SuperWriter: Reflection-Driven Long-Form Generation with Large Language Models

Anonymous ACL submission

Abstract

Long-form text generation remains a significant challenge for large language models (LLMs), particularly in maintaining coherence, ensuring logical consistency, and preserving text quality as sequence length increases. To address these limitations, we propose *SuperWriter*-Agent, an agent-based framework designed to enhance the quality and consistency of long-form text generation. *SuperWriter*-Agent introduces explicit structured thinking-through planning and refinement stages—into the generation pipeline, guiding the model to follow a more deliberate and cognitively grounded process akin to that of a professional writer. Based on this framework, we construct a supervised fine-tuning dataset to train a 7B *SuperWriter*-LM. We further develop a hierarchical Direct Preference Optimization (DPO) procedure that uses Monte Carlo Tree Search (MCTS) to propagate final quality assessments and optimize each generation step accordingly. Empirical results across diverse benchmarks demonstrate that *SuperWriter*-LM achieves state-of-the-art performance, surpassing even larger-scale baseline models in both automatic evaluation and human evaluation. Furthermore, comprehensive ablation studies demonstrate the effectiveness of hierarchical DPO and underscore the value of incorporating structured thinking steps to improve the quality of long-form text generation.

1 Introduction

Effective long-form text generation remains a fundamental challenge for advancing large language models (LLMs) (Yao et al., 2019; Schmidgall et al., 2025; Wang et al., 2024c). Unlike short-form generation tasks, where LLMs have demonstrated remarkable success, generating extended sequences often leadss to degraded coherence and logical consistency as the length grows.

Previous studies on long-form writing (Bai et al., 2024b; Pham et al., 2024) typically employ LLMs to generate text in a single pass without structured intermediate thinking or explicit planning. Consequently, the generated texts often appear plausible initially but lack sustained coherence and can exhibit logical contradictions over longer spans (Goldfarb-Tarrant et al., 2019; Hua and Wang, 2020; Rasheed et al., 2025; Minaee et al., 2024; Wu et al., 2025a).

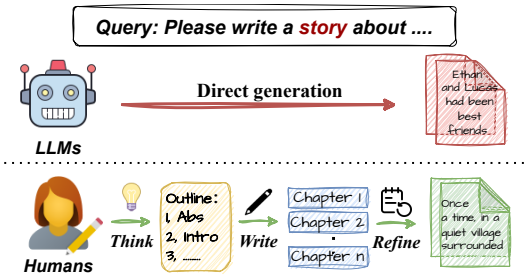

Figure 1: Current LLMs directly generate long text in a single pass, while human writers follow an iterative process of *thinking*, *outlining*, *writing*, and *refining* to ensure coherence and quality.

In practice, human writers rarely produce complex or lengthy documents in a single uninterrupted attempt. Instead, they commonly utilize iterative processes, including outlining, drafting, reviewing, and revising their writing to maintain coherence and logical consistency (Flower, 1981; Becker, 2006; Gollins and Gentner, 2016), as shown in Fig 1. Moreover, such writing processes are often accompanied by extensive cognitive activities, such as planning, foreshadowing, and strategic structuring—especially in tasks like writing detective novels or academic papers (Flower, 1981). Building on this observation, we propose *SuperWriter*-agent, a novel agent pipeline explicitly designed to embed structured thinking paradigms within long-form text generation. By incorporating explicit intermediate cognitive steps, *SuperWriter*-agent simulates the human writing process through a coordinated

agent-based framework.

Formally, *SuperWriter*-Agent generates training data through a structured three-stage framework: Planning → Writing → Refining. In the Planning stage, two agents collaboratively reflect and outline key arguments, decompose complex ideas, and establish logical connections. During the Writing stage, each paragraph is composed based on the structured plan, incorporating explicit thinking steps into the content. In the Refining stage, the text is carefully reviewed and revised to ensure clarity and structural integrity. Unlike previous SFT datasets for long-form generation (Bai et al., 2024b; Pham et al., 2024), our approach explicitly segments the data into these three stages, each enriched with thinking-oriented supervision. To further guide the *SuperWriter*-LM's writing capabilities, we apply hierarchical Direct Preference Optimization by comparing pairs of finalized response (Rafailov et al., 2023) and propagate the feedback to each step through a Monte Carlo Tree Search (MCTS). Through the integration of structured thinking signals and hierarchical feedback, *SuperWriter*-LM demonstrates notable improvements in fluency, coherence, and overall textual quality.

Finally, we validate the effectiveness of our trained *SuperWriter*-LM through both human and automatic evaluations on the WritingBench benchmark (Wu et al., 2025b). Extensive ablation studies further underscore the critical role of structured thinking data and hierarchical Direct Preference Optimization (DPO) in enabling high-quality text generation. Our contributions are summarized as follows:

- We introduce *SuperWriter*-agent, an agent-based framework with a structured Planning–Writing–Refining workflow that simulates human writing processes by injecting intermediate thinking steps into long-form text generation.
- We construct a thinking-supervised, stage-segmented dataset and train the *SuperWriter*-LM on it, then apply hierarchical DPO over final output comparisons to optimize each generation step.
- We empirically demonstrate the effectiveness of our approach, showing substantial improvements in fluency, coherence, and logical consistency compared to existing methods.

2 Related Work

Long-context Language Models Recent research on long-context language models has primarily focused on extending the input context length, enabling models to process substantially longer inputs. These efforts fall into two categories: zero-shot methods that expand the model's receptive field without further training (Han et al., 2023; Xiao et al., 2023; Zhang et al., 2024; Jin et al., 2024; An et al., 2024), and fine-tuning approaches that train models on longer sequences to improve long-context handling (Chen et al., 2023a; Peng et al., 2023; Xiong et al., 2024; Chen et al., 2023b; Bai et al., 2024a; Fu et al., 2024). However, most work has focused on input expansion, neglecting the equally important output capabilities. Empirical studies by Bai et al. (2024b) and Quan et al. (2024) reveal a significant gap between input length (over 100K tokens) and output length (approximately 2K words).

Long-form Text Generation Recent work highlights architectural innovations and specialized training paradigms (Salemi et al., 2025a; Que et al., 2024; Liu et al., 2023; Li et al., 2023). Re3 employs recursive prompting for coherent extended storytelling (Yang et al., 2022), while DOC (Yang et al., 2023) and hierarchical outlining (Wang et al., 2024b) use structured decomposition to improve narrative structure. Personalization is emphasized by models like LongLaMP (Kumar et al., 2024) and reasoning-enhanced self-training (Salemi et al., 2025b), which tailor outputs to user preferences. Long-form QA approaches further address complex queries with detailed responses (Dasigi et al., 2021; Stelmakh et al., 2022; Lee et al., 2023; Tan et al., 2024). Yet, challenges remain, including inconsistent coherence and difficulty generating high-quality large-scale text (Wu et al., 2024; Que et al., 2024). To address these, we propose a theoretical agent framework for long-form writing with new training strategies to enhance coherence and extended text generation.

