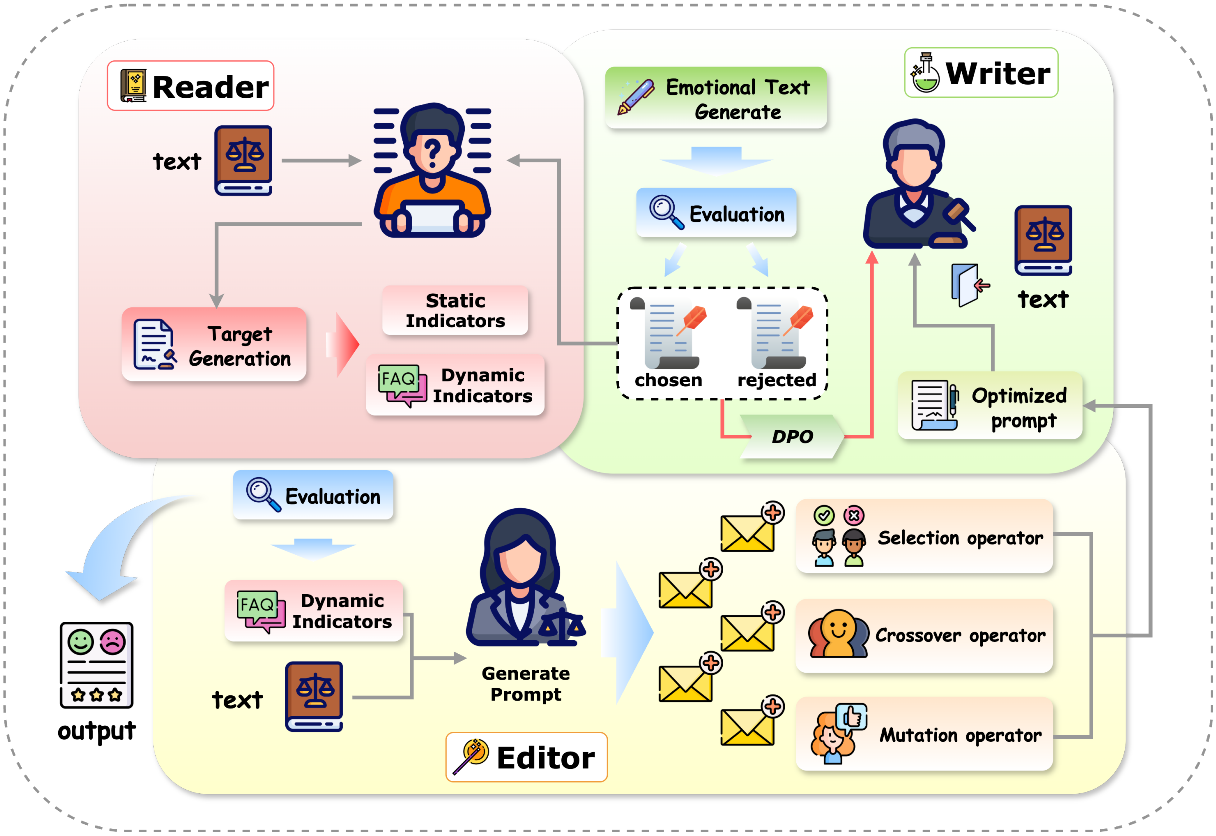

Figure 2: The architecture of the proposed **WRitEer** framework. It consists of three collaborative agents: the **Reader**, which evaluates input texts based on static and dynamic indicators; the **Editor**, which generates optimized prompts through evolutionary operators including selection, crossover, and mutation; and the **Writer**, which produces emotional text and is fine-tuned with Direct Preference Optimization (DPO). Together, these components form a generate–evaluate–optimize pipeline that enhances both the fluency and emotional alignment of generated texts.

semantic content of the dynamic metrics to ensure diversity for subsequent evolution.