3 *SuperWriter*-Agent

Current LLM training corpora offer abundant supervision for intermediate "*thinking*" in tasks like math reasoning and code generation (DeepSeek-AI et al., 2025a), but contain very little such data for writing (Soboleva et al., 2023). Most pretraining data (Parmar et al., 2024; Weber et al., 2024) con-

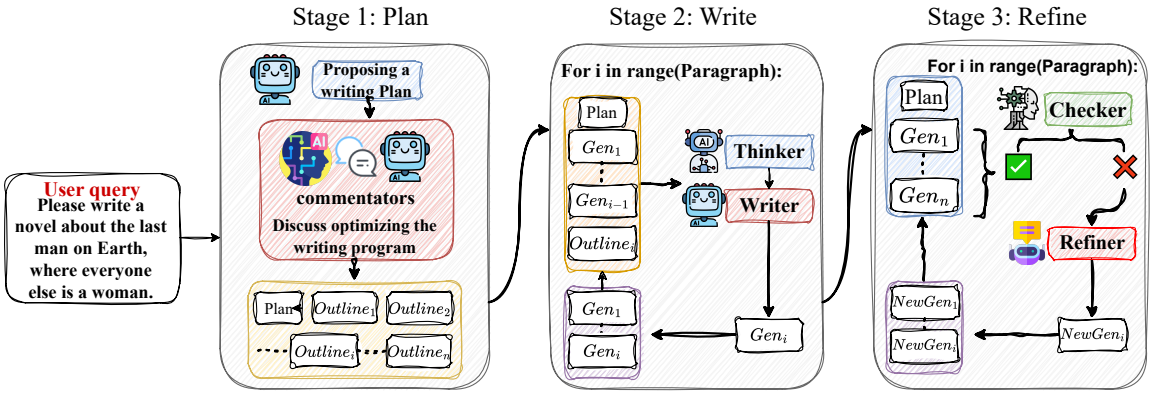

Figure 2: This figure illustrates a three-stage agent framework for long-form generation. In Stage 1 (Plan), the framework proposes a structured writing plan through discussions between AI commentators and a writer. In Stage 2 (Write), the text is incrementally generated using a thinker-writer collaboration, and in Stage 3 (Refine), a checker and refiner iteratively improve the generated text to enhance coherence and quality.

sists of finished texts (e.g., articles, books), which omit the underlying planning and structuring processes. However, writing—especially long-form writing—requires explicit thinking steps for maintaining coherence, logical flow, and structural consistency.

To address this gap, we propose the *SuperWriter*-agent (illustrated in Figure 2), a framework designed to generate high-quality, thought-enriched supervised fine-tuning data for writing. The *SuperWriter*-agent enables structured content generation through three coordinated stages: careful planning, targeted paragraph-level writing, and iterative refinement. This process explicitly embeds intermediate thinking signals into the writing pipeline, thereby enhancing the fluency, coherence, and narrative consistency of the generated text.

3.1 Stage 1: Plan

Inspired by the Story Workshop[1](Schultz, 1977; Tatiana, 2021; Sheflett, 1973), Stage 1 of *SuperWriter*-agent begins with oral narration and iterative dialogue to distill and expand initial ideas. This planning stage guides discussion agents in articulating core themes, central arguments, character backgrounds (for genres like fiction), and paragraph-level structures, collectively forming the Background component. By systematically allocating word counts and linking key ideas to paragraph units, it produces a detailed outline for downstream writing. This framework enhances coherence and organization, enabling agents to strategically develop and refine ideas for more focused and well-developed outputs. AppendixA.1 provides the detailed prompt for this stage.

3.2 Stage 2: Write

Motivated by recent advancements in reasoning-oriented LLMs—notably their remarkable scalability during inference in tasks such as mathematical reasoning and code generation—we build upon paradigms exemplified by reasoning models like OpenAI o1 and DeepSeek-R1. (OpenAI, 2024; DeepSeek-AI et al., 2025a). These models typically engage in an explicit reasoning and planning phase prior to generating final responses, which has proven highly effective in enhancing output quality. Following this paradigm, we propose a two-stage generation framework that simulates the process of "thinking before writing," aiming to produce structurally coherent and logically consistent paragraphs. Appendix A.2 provides the detailed prompt for the actual paragraph write stage.

- **Thinker Step:** In this initial phase, the model refrains from generating surface-level text. Instead, it identifies and organizes key ideas, thematic elements, and logical structures relevant to the paragraph. This explicit reasoning process provides a clear directional scaffold for subsequent text generation.
- **Writer Step:** Building on the structured outline from the Thinker stage and incorporating the preceding paragraph (i.e., the $(n-1)$-th paragraph) as contextual input, the model proceeds to generate the current paragraph. This use of prior

[1]A writing instruction method developed at Columbia College Chicago that emphasizes oral storytelling, collaborative discussion, and reflective dialogue for idea generation and refinement.

context ensures smooth transitions between paragraphs and contributes to the overall logical flow of the document.

3.3 Stage 3: Refine

The final refinement stage goes beyond superficial edits by systematically evaluating the overall quality of the generated text and identifying specific paragraphs that require targeted revisions.

Specifically, the refinement workflow consists of two key steps:

- **Checker Step:** The model conducts a comprehensive assessment of each paragraph, identifying issues such as logical inconsistencies, unclear expressions, or deviations from the intended narrative structure.
- **Editor Step:** Based on the feedback from the Checker stage, the model performs precise and targeted modifications to improve textual accuracy, fluency, and structural coherence.

This iterative and structured refinement process ensures that the final output not only accurately conveys the original intent and narrative objectives but also meets the rigorous standards expected in academic writing. Appendix A.3 provides the detailed prompts for paragraph refine stage.

4 *SuperWriter*-LM

Following the development of the *SuperWriter*-agent, which introduces structured thinking and iterative discussion mechanisms to significantly enhance text generation quality, we are motivated to explore a central research question: *Can large language models, when guided by the thinking paradigm provided by the SuperWriter-agent data, internalize the ability to generate high-quality long-form content through substantially fewer inference steps—rather than relying on 30 to 40 separate agent calls per sample?*

To answer this question, we conduct targeted model training experiments. Our objective is not merely to extend output length, but to fundamentally improve coherence, relevance, and depth by distilling the agent's structured thinking process directly into the model itself. In the following sections, we describe the construction of our high-quality training dataset and the strategic methodology for training LLMs to acquire and internalize the *SuperWriter*-agent's structured thinking and writing capabilities.

4.1 SFT-Training

Our training dataset combines English and Chinese instructions from two real-world instruction tuning datasets: WildChat-1M (Zhao et al., 2024) and LMSYS-Chat-1M (Zheng et al., 2023a). To ensure quality and relevance for long-form writing, we apply a filtering process using the DeepSeek-R1-Distill-Qwen-32B model (DeepSeek-AI et al., 2025a), as detailed in Appendix A.7.

We generate SFT data using the *SuperWriter*-agent (powered by GPT-4o-2024-08-06 (OpenAI et al., 2024)) from 4,000 filtered instructions. Each instance follows a structured pipeline: query $\rightarrow$ outline $\rightarrow$ draft $\rightarrow$ final output, divided into three stages matching the agent's internal design: *plan* (query $\rightarrow$ outline), *write* (outline $\rightarrow$ draft), and *refine* (draft $\rightarrow$ final output). Instead of training on full instruction-to-answer pairs, we use stage-wise training (Fig 3) for two reasons: (1) it aligns with real-world workflows where users review intermediate steps (e.g., outlines), and (2) complete outputs can exceed 100K tokens, challenging long-context models. Segmenting the process keeps each sample under 32K tokens, ensuring model tractability. Each stage—*plan*, *write*, and *refine*—includes 4,000 instances, totaling 12,000 high-quality training samples. During inference, generation proceeds in these three sequential stages.

We train our *SuperWriter*-LM model on Qwen2.5-7B (Yang et al., 2024), which supports up to 128K context tokens—ideal for long-form generation. To boost efficiency, we adopt the packing-based training strategy with loss weighting from Bai et al. (2024a). Training is performed on a single node with 8$\times$H800 80GB GPUs using DeepSpeed (ZeRO-3 with CPU offloading) (Rasley et al., 2020), with a batch size of 32, learning rate of 2×10^{-5}, and 32K-token context window. The model is trained for four epochs.

4.2 Hierarchical DPO for Multi-Stage Generation

Direct Preference Optimization (DPO) has demonstrated effectiveness in aligning policies directly with pairwise human (or proxy-model) preferences for *single-pass* generation tasks (Rafailov et al., 2024). However, in our scenario, the writing process unfolds sequentially over three distinct stages: *planning*, *drafting*, and *refinement*. Applying conventional DPO exclusively to final outputs neglects the valuable preference signals inherently

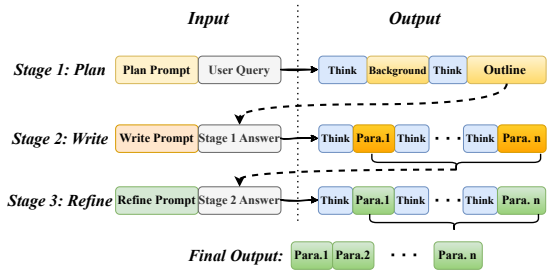

Figure 3: Format of the SFT dataset constructed by the *SuperWriter*-agent. The agent generates training data across three stages—planning, writing, and refining—each incorporating intermediate "Think" steps. During inference, the output of Stage 3 is used as the final answer.

present in earlier stages. To address this gap, we introduce a hierarchical multi-stage DPO framework that combines structured preference-data construction with systematic evaluation, inspired by process-annotation methods such as TPO (Liao et al., 2025), CHIP (Fu et al., 2025), and Math-Shepherd (Wang et al., 2024a).

Specifically, as depicted in Figure 4, the writing process is structured as a tree explored through Monte-Carlo Tree Search. Each path through this tree, labeled (i, j, k), corresponds sequentially to Stage-1 (*plan* i), Stage-2 (*draft* j), and Stage-3 (*refinement* k). We embed two key assumptions: well-structured initial *plans* lead to higher-quality draft (Stage-1 Plan $\rightarrow$ Stage-2 Write) and well-refined drafts typically yield better final outputs (Stage-2 Write $\rightarrow$ Stage-2 Refine). Consequently, we *back-propagate* quality signals from leaf nodes (final outputs) upwards through intermediate stages, ensuring the policy learns from decisions at every level rather than only from final outcomes.

Structured evaluation (*Write-judge*). To score the final output at each leaf, we introduce *Write-judge*[2], a six-dimension rubric (0–10 each) chosen from a larger pool of twenty dimensions according to the instruction type (e.g. creativity vs. logical coherence). Following best practice to curb evaluation bias, we use the QwQ-32B model (Team, 2025) to score every output three times under the same temperature settings and take the average. Then we propagate scores from the leaf nodes upward to construct DPO pairs as follows.

Step 1: Leaf-score discretization. Let s_{ijk} denote the averaged raw evaluation score assigned to leaf (i, j, k), and let N represent the total number of leaf nodes. We define the ranking and percentile for each leaf node as follows:

$$\begin{aligned} \mathrm{rank}_{ijk} &= 1 + \left|\{(p,q,r) \mid s_{pqr} > s_{ijk}\}\right|, \\ \pi_{ijk} &= \frac{\mathrm{rank}_{ijk}}{N}. \end{aligned} \tag{1}$$

The percentile scores π_{ijk} are subsequently discretized into ordinal rewards r_{ijk}:

$$r_{ijk} = \begin{cases} +2, & \pi_{ijk} \le \frac{1}{6}, \\ +1, & \frac{1}{6} < \pi_{ijk} \le \frac{2}{6}, \\ 0, & \frac{2}{6} < \pi_{ijk} \le \frac{4}{6}, \\ -1, & \frac{4}{6} < \pi_{ijk} \le \frac{5}{6}, \\ -2, & \pi_{ijk} > \frac{5}{6}. \end{cases} \tag{2}$$

Step 2: Reward aggregation. Ordinal rewards are propagated upwards through the tree by averaging across child nodes at each intermediate stage:

$$\begin{aligned} \hat{r}_{ij} &= \frac{1}{|\{k\}|} \sum_k r_{ijk} \quad \text{(Stage 2 node)} \\ \hat{r}_i &= \frac{1}{|\{j\}|} \sum_j \hat{r}_{ij} \quad \text{(Stage 1 node)} \end{aligned} \tag{3}$$

Step 3: Preference-pair construction. We harvest pairwise preferences at *every* hierarchy level, ensuring that each decision contributes to the optimization at every step.

L1. Stage 1 (Plan). Let $B = \arg\max_i \hat{r}_i$ be the best plan and W_1, W_2 the worst two.

$$P_1 = \{\,(B,\, W_1),\ (B,\, W_2)\}\,. \tag{4}$$

L2. Stage 2 (Write). For the best two plans $i \in \{B, S\}$ (with S being the second best),

$$\begin{aligned} P_2(i) &= \left(\arg\max_j \hat{r}_{ij},\ \arg\min_j \hat{r}_{ij}\right), \\ P_2 &= \bigcup_i P_2(i). \end{aligned} \tag{5}$$

L3. Stage 3 (Refine). For each $(i, j^\star) \in P_2$,

$$\begin{aligned} P_3(i, j^\star) &= \left(\arg\max_k s_{ijk},\ \arg\min_k s_{ijk}\right), \\ P_3 &= \bigcup_{(i,j^\star)} P_3(i, j^\star). \end{aligned} \tag{6}$$

The complete preference dataset is therefore

$$\mathcal{D}_{\text{DPO}} = P_1 \cup P_2 \cup P_3. \tag{7}$$

[2]Pipeline details and prompts appear in App. A.4.