In each iteration t ($t = 1, 2, 3$), a single evolutionary operator is executed. Except for the first round, before each iteration, the Editor evaluates the prompt combination $p^{(t-1)}$ using WritingBench:

$$\mathbf{S}(p^{(t-1)}) = \{s_1(p^{(t-1)}), \dots, s_m(p^{(t-1)})\}, \quad (6)$$

$$\mathbf{D}(p^{(t-1)}) = \{d_1(p^{(t-1)}), \dots, d_n(p^{(t-1)})\}. \quad (7)$$

Subsequently, the Editor executes one of the following evolutionary operators in a pre-defined order:

Selection Operator: Evaluate all individuals in $P^{(t-1)}$ using $f(p)$ and select the one with the highest fitness as the result of the current round:

$$p^{(t)} = \arg\max_{p_j^{(t-1)}} f(p_j^{(t-1)}). \quad (8)$$

Crossover Operator: Perform crossover between $p^{(t-1)}$ and the highest-fitness individual p_{sel} in the population. A subset of indices $I \subseteq \{1, \dots, n\}$ is randomly selected, and a new combination is constructed as:

$$p_i^{(t)} = \begin{cases} p_{\text{sel},i}, & i \in I, \\ p_i^{(t-1)}, & i \notin I. \end{cases} \quad (9)$$

Mutation Operator: Identify the y worst-performing dimensions in the dynamic metrics of $p^{(t-1)}$, forming index set $J \subseteq \{1, \dots, n\}$, and regenerate the corresponding sub-prompts using the LLM:

$$p_i^{(t)} = \begin{cases} \mathit{Regenerate}(p_i^{(t-1)}), & i \in J, \\ p_i^{(t-1)}, & i \notin J. \end{cases} \quad (10)$$

A total of three iterations are performed, each applying one evolutionary operator. After the third iteration, the optimized prompt combination is obtained:

$$p^* = p^{(t)}, \quad (11)$$

which serves as the output of the Editor module.

This optimized prompt is expected to improve the overall performance on static metrics while ensuring that each dimension of the dynamic metrics is closer to its ideal target. The resulting prompt combination is then passed to the

Writer module, guiding the final text generation process and ensuring that the output is stylistically accurate, fluent, and expressive.

Writer: Text Generation via Preference Construction and Direct Preference Optimization

As illustrated in Figure 2, the **Writer** module in the WRitEer framework is responsible for the final stage of text generation and optimization. Specifically, the Writer receives the optimized prompt combination p^* produced by the Editor module, along with the original input text T, and uses a language model to generate two candidate outputs, denoted as $T_{p^*}^{(1)}$ and $T_{p^*}^{(2)}$.

To determine which of the two texts is preferable, the system invokes WritingBench to evaluate their static metric scores. The overall fitness score for each candidate is computed as:

$$f(T_{p^*}^{(k)}) = \sum_{i=1}^{m} s_i\big(T_{p^*}^{(k)}\big), \quad k \in \{1, 2\}. \tag{12}$$

The text with the higher fitness score is selected as the final optimized output of the current iteration:

$$T_{\text{chosen}} = \arg\max_{T_{p^*}^{(k)},\, k\in\{1,2\}} f(T_{p^*}^{(k)}), \tag{13}$$

and is subsequently passed back to the Reader module as the input for the next optimization round.

To further enhance the model's capability in advanced text generation, we integrate the **Direct Preference Optimization (DPO)** technique to fine-tune the base language model. In each optimization round, the Writer constructs a preference-based training triple $(p^*, T_{\text{chosen}}, T_{\text{rejected}})$, where:

$$T_{\text{rejected}} = \arg\min_{T_{p^*}^{(k)},\, k\in\{1,2\}} f(T_{p^*}^{(k)}). \tag{14}$$

Each input text undergoes three optimization rounds, corresponding to the application of the selection, crossover, and mutation operators in the Editor module. In each round, the Writer generates two candidate outputs, resulting in three preference triples:

$$\left\{ \left(p^{*(t)},\, T_{\text{chosen}}^{(t)},\, T_{\text{rejected}}^{(t)} \right) \right\}_{t=1}^{3}. \tag{15}$$

Once approximately x input texts have been processed, yielding about $3x$ preference triples, the DPO fine-tuning process is triggered. During DPO training, we define the conditional generation policy as $\pi_\theta(T \mid p)$, where θ denotes the parameters of the base language model. Given each preference triple, we define the following loss function to guide the optimization:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(p, T_{\text{chosen}}, T_{\text{rejected}})\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(T_{\text{chosen}} \mid p)}{\pi_\theta(T_{\text{rejected}} \mid p)} \right) \right]. \tag{16}$$

where $\mathcal{D}$ is the accumulated dataset of preference triples, $\sigma(\cdot)$ is the sigmoid function, and β is a hyperparameter controlling preference strength.