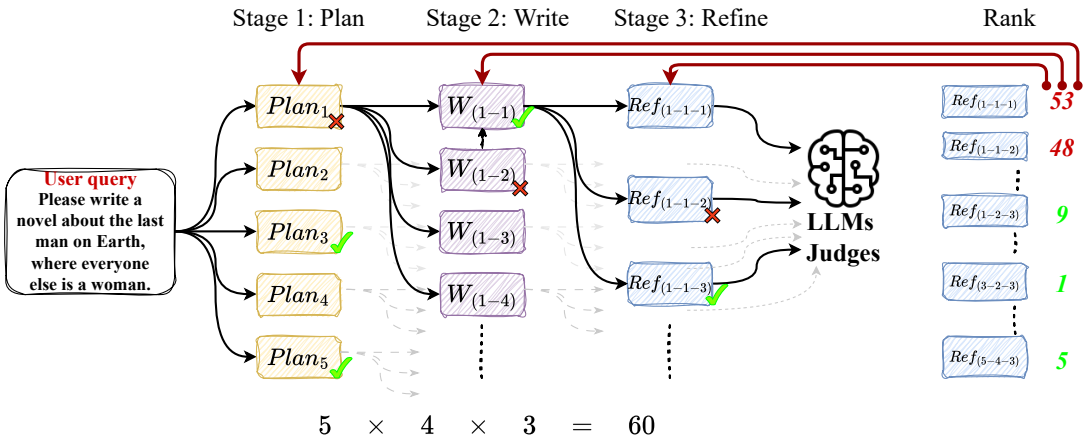

Figure 4: The MCTS begins with 5 distinct writing plans, each leading to 4 written drafts, totaling 20 initial outputs. Each draft is then refined 3 times, resulting in 60 unique final outputs for a single root. A judge LLM ranks these final outputs, and the rankings are back-propagated through the refinement and planning stages. This scoring mechanism helps identify the most effective planning and writing strategies, thereby optimizing the overall writing process.

Optimization objective. Finally, we optimize the policy π_θ using the standard DPO loss:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}_{\text{DPO}}}\Big[\log\sigma\big(\beta\,(s_\theta(x,y^+) - s_\theta(x,y^-))\big)\Big]. \tag{8}$$

Using the above method, we obtained a DPO preference dataset and employed 360-LLaMA-factory (Haosheng Zou and Zhang, 2024) to continue context-parallel DPO training of the supervised fine-tuned *SuperWriter*-LM with a batch size of 32 and a learning rate of 1×10^{-6}.

5 Experiment

5.1 Experiment Setup

Inference Setup. We use SGLang (Zheng et al., 2023c), which optimizes key-value (KV) cache memory for efficient large-scale inference, reducing memory overhead and maximizing throughput. Inference is conducted in BFloat16 on 8× NVIDIA H800 GPUs, ensuring consistency and efficiency.

Benchmark Setup. We conduct experiments on two datasets. The first, WritingBench (Wu et al., 2025b), is a comprehensive benchmark evaluating large language models across six writing domains and 100 sub-domains, covering creative, persuasive, informational, and technical tasks. It contains 1,200 real-world prompts, each with five instance-specific evaluation criteria. Using a query-dependent framework, WritingBench employs a Qwen2.5-7B critic fine-tuned on 50K human-labeled samples to assess style, format, and length, achieving 83% agreement with human judgments.

The second dataset comprises 200 real-world user instructions manually curated by us. We assess model performance via pairwise Win-rate comparisons against baselines including LongWriter-8B (Bai et al., 2024b), DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025a), Writing-Model-Qwen-7B (Wu et al., 2025b)[3], Qwen3 (Team, 2024), LLaMA-4-Scout (AI, 2025), DeepSeek-V3 (DeepSeek-AI et al., 2025b), and DeepSeek-R1 (DeepSeek-AI et al., 2025a). All models use consistent decoding settings (temperature = 0.6, top-p = 0.95), and win rates quantify comparative performance.

5.2 WritingBench Result

We evaluate our 7B sized *SuperWriter*-LM model on the WritingBench benchmark. As shown in Tab 1, *SuperWriter*-LM achieves an overall performance (Avg) of 8.51—second-best among all models, closely following DeepSeek-R1 (DeepSeek-AI et al., 2025a). It demonstrates strong capabilities in both Chinese–8.6 and English–8.5, even matching DeepSeek-R1 in English performance.

Across different domains, *SuperWriter*-LM achieves the highest scores in three major areas: (D1) Academic & Engineering–8.6, (D2) Finance & Business–8.7, (D3) Politics & Law–8.7 and (D5) Education–8.7, even slightly outperforming the

[3]SFT model obtained via rejection sampling using the WritingBench critic.

Models	Avg	Languages		Domains						Requirements					
		ZH	EN	D1	D2	D3	D4	D5	D6	R1	C	R2	C	R3	C
Proprietary LLMs															
ChatGPT-4o-latest	8.16	8.3	8.1	8.1	8.1	8.2	8.1	8.4	8.1	8.3	8.7	8.2	8.9	8.2	8.3
o1-Preview	8.15	8.1	8.2	8.0	8.1	8.2	8.2	8.4	8.1	8.2	8.6	8.2	8.8	8.2	8.2
Claude-3.5-Sonnet	7.71	7.7	7.7	7.6	7.5	7.6	7.7	7.9	8.0	7.9	8.5	7.7	8.5	7.9	8.0
Gemini-1.5-Pro	7.78	7.8	7.7	7.7	7.5	7.8	7.9	8.0	7.9	7.9	8.6	7.9	8.8	7.9	8.0
Qwen-Max	8.37	8.4	8.3	8.3	8.3	8.4	8.4	8.5	8.4	8.5	8.7	8.4	**9.0**	8.4	8.5
Open-source LLMs															
Deepseek-R1	**8.55**	**8.7**	**8.5**	8.5	8.5	8.6	**8.6**	**8.7**	**8.6**	**8.7**	**8.9**	**8.6**	**9.0**	**8.6**	**8.7**
Deepseek-V3	7.95	8.0	7.9	7.9	7.8	8.0	7.8	8.2	8.0	8.1	8.6	8.0	8.9	8.0	8.2
Mistral-Large-Instruct	7.64	7.6	7.7	7.7	7.6	7.8	7.3	7.9	7.6	7.7	8.2	7.7	8.7	7.7	7.9
Qwen-2.5-72B-Instruct	7.90	8.0	7.9	8.0	7.8	8.1	7.7	8.2	7.8	8.0	8.3	8.0	8.8	7.9	8.0
Qwen-2.5-7B-Instruct	7.43	7.3	7.5	7.7	7.4	7.6	6.9	7.8	7.3	7.5	7.9	7.6	8.6	7.4	7.5
Llama-3.3-70B-Instruct	7.01	6.7	7.3	7.0	6.9	7.0	6.8	7.3	7.3	7.1	7.8	7.1	8.2	7.0	7.2
Llama-3.1-8B-Instruct	6.35	5.7	6.9	6.6	6.4	6.1	6.0	6.7	6.6	6.4	7.0	6.4	7.6	6.3	6.4
Capability-enhanced LLMs															
Suri-I-ORPO	4.97	4.4	5.5	5.6	5.3	5.0	4.1	5.0	5.1	4.8	5.2	5.0	5.4	4.5	4.0
LongWriter-8B	7.91	7.9	7.9	8.0	8.1	8.1	7.7	8.1	7.6	7.9	8.2	8.1	8.8	7.7	7.7
Writing-Model-Qwen-7B	8.49	8.6	8.4	8.4	8.4	8.6	8.4	8.6	8.5	8.6	8.8	8.5	9.0	8.5	8.6
***SuperWriter*-LM (ours)**	8.51	8.6	**8.5**	**8.6**	**8.7**	**8.7**	8.2	**8.7**	8.2	8.4	8.5	**8.6**	8.4	8.0	6.3