By minimizing this loss, the model is encouraged to generate texts that align more closely with human preferences. Each DPO fine-tuning phase is executed once for every collection of $3x$ new preference triples. Through iterative training, the base language model continuously improves its performance in stylistic alignment, linguistic fluency, and expressive power.

Ultimately, with the support of optimized prompts and progressively fine-tuned models, the Writer module produces outputs that more accurately reflect human emotional preferences and expressive intent.

Experiments

Datasets

To comprehensively evaluate the performance of the **WRitEer** framework, we selected two representative datasets covering both news summarization and narrative text generation tasks. For the news summarization datasets, we randomly sampled **3,000 non-overlapping examples** per trial and repeated the experiments **five times** to ensure stability. For the narrative dataset, we conducted experiments on the **entire corpus**, also repeated **five times** for reliability. The datasets are as follows:

- **CNN/DailyMail (Nallapati et al. 2016)**
  This dataset contains over 300,000 news articles paired with human-written summaries. It is widely used in abstractive summarization tasks and serves to evaluate the model's ability to generate coherent, informative, and stylistically natural summaries.
- **Harry Potter Custom Dataset**
  To assess the effectiveness of WRitEer in emotionally expressive and narrative-rich text generation, we constructed a custom dataset based on the *Harry Potter* series. We replaced key character and entity names (e.g., Ron $\rightarrow$ Rick, Harry $\rightarrow$ Henry) to prevent models from leveraging memorized knowledge. A total of **1,000 long-form excerpts** (including dialogues) were selected and manually abbreviated under fixed rules to remove explicit emotional markers, creating emotion-neutral baselines. The exact construction of the dataset can be found in the Appendix.

Implementation Details

In our experiments, we primarily adopted **Qwen2.5-7B** (Hui et al. 2024) as the backbone model for the **Writer** agent. Following the official recommendations of WritingBench, we employed **Claude-Sonnet-4** as the critic model, which was deployed in both the **Reader** agent and the WritingBench evaluation module. In addition, we provided an option to replace the Writer agent model with a local API deployment, allowing flexible adaptation to different hardware environments.

For comparative evaluation, we categorized the tested methods into three groups: **Direct Answer**, **Multi-turn Conversation**, and our proposed **WRitEer** framework. The inclusion of multi-turn conversation methods ensures a fairer

Method Type	Method	JS&R	LP&G	LR&P	Average	Std
Direct Answer	*Claude-Sonnet-4* (Anthropic 2025)	3.22	6.02	4.18	4.47	1.53
	GPT-4.1 (OpenAI 2025)	3.32	6.66	4.16	4.71	1.43
	GPT-4.5 (OpenAI 2024a)	3.30	6.22	4.16	4.56	1.47
	GPT-4o (OpenAI 2024b)	3.20	6.30	4.22	4.57	1.46
	GPT-o3 (OpenAI 2024d)	3.28	6.26	4.10	4.55	1.65
	GPT-o4-mini (OpenAI 2024c)	3.24	6.38	4.12	4.58	1.49
	Gemini-2.5-pro (DeepMind 2025)	3.34	6.64	4.20	4.73	1.37
	Grok-3 (xAI 2025)	3.40	6.64	4.26	4.77	1.41
	Llama-3-70B (Meta 2024)	3.32	6.42	4.22	4.65	1.56
	Qwen3-235B (Yang et al. 2025)	3.26	6.18	4.04	4.49	1.55
	DeepSeek-R1 (DeepSeek-AI 2025)	3.28	6.80	4.20	4.76	1.38
	DeepSeek-v3 (DeepSeek-AI 2024)	3.16	6.10	4.14	4.47	1.60
	Kimi-k2 (Team et al. 2025)	3.30	6.28	4.20	4.59	1.50
Multi-turn Conversation	*Grok3-Multi-Conversation*	6.88	7.88	6.64	7.13	1.30
	Qwen2.5-7B-DPO-Multi-Conversation	3.32	7.34	4.24	4.97	0.94
WRitEer	*WRitEer-Qwen2.5-7B*	8.00	8.52	6.48	7.67	1.08
	WRitEer-Qwen2.5-7B-DPO	8.14	8.92	8.12	8.39	**0.38**
	WRitEer-Qwen3-235B	**8.80**	**9.10**	**8.54**	**8.81**	0.40