Table 1: WritingBench performance of different LLMs across 6 domains and 3 writing requirements evaluated with our critic model (scale: 1-10). The six domains include: (D1) Academic & Engineering, (D2) Finance & Business, (D3) Politics & Law, (D4) Literature & Art, (D5) Education, and (D6) Advertising & Marketing. The three writing requirements assessed are: (R1) Style, (R2) Format, and (R3) Length. Here, "C" indicates category-specific score.

DeepSeek-R1 model. Additionally, *SuperWriter*-LM satisfies various special writing requirements, except the length_C setting. This discrepancy is primarily due to the nature of agent-generated data, which tends to produce longer outputs even for short-text tasks—an issue that does not affect long-form generations.

5.3 Win-Rate Result

WritingBench adopts a critic model to evaluate model outputs by assigning scores (ranging from 1 to 10) across 4–5 distinct dimensions. However, this evaluation approach has several limitations. First, due to the relatively small size of the critic model, it may be vulnerable to some word/sentence hacking. Second, we observe that WritingBench primarily focuses on formal or professional writing tasks—such as summaries and reports—whereas our analysis of real-world data from *WildChat* (Zhao et al., 2024) and *LMSYS-Chat-1M* (Zheng et al., 2023a) reveals that creative writing (e.g., storytelling, fiction) constitutes a significant portion of user queries. To address these concerns, we adopt a more direct and interpretable evaluation metric: **win-rate**. We evaluate model performance on nearly 200 real-world user queries collected from the aforementioned datasets. For each query, responses are generated by *SuperWriter*-LM and six baseline models.

LLM Evaluation. For automatic evaluation, we adopt LLM-as-a-judge (Bai et al., 2023; Zheng et al., 2023b) to perform pairwise comparison using `GPT-4.1-2025-04-14` (OpenAI, 2025). To mitigate positional bias, we conduct two evaluations per pair by swapping the response order: `Evaluation_Prompt+A+B` and `Evaluation_Prompt+B+A`. Based on the two judgments, we categorize the results into win, loss, or tie and compute win-rate results[4].

As shown in Figure 5, *SuperWriter*-LM demonstrates a substantial performance lead among models of the same size (models 1, 2, and 3). Furthermore, in comparisons against larger-sized models (models 4, 5, 6, and 7), *SuperWriter*-LM remains competitive and, in some cases, even slightly outperforms state-of-the-art LLMs. Taken together, these results suggest that *SuperWriter*-LM sets a new performance benchmark among 7B-scale models and even challenges the capabilities of existing state-of-the-art systems. We also present several case studies in Appendix A.6.

Human Evaluation. To mitigate potential inaccuracies in automatic evaluation, we conduct

[4]Evaluation prompts are provided in Appendix A.5.

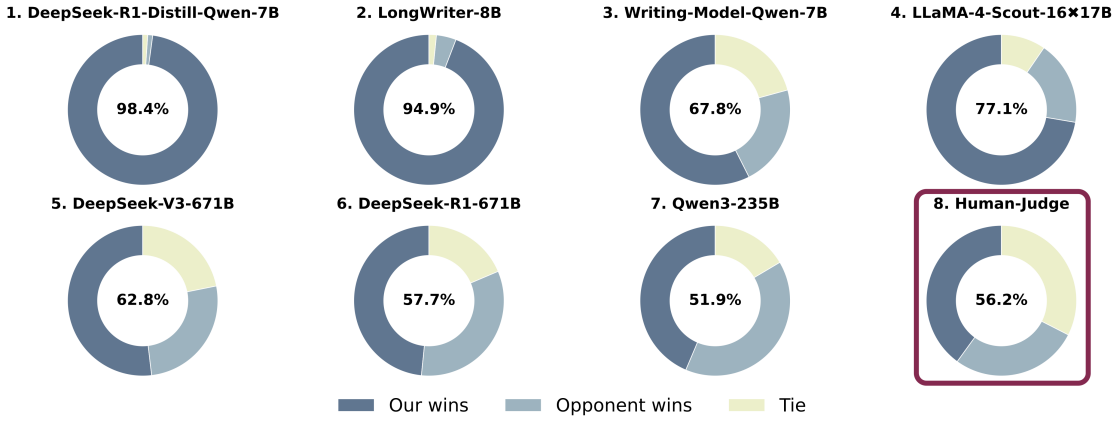

Figure 5: This figure presents eight donut charts comparing the win rates of our model against seven different baselines. The win-rate in the 8th chart is evaluated by human (Ours vs. Writing-Model-Qwen-7B) while the rest are evaluated by GPT-4.1. Each chart shows the proportions of wins, losses, and ties, with the central percentage indicating the overall win rate. When calculating the win rate, a tie is counted as 0.5 win for both sides.

Model	Avg	ZH	EN
Qwen2.5-7B-Instruct	7.43	7.3	7.5
+ *SuperWriter final output*	8.21	8.3	8.2
+ *Three-Stage*	8.47	8.5	8.4
+ *Hierarchical DPO*	**8.51**	**8.6**	**8.5**

Table 2: Ablation study on WritingBench (Wu et al., 2025b): average, Chinese (ZH), and English (EN) performance scores.

an human supplementary assessment on approximately 200 real-world user queries, comparing *SuperWriter*-LM with Writing-Model-Qwen-7B (Wu et al., 2025b). For each query, three independent annotators with undergraduate degrees. were tasked with evaluating and determining the preferred response, with outcomes categorized as win, loss, or tie—following the same standard as the automatic evaluation. The aggregated results are shown in Figure 5 (8), and *SuperWriter*-LM demonstrates stronger performance under human judgment. However, due to the annotators' tendency to assign a tie when the differences between two responses are subtle, the overall win rate appears slightly lower.