Table 1. Performance of different methods across three evaluation metrics: JS&R (*Journalistic Style & Readability*), LP&G (*Language Precision & Grammar*), and LR&P (*Lexical Richness & Precision*). More detailed analyses (Zhang et al. 2019) and additional results are provided in the appendix.

comparison, as they better reflect interactive writing scenarios where the system can iteratively refine outputs through dialogue-like exchanges. Specific implementation details can be found in the appendix.

Hyperparameter Settings

- **Optimization Rounds**: Each input text underwent **3 optimization rounds**, sequentially applying the selection, crossover, and mutation operators.
- **Population Size**: **5 candidate prompt combinations** were generated in each round.
- **Fitness Function**: The weighting parameter was empirically set to $\lambda = 0.3$, as this value demonstrated the best balance between static quality scores and dynamic bonus contributions in preliminary trials.
- **DPO Fine-tuning**: Direct Preference Optimization was triggered every time approximately 300 preference triples were collected, with the preference strength coefficient set to $\beta = 0.1$, a value chosen based on empirical evidence indicating superior performance in advanced alignment.

Prompt Engineering Task-specific prompt templates were designed for each agent, guaranteeing consistency in style and appropriateness of stylistic expression throughout the optimization process. Specific templates for the prompt words can be found in the appendix.

Experimental Environment All experiments were conducted on a multi-GPU server equipped with **NVIDIA A800 GPUs**, using **PyTorch 2.5.1** and **CUDA 12.4**. Pre-trained model weights were obtained from HuggingFace and further fine-tuned within the **WRitEer** multi-agent pipeline.

Results and Analysis

To provide a clear overview of the evaluation outcomes, we first present in Table **??** the average scores of all models across the three key static evaluation metrics, covering the CNN/DailyMail dataset. The results for the Harry Potter Custom Dataset dataset are reported separately in the appendix.

Overall Model Performance Comparison The overall analysis reveals substantial performance differences among the evaluated approaches. Within the **Direct Answer** category, most models achieved moderate scores, with averages ranging between 4.47 and 4.77, and relatively high standard deviations of around 1.4–1.6, indicating noticeable variability in output quality. In contrast, **Multi-turn Conversation** methods demonstrated stronger performance: Grok3-Multi-Conversation attained an average score of 7.13, significantly outperforming all Direct Answer models, while Qwen2.5-7B-DPO-Multi-Conversation also achieved competitive results with enhanced stability (Std = 0.94). Most notably, the **WRitEer Framework** consistently outperformed both Direct Answer and Multi-turn Conversation baselines. WRitEer-Qwen2.5-7B obtained an average score of 7.67, while its DPO-enhanced variant further improved both accuracy and stability, reaching 8.39 with a remarkably low standard deviation of 0.38. The best performance was delivered by WRitEer-Qwen3-235B, which achieved an average of 8.81 across all metrics, setting a new benchmark for high-quality and stable text generation. These results collectively underscore the effectiveness of WRitEer in elevating both the overall quality and the consistency of generated texts, suggesting that the integration of evolutionary search

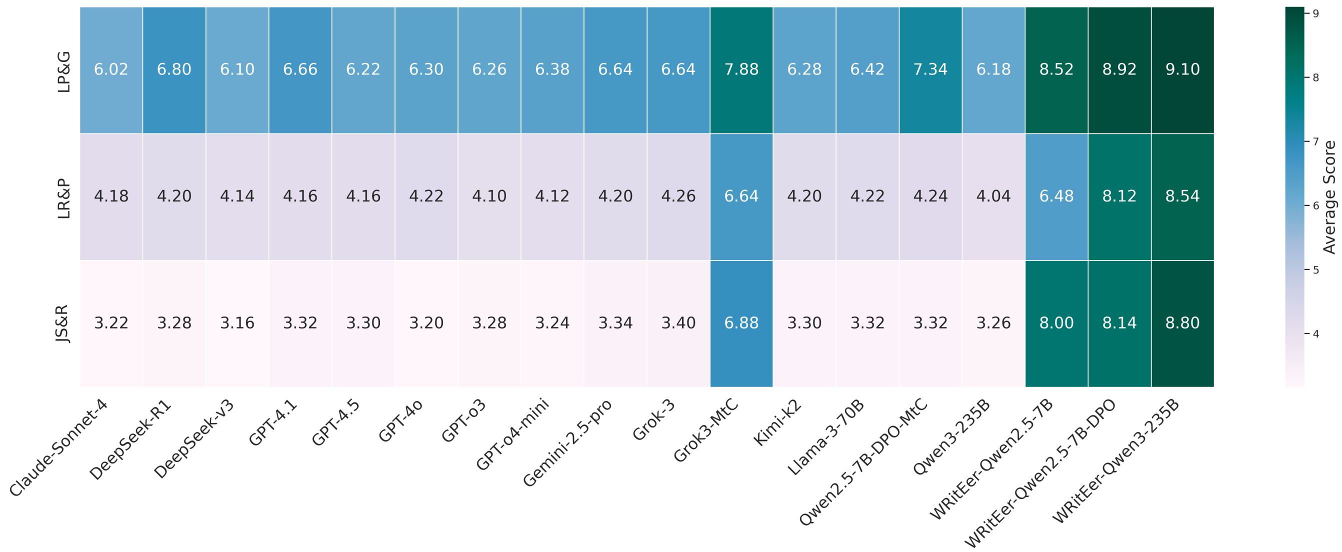

Figure 3: The heatmap reveals consistent performance differences among Direct Answer, Multi-turn Conversation, and WRitEer-based models, with WRitEer demonstrating the strongest overall results across all metrics. The models (from left to right) are:Claude-Sonnet-4, DeepSeek-R1, DeepSeek-v3, GPT-4.1, GPT-4.5, GPT-4o, GPT-o3, GPT-o4-mini, Gemini-2.5-pro, Grok-3, Grok3-MtC, Kimi-k2, Llama-3-70B, Qwen2.5-7B-DPO-MtC, Qwen3-235B, WRitEer-Qwen2.5-7B, WRitEer-Qwen2.5-7B-DPO, WRitEer-Qwen3-235B

and preference-based optimization provides significant advantages over conventional approaches.

Model Comparison Across Metrics The heatmap analysis (Figure 3) provides a detailed comparison of model performance across the three key evaluation metrics, revealing substantial variation among different approaches. In terms of *JS&R*, most baseline models scored relatively low, reflecting the persistent difficulty of emulating professional journalistic style. While Direct Answer models such as GPT-4.5, GPT-4.1, and Gemini-2.5-pro remained in the range of approximately 3.2–3.4, multi-turn conversation approaches showed notable improvement, with Grok3-Multi-Conversation reaching 6.88. The most remarkable results, however, were achieved by the WRitEer framework, where WRitEer-Qwen2.5-7B and WRitEer-Qwen2.5-7B-DPO scored 8.00 and 8.14 respectively, and WRitEer-Qwen3-235B attained the highest score of 8.80, indicating a near-human level of stylistic fluency and readability.

For *LP&G*, the performance gap was even more pronounced. While strong Direct Answer models such as DeepSeek-R1 and GPT-4.5 achieved scores around 6.1–6.8, the WRitEer framework delivered a substantial leap forward: WRitEer-Qwen2.5-7B reached 8.52, WRitEer-Qwen2.5-7B-DPO further improved to 8.92, and WRitEer-Qwen3-235B set the benchmark with a score of 9.10. These results highlight WRitEer's ability to ensure both grammatical correctness and high linguistic precision.