5.4 Ablation Study

Finally, we conduct an ablation study comprising four different setups evaluated on the WritingBench benchmark. The first setup uses the base model, *Qwen2.5-Instruct*, as the performance baseline. The second setup, *SuperWriter-final-answer*, takes the user query as input and produces the final output from the Stage-3 *refine* step of the *SuperWriter*-agent—this is a one-pass generation without any explicit thinking process and achieves an average score of 8.21 (ZH: 8.3, EN: 8.2). The third setup, *+Three-Stage*, corresponds to our SFT-trained model, which explicitly performs planning, drafting, and refining in a chained, multi-stage manner, incorporating structured thinking. This setup further improves performance to 8.47 (ZH: 8.5, EN: 8.4). The final setup is the full model further enhanced with our hierarchical DPO optimization, reaching the highest score of 8.51 (ZH: 8.6, EN: 8.5). As shown in Table 2, each additional component leads to consistent performance improvements, demonstrating the effectiveness of our proposed approach in structured writing tasks.

6 Conclusion

SuperWriter addresses the challenge of long-form text generation by introducing a structured writing process—planning, writing, and refining—guided by the *SuperWriter*-agent. This approach teaches the model to "think before writing" and produces high-quality supervision signals. Combined with a hierarchical DPO strategy, the model learns to align its output across all writing stages.

Experiments show strong results: *SuperWriter*-LM outperforms all same-size models on WritingBench and even exceeds the 671B DeepSeek-R1 model (DeepSeek-AI et al., 2025a) in key domains. It also wins over 98% of real-user comparisons against top open-source baselines. These results confirm the value of multi-stage generation and structured preference learning for improving writing quality.

7 Limitation

While *SuperWriter*-LM demonstrates strong performance in long-form text generation, several limitations remain:

(1) Inference latency. Compared to single-pass generation models such as *LongWriter* (Bai et al., 2024b) or *Suri* (Pham et al., 2024), our method incurs additional inference time due to its three-stage Framework. Although this is significantly more efficient than multi-round agent-based pipelines (e.g., requiring 30–40 calls per output), the structured *plan* $\rightarrow$ *write* $\rightarrow$ *refine* process still requires three sequential forward passes, which may increase user-perceived latency in real-world applications.

(2) Model scale. Our current implementation is built on a 7B parameter backbone (Qwen2.5)(Qwen et al., 2025), which strikes a balance between performance and cost. However, this moderate scale may limit the model's internal world knowledge, particularly in knowledge-intensive or specialized writing scenarios (e.g., legal, medical, and scientific domains). In our qualitative analysis, some outputs exhibited shallow factual grounding or subtle reasoning errors.

(3) Lack of online reinforcement learning. Lack of online reinforcement learning. Our alignment stage relies solely on offline Direct Preference Optimization (DPO) (Rafailov et al., 2024), trained from static preference pairs. While effective, this setup lacks the adaptivity of online Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2023), which allows models to continually refine outputs through exploration. The key bottleneck is the high rollout cost when applying general-purpose reward models to long outputs. Designing scalable, low-latency reward models or reward distillation methods (Bai et al., 2022; Ouyang et al., 2022) for long-form tasks is thus a promising direction for future research.

References

Meta AI. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation.

Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. 2024. Training-free long-context scaling of large language models. *arXiv preprint arXiv:2402.17463*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024a. LongAlign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395, Miami, Florida, USA. Association for Computational Linguistics.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, and 1 others. 2023. Benchmarking foundation models with language-model-as-an-examiner. *Advances in Neural Information Processing Systems*, 36:78142–78167.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024b. Longwriter: Unleashing 10,000+ word generation from long context llms. *arXiv preprint arXiv:2408.07055*.

Anne Becker. 2006. *Revision: History, theory, and practice*, pages 25–49.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences. *Preprint*, arXiv:1706.03741.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proc. of NAACL*, pages 4599–4610.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others.

2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

L Flower. 1981. A cognitive process theory of writing. *Composition and communication*.

Jinlan Fu, Shenzhen Huangfu, Hao Fei, Xiaoyu Shen, Bryan Hooi, Xipeng Qiu, and See-Kiong Ng. 2025. Chip: Cross-modal hierarchical direct preference optimization for multimodal llms. *arXiv preprint arXiv:2501.16629*.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. 2024. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*.

Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. 2019. Plan, write, and revise: an interactive system for open-domain story generation. *Preprint*, arXiv:1904.02357.

Allan Gollins and Dedre Gentner. 2016. A framework for a cognitive theory of writing. In *Cognitive processes in writing*, pages 51–72. Routledge.

Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2023. Lm-infinite: Simple on-the-fly length generalization for large language models. *arXiv preprint arXiv:2308.16137*.

Shousheng Jia Haosheng Zou, Xiaowei Lv and Xiangzheng Zhang. 2024. 360-llama-factory.

Xinyu Hua and Lu Wang. 2020. Pair: Planning and iterative refinement in pre-trained transformers for long text generation. *Preprint*, arXiv:2010.02301.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325*.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, Chien Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. 2024. Longlamp: A benchmark for personalized long-form text generation. *Preprint*, arXiv:2407.11016.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-In Lee, and Moontae Lee. 2023. QASA: Advanced question answering on scientific articles. In *Proc. of ICML*, pages 19036–19052.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023. Teach llms to personalize - an approach inspired by writing education. *ArXiv*.

Weibin Liao, Xu Chu, and Yasha Wang. 2025. TPO: Aligning large language models with multi-branch & multi-step preference trees. In *The Thirteenth International Conference on Learning Representations*.

Siyang Liu, Naihao Deng, Sahand Sabour, Yilin Jia, Minlie Huang, and Rada Mihalcea. 2023. Task-adaptive tokenization: Enhancing long-form text generation efficacy in mental health and beyond. In *Proc. of EMNLP*, pages 15264–15281.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *Preprint*, arXiv:2402.06196.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2024. Learning to reason with llms.

OpenAI. 2025. Introducing gpt-4.1 in the api. Accessed: 2025-05-02.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Bo Liu, Aastha Jhunjhunwala, Zhilin Wang, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Data, data everywhere: A guide for pretraining dataset construction. *arXiv preprint arXiv:2407.06380*.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Chau Minh Pham, Simeng Sun, and Mohit Iyyer. 2024. Suri: Multi-constraint instruction following for long-form text generation. *arXiv preprint arXiv:2406.19371*.

Shanghaoran Quan, Tianyi Tang, Bowen Yu, An Yang, Dayiheng Liu, Bofei Gao, Jianhong Tu, Yichang Zhang, Jingren Zhou, and Junyang Lin. 2024. Language models can self-lengthen to generate long texts. *arXiv preprint arXiv:2410.23933*.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, Junran Peng, Zhaoxiang Zhang, Songyang Zhang, and Kai Chen. 2024. Hellobench: Evaluating long text generation capabilities of large language models. *Preprint*, arXiv:2409.16191.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Zeeshan Rasheed, Muhammad Waseem, Kai Kristian Kemell, Aakash Ahmad, Malik Abdul Sami, Jussi Rasku, Kari Systä, and Pekka Abrahamsson. 2025. Large language models for code generation: The practitioners perspective. *Preprint*, arXiv:2501.16998.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Alireza Salemi, Julian Killingback, and Hamed Zamani. 2025a. Expert: Effective and explainable evaluation of personalized long-form text generation. *Preprint*, arXiv:2501.14956.

Alireza Salemi, Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, Tao Chen, Zhuowan Li, Michael Bendersky, and Hamed Zamani. 2025b. Reasoning-enhanced self-training for long-form personalized text generation. *Preprint*, arXiv:2501.04167.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *Preprint*, arXiv:2501.04227.

John Schultz. 1977. The story workshop method: Writing from start to finish. *College English*, 39(4):381–389.

Betty Sheflett. 1973. Story workshop as a method of teaching writing. *College English*, 35(2):141–160.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-versio

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proc. of EMNLP*, pages 8273–8288.

Haochen Tan, Zhijiang Guo, Zhan Shi, Lu Xu, Zhili Liu, Yunlong Feng, Xiaoguang Li, Yasheng Wang, Lifeng Shang, Qun Liu, and Linqi Song. 2024. ProxyQA: An alternative framework for evaluating long-form text generation with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6806–6827, Bangkok, Thailand. Association for Computational Linguistics.

Pospelova Tatiana. 2021. The collaborative discussion model: Developing writing skills through prewriting discussion. *Journal of Language and Education*, 7(1 (25)):156–170.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

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

Qianyue Wang, Jinwu Hu, Zhengping Li, Yufeng Wang, daiyuan li, Yu Hu, and Mingkui Tan. 2024b. Generating long-form story using dynamic hierarchical outlining with memory-enhancement. *Preprint*, arXiv:2412.13575.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. 2024c. Autosurvey: Large language models can automatically write surveys. In *Proc. of NeurIPS*.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. *NeurIPS Datasets and Benchmarks Track*.

Yuhao Wu, Yushi Bai, Zhiqing Hu, Shangqing Tu, Ming Shan Hee, Juanzi Li, and Roy Ka-Wei Lee.

2025a. Shifting long-context llms research from input to output. *Preprint*, arXiv:2503.04723.

Yuhao Wu, Ming Shan Hee, Zhiqing Hu, and Roy Ka-Wei Lee. 2024. Longgenbench: Benchmarking long-form generation in long context llms. *arXiv preprint arXiv:2409.02076*.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. 2025b. Writingbench: A comprehensive benchmark for generative writing. *Preprint*, arXiv:2503.05244.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, and 1 others. 2024. Effective long-context scaling of foundation models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4643–4663.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 2023. DOC: Improving long story coherence with detailed outline control. In *Proc. of ACL*, pages 3378–3465.

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. In *Proc. of EMNLP*, pages 4393–4479.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. *Preprint*, arXiv:1811.05701.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024. Soaring from 4k to 400k: Extending llm's context with activation beacon. *arXiv preprint arXiv:2401.03462*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric. P Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2023a. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *Preprint*, arXiv:2309.11998.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2023c. Sglang: Efficient execution of structured language model programs. *arXiv preprint arXiv:2312.07104*.

A Appendix

A.1 Stage-1 Plan Prompt

This appendix provides a brief overview of the prompt modules used in *SuperWriter*-Agent Stage-1 Plan. There are a total of **6 modules**, each serving a specific role in the writing and evaluation pipeline:

- **BrainStorm**: Generates an initial in-depth thinking process to analyze and develop a preliminary writing plan for a given task, ensuring a comprehensive and thorough design.
- **BrainStorm Review**: Critically evaluates the task design, raising questions about potential flaws, ambiguities, or unclear requirements to refine the task's overall logic and readability.
- **BrainStorm Refine**: Integrates reviewer feedback into the task design by applying editorial judgment to revise or completely rewrite the task, ensuring it is rigorous and well-structured.
- **Outline**: Constructs a structured article outline based on the task design, including the estimated word count per paragraph and a logical framework to guide the writing process.
- **Check outline**: Acts as a reviewer evaluating the logical structure and completeness of the outline, pointing out logical gaps or missing elements to ensure it aligns with the intended objectives.
- **Refine outline**: Edits and improves the previously generated outline based on reviewer feedback, ensuring clarity, completeness, and alignment with the writing objectives.

The following sections provide the detailed prompt templates and usage notes for each module.

Think template

1. Define the Purpose and Type of Writing - Purpose: Clearly establish the objective of the piece (e.g., to inform, persuade, or inspire), setting the tone and direction accordingly. - Type: Choose an appropriate writing style (e.g., argumentative, expository, or business writing) that aligns with the intended format and structure. Clearly identify the genre and describe its stylistic characteristics.
2. Plan Content and Structure - Key Points: Outline the essential information to ensure a clear and focused topic. - Structure: Develop a coherent framework that maintains a logical flow throughout the piece.
3. Characters and Plot (for Narrative Writing) - Character Development: Define the traits and motivations of all characters. Provide detailed descriptions, including specifics such as names, gender, and relationships. - Plot Development: Establish pivotal plot points and emotional cues to drive the narrative. Offer a detailed explanation based on the chosen structure.
4. Additional Guidelines - Formatting Requirements: Automatically select an appropriate output format (e.g., Markdown, bullet points) based on content and presentation needs to enhance visual clarity and appeal. - Other Key Elements: Include any genre- or task-specific components. For instance, in summaries, a step-by-step approach is sufficient due to their simpler logic.

BrainStorm

You are a professional writer responsible for creating an initial design based on the thinking template below. Please use the template to thoroughly analyze and develop a detailed preliminary plan for the task 'topic'. Carefully examine each point in the template to ensure all aspects are fully considered, providing a comprehensive and complete writing plan with no vague information. At the same time, make sure the structure remains manageable and not overly complex. Thinking Template: think_template. At this stage, you do not need to provide an outline—just complete the in-depth thinking required for task design. Please respond:

BrainStorm Review

You are a critical reviewer, specializing in identifying issues and flaws in task design. Please evaluate the following task design and raise at least two questions to ensure it meets the requirements of 'topic'. These questions should be carefully considered and clarified before the actual writing begins, highlighting any logical flaws, ambiguities, or areas that might confuse the reader. Depending on the type of task, you will analyze it from various angles—for example, whether a speech is engaging, a story is original, or an academic paper is rigorous. Task Topic: topic. Task Design: task_output. Provide detailed questions along with specific and practical suggestions for improving the task design:

BrainStorm Refine

You are an experienced editor responsible for revising the task based on the reviewer's feedback. Original Task Design: task_output. Feedback: feedback. Using this feedback, provide a detailed and specific revision of the current task design. Apply your own judgment rather than simply implementing the feedback verbatim to address all identified issues. If necessary, rewrite the original task design entirely. Please provide the revised task design:

Outline

Thoroughly understand the task design of task_define_result. Based on the task design, determine a suitable article title and generate a detailed, structured outline. The outline should not exceed 20 paragraphs, and each paragraph must include a detailed description along with a specific word count. The total word count should be appropriate for the task but must not exceed 16,000 words. Allocate the total word count to each paragraph according to the complexity of the task. Then, provide the full outline along with the word count for each paragraph. Ensure that each paragraph description includes the expected content, maintains clear logic, and aligns with the user's objective: topic.