In the dimension of *LR&P*, scores among Direct Answer models clustered between 4.0 and 4.3, reflecting limited vocabulary diversity. Grok3-Multi-Conversation demonstrated a notable advantage with 6.64, but it was again the WRitEer framework that clearly outperformed all baselines. WRitEer-Qwen2.5-7B achieved 6.48, while WRitEer-Qwen2.5-7B-DPO and WRitEer-Qwen3-235B elevated performance to 8.12 and 8.54 respectively.

Conclusion

We introduce WRitEer, a multi-agent framework that advances text generation by integrating multi-objective optimization, evolutionary search, and preference-based finetuning. Experiments on CNN/DailyMail and the custom Harry Potter datasets (see Appendix) show WRitEer consistently surpasses mainstream large language models in both linguistic precision and stylistic quality.

The combination of evolutionary search with Direct Preference Optimization enables progressive refinement of outputs, better aligning them with human-like stylistic and emotional preferences. Its modular design supports adaptation to varied datasets and backbone models, enhancing robustness.

Future work will explore multi-turn interactive writing, incorporate multimodal signals (e.g., images and audio) (Alayrac et al. 2022), and pursue lightweight deployment to lower computational cost (Cui et al. 2024; Wu et al. 2023; Yin et al. 2024).

References

Akiba, T.; Shing, M.; Tang, Y.; Sun, Q.; and Ha, D. 2025. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7(2): 195–204.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Amirkhani, A.; and Barshooi, A. H. 2022. Consensus in multi-agent systems: a review. *Artificial Intelligence Review*, 55(5): 3897–3935.

Anthropic. 2025. Claude Sonnet. https://www.anthropic.com/claude/sonnet. Accessed: 2025-03-24.

Çalık, E. Y.; and Akkuş, T. R. 2025. Enhancing human-like responses in large language models. *arXiv preprint arXiv:2501.05032*.

Celikyilmaz, A.; Clark, E.; and Gao, J. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

Chen, J.; Wang, S.; Qi, Z.; Zhang, Z.; Wang, C.; and Zheng, H. 2024. A Combined Encoder and Transformer Approach for Coherent and High-Quality Text Generation. In *2024 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, 536–541. IEEE.

Cui, C.; Ma, Y.; Cao, X.; Ye, W.; Zhou, Y.; Liang, K.; Chen, J.; Lu, J.; Yang, Z.; Liao, K.-D.; et al. 2024. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 958–979.

DeepMind, G. 2025. Flash Thinking: Behind the Scenes of Gemini. https://deepmind.google/technologies/gemini/flash-thinking/. Accessed: 2025-03-24.

DeepSeek-AI. 2024. DeepSeek-V3 Technical Report. *ArXiv*, abs/2412.19437.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *ArXiv*, abs/2501.12948.

Di Palma, D.; De Bellis, A.; Servedio, G.; Anelli, V. W.; Narducci, F.; and Di Noia, T. 2025. LLaMAs Have Feelings Too: Unveiling Sentiment and Emotion Representations in LLaMA Models Through Probing. *arXiv preprint arXiv:2505.16491*.

Fourney, A.; Bansal, G.; Mozannar, H.; Tan, C.; Salinas, E.; Niedtner, F.; Proebsting, G.; Bassman, G.; Gerrits, J.; Alber, J.; et al. 2024. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*.

He, X.; Tian, Y.; Sun, Y.; Chawla, N.; Laurent, T.; LeCun, Y.; Bresson, X.; and Hooi, B. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37: 132876–132907.

Hui, B.; Yang, J.; Cui, Z.; Yang, J.; Liu, D.; Zhang, L.; Liu, T.; Zhang, J.; Yu, B.; Dang, K.; Yang, A.; Men, R.; Huang, F.; Quan, S.; Ren, X.; Ren, X.; Zhou, J.; and Lin, J. 2024. Qwen2.5-Coder Technical Report. *ArXiv*, abs/2409.12186.

Lee, A.; Bai, X.; Pres, I.; Wattenberg, M.; Kummerfeld, J. K.; and Mihalcea, R. 2024a. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity. *arXiv preprint arXiv:2401.01967*.

Lee, S. Y. M.; Chen, Y.; and Huang, C.-R. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, 45–53.

Lee, Y.; Ka, S.; Son, B.; Kang, P.; and Kang, J. 2024b. Navigating the path of writing: Outline-guided text generation with large language models. *arXiv preprint arXiv:2404.13919*.

Li, C.; Wang, J.; Zhang, Y.; Zhu, K.; Wang, X.; Hou, W.; Lian, J.; Luo, F.; Yang, Q.; and Xie, X. 2023. The good, the bad, and why: Unveiling emotions in generative ai. *arXiv preprint arXiv:2312.11111*.

Lin, Y.-C.; Chen, K.-C.; Li, Z.-Y.; Wu, T.-H.; Wu, T.-H.; Chen, K.-Y.; Lee, H.-y.; and Chen, Y.-N. 2025. Creativity in LLM-based Multi-Agent Systems: A Survey. *arXiv preprint arXiv:2505.21116*.

Liu, B.; Li, X.; Zhang, J.; Wang, J.; He, T.; Hong, S.; Liu, H.; Zhang, S.; Song, K.; Zhu, K.; et al. 2025. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*.

Liu, F.; Tong, X.; Yuan, M.; Lin, X.; Luo, F.; Wang, Z.; Lu, Z.; and Zhang, Q. 2024. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. *arXiv preprint arXiv:2401.02051*.

Lu, S.; Shao, J.; Luo, B.; and Lin, T. 2024. Morphagent: Empowering agents through self-evolving profiles and decentralized collaboration. *arXiv preprint arXiv:2410.15048*.

Luo, J.; Zhang, W.; Yuan, Y.; Zhao, Y.; Yang, J.; Gu, Y.; Wu, B.; Chen, B.; Qiao, Z.; Long, Q.; et al. 2025. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*.

Lützenberger, M.; Küster, T.; Masuch, N.; and Fähndrich, J. 2016. Multi-Agent System in Practice: When Research Meets Reality. In *AAMAS*, 796–805.

Ma, F.; Xie, Y.; Li, Y.; He, Y.; Zhang, Y.; Ren, H.; Liu, Z.; Yao, W.; Ren, F.; Yu, F. R.; et al. 2025. A review of human emotion synthesis based on generative technology. *IEEE Transactions on Affective Computing*.

Meta. 2024. The Llama 3 Herd of Models. *ArXiv*, abs/2407.21783.

Miikkulainen, R.; Liang, J.; Meyerson, E.; Rawal, A.; Fink, D.; Francon, O.; Raju, B.; Shahrzad, H.; Navruzyan, A.; Duffy, N.; et al. 2024. Evolving deep neural networks. In *Artificial intelligence in the age of neural networks and brain computing*, 269–287. Elsevier.

Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B.; et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Novikov, A.; Vũ, N.; Eisenberger, M.; Dupont, E.; Huang, P.-S.; Wagner, A. Z.; Shirobokov, S.; Kozlovskii, B.; Ruiz, F. J.; Mehrabian, A.; et al. 2025. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*.

OpenAI. 2024a. GPT-4.5. https://platform.openai.com. Large language model. Accessed via OpenAI API.

OpenAI. 2024b. GPT-4o System Card. *ArXiv*, abs/2410.21276.

OpenAI. 2024c. GPT-o4-mini. https://platform.openai.com/models/gpt-o4-mini. Compact multimodal large language model; text+ision via OpenAI API. Accessed via OpenAI API.

OpenAI. 2024d. OpenAI o3 System Card. arXiv:2412.16720.

OpenAI. 2025. GPT-4.1. https://platform.openai.com. Large language model. Accessed via OpenAI API.

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

Park, C.; Han, S.; Guo, X.; Ozdaglar, A.; Zhang, K.; and Kim, J.-K. 2025. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. *arXiv preprint arXiv:2502.18439*.

Park, J. S.; O'Brien, J.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, 1–22.

Park, S.; and Sugumaran, V. 2005. Designing multi-agent systems: a framework and application. *Expert Systems with Applications*, 28(2): 259–271.

Pico, A.; Vivancos, E.; Garcia-Fornes, A.; and Botti, V. J. 2024. Exploring Text-Generating Large Language Models (LLMs) for Emotion Recognition in Affective Intelligent Agents. In *ICAART (1)*, 491–498.

Qi, C.; Ma, R.; Li, B.; Du, H.; Hui, B.; Wu, J.; Laili, Y.; and He, C. 2025. Large language models meet symbolic provers for logical reasoning evaluation. *arXiv preprint arXiv:2502.06563*.

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

Team, K.; Bai, Y.; Bao, Y.; Chen, G.; Chen, J.; Chen, N.; Chen, R.; Chen, Y.; Chen, Y.; Chen, Y.; et al. 2025. Kimi K2: Open Agentic Intelligence. *arXiv preprint arXiv:2507.20534*.

Tran, K.-T.; Dao, D.; Nguyen, M.-D.; Pham, Q.-V.; O'Sullivan, B.; and Nguyen, H. D. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*.

Wan, D.; Chen, J. C.-Y.; Stengel-Eskin, E.; and Bansal, M. 2025. MAMM-refine: A recipe for improving faithfulness in generation with multi-agent collaboration. *arXiv preprint arXiv:2503.15272*.

Wei, Y.; Shan, X.; and Li, J. 2025. LERO: LLM-driven Evolutionary framework with Hybrid Rewards and Enhanced Observation for Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2503.21807*.

Wu, J.; Gan, W.; Chen, Z.; Wan, S.; and Yu, P. S. 2023. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, 2247–2256. IEEE.

Wu, Y.; Mei, J.; Yan, M.; Li, C.; Lai, S.; Ren, Y.; Wang, Z.; Zhang, J.; Wu, M.; Jin, Q.; et al. 2025. Writingbench: A comprehensive benchmark for generative writing. *arXiv preprint arXiv:2503.05244*.

xAI. 2025. Grok-3 and the Next Phase of xAI. https://x.ai/news/grok-3. Accessed: 2025-03-24.

Xie, J. J.; and Agrawal, A. 2023. Emotion and sentiment guided paraphrasing. *arXiv preprint arXiv:2306.05556*.

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

Yan, B.; Zhou, Z.; Zhang, L.; Zhang, L.; Zhou, Z.; Miao, D.; Li, Z.; Li, C.; and Zhang, X. 2025. Beyond self-talk: A communication-centric survey of llm-based multi-agent systems. *arXiv preprint arXiv:2502.14321*.

Yan, J.; Wang, C.; Huang, J.; and Zhang, W. 2024. Do large language models understand logic or just mimick context? *arXiv preprint arXiv:2402.12091*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Ye, H.; Wang, J.; Cao, Z.; Berto, F.; Hua, C.; Kim, H.; Park, J.; and Song, G. 2024. Reevo: Large language models as hyper-heuristics with reflective evolution. *Advances in neural information processing systems*, 37: 43571–43608.

Yin, S.; Fu, C.; Zhao, S.; Li, K.; Sun, X.; Xu, T.; and Chen, E. 2024. A survey on multimodal large language models. *National Science Review*, 11(12): nwae403.

Yuan, S.; Song, K.; Chen, J.; Tan, X.; Li, D.; and Yang, D. 2024. Evoagent: Towards automatic multi-agent generation via evolutionary algorithms. *arXiv preprint arXiv:2406.14228*.

Zhang, H.; Song, H.; Li, S.; Zhou, M.; and Song, D. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3): 1–37.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhang, Z.; Zhang, Y.; Li, L.; Gao, H.; Wang, L.; Lu, H.; Zhao, F.; Qiao, Y.; and Shao, J. 2024. Psysafe: A comprehensive framework for psychological-based attack, defense, and evaluation of multi-agent system safety. *arXiv preprint arXiv:2401.11880*.

Zhong, H.; Shan, Z.; Feng, G.; Xiong, W.; Cheng, X.; Zhao, L.; He, D.; Bian, J.; and Wang, L. 2024. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*.