Check outline

You are a reviewer. Your task is to evaluate the following outline and assess whether its logical structure is complete and aligned with the intended objectives. Task Design: task_define_result Outline: outline Please provide detailed feedback, pointing out any logical gaps or missing elements.

Refine outline

You are a professional outline editor. Based on the following feedback, revise the outline to ensure it includes all necessary components: Feedback: check_output Current Outline: outline. Please provide the revised article title and the updated outline content:

A.2 Stage-2 Write Prompt

This appendix provides an overview of the prompt modules used in *SuperWriter*-Agent Stage-2 Write. There are a total of **2 modules**, each serving a specific role in the writing pipeline:

- **Write-thinker**: This module is designed to guide the planning phase for each paragraph. It takes the structured outline, the previous paragraphs, and the key point for the current paragraph as input. The module then prompts the model to develop a detailed thought process covering the paragraph's purpose, structure, transitions, details and examples, language style, and other relevant aspects. It ensures that each paragraph is planned thoroughly before the actual writing begins.

- **Write**: This module transforms the thought process from the Write-thinker stage into the actual written paragraph. It uses the same structured outline, previous paragraphs, key point, and the generated thought process as guidance to produce a coherent and logically sound paragraph. The output is formatted with delimiters (e.g., `$$content$$`) to clearly separate the paragraph from other text.

The following sections provide the detailed prompt templates and usage notes for each module.

Write-thinker

You are a writing expert skilled in thoughtful planning before generating each paragraph. Outline: `outline`
Previous Paragraphs: `previous_paragraphs`
Key Point for the Current Paragraph: `key_point`

Please carefully develop a writing plan for the new paragraph. You may consider the following aspects:
1. **Purpose**: What is the main objective of this paragraph? What message or emotion should it convey?
2. **Structure**: How should the content of this paragraph be organized? What logical sequence would best ensure clarity and coherence, and how will it connect tightly with the previous content?
3. **Transitions**: How will this paragraph naturally link to the one before it? Are there specific transition sentences or bridging techniques that can be used?
4. **Details and Examples**: What details, facts, or examples are needed to support the main idea? How should these be arranged for maximum impact?
5. **Language Style and Techniques**: What kind of language style should be used to achieve the goal? Are there rhetorical devices (such as metaphors or analogies) that could enhance the paragraph's impact—while still being clear, readable, and easy to understand for the audience?
6. **Markdown Format**: Use Markdown to structure the output neatly, including headings, bullet points, or bold text to improve readability.
Based on the outline and the key point for this paragraph, construct a detailed writing plan. Add any other relevant considerations as needed, and keep the word count requirements in mind. Only the thought process behind the paragraph is needed, not the paragraph itself.

Writer

You are an exceptional writing expert, skilled at completing writing tasks in a clear, accessible, and logically sound manner. Outline: `outline`
Previous Paragraphs: `previous_paragraphs`
Key Point for the New Paragraph: `key_point`
Thought Process: `thought_response`

Based on the thought process above and the existing paragraphs, write the next paragraph, ensuring it meets the word count requirement. Only provide the full content of the new paragraph. Enclose the paragraph content with $$content$$.

A.3 Stage-3 Refine Prompt

In Stage-3, the system focuses on reviewing and revising the paragraphs to ensure high quality and alignment with the overall document structure. This stage comprises two main steps:

- **Paragraph Review**: This module acts as a meticulous reviewer. It analyzes the entire document, ensuring that each paragraph is logically consistent, complete, and coherent within the context of the overall text. The reviewer provides at least two specific revision suggestions to address logical issues, missing details, or awkward transitions.

- **Paragraph Modification**: This module takes the reviewer's feedback and applies it to revise the paragraph. It strictly follows the feedback to ensure all identified issues are resolved. The revised paragraph is provided in isolation, enclosed with delimiters (e.g., `$$content$$`) to clearly separate it from other text.

Together, these modules ensure that the document achieves high logical coherence, completeness, and clarity throughout.

Stage-3 Refine: Paragraph Review

As a meticulous document reviewer, your task is to carefully read the entire document, understand its overall structure and logic, and then conduct a detailed review of paragraph `idx+1`, providing revision suggestions:
`combined_document`
When reviewing paragraph `idx+1`, you may refer to the following points:
1. Logical Consistency: Is this paragraph logically consistent with the rest of the document? Are there any illogical transitions or abrupt shifts?
2. Completeness: Does this paragraph provide enough information to support its main idea? Are there any important missing details?
3. Coherence: Does this paragraph connect smoothly with the surrounding paragraphs? Would transitional sentences help improve the flow?

Please provide at least two specific improvement suggestions.
Focus on offering detailed revision suggestions for paragraph `idx+1`. Only provide suggestions and possible ways to improve—do not rewrite the paragraph itself.
Suggestions for improvement:

Stage-3 Refine: Paragraph Modification

As a text editor, your task is to revise the paragraph based on the following feedback:
`review_feedback`
Original Paragraph:
`updated_document[idx]`
Ensure the revision strictly follows the specific suggestions in the feedback. Only provide the revised paragraph. Enclose the paragraph content with `$$`, like: `$$content$$` Revised Paragraph:

A.4 Evaluation Prompt for Hierarchical DPO

To support structured preference data construction for Direct Preference Optimization (DPO), we design and apply a sequence of modular prompts. Each serves a specific role within the evaluation pipeline. Below is an overview of their usage:

The evaluation pipeline comprises the following steps:

1. **Rubric Definition** (`evaluation_criteria`): Defines the complete set of General and Special evaluation dimensions. This rubric is reused across all queries.
2. **Criterion Selection Schema** (`format_query`): Specifies the JSON format for selecting six criteria (three General, three query-relevant Special) and rewriting their Definitions and Standards to match the specific query context.
3. **Criterion Selection Prompt**: Combines the rubric and schema to instruct the model to select and customize criteria. The output is a JSON object referred to as `evaluate_standard`.
4. **Scoring Format Schema** (`format_eval`): Specifies the expected evaluation output format: for each selected criterion, the model must return an *Analysis* string and a numeric *Score*.
5. **Final Scoring Prompt**: Provides the model with a query, its generated result, the customized `evaluate_standard`, and the `format_eval` schema. The model performs criterion-wise evaluation and outputs a structured JSON.

Outcome: This pipeline yields structured, query-specific evaluations that are interpretable, machine-parsable, and suitable for training with DPO loss.

`evaluation_criteria` Prompt

Evaluation Criteria
1. General Criteria (Applicable to All Genres)
1.1 Relevance
Definition: How well the content matches the user's request, and whether it addresses the intended purpose or topic.
Standards:
10: Fully aligned with the user's needs, highly relevant to the request.
7–9: Mostly relevant, with some minor deviations or less-than-perfect alignment.
4–6: Partially relevant, with the majority of the content not matching the user's request.
1–3: Completely irrelevant, fails to meet the user's needs.

1.2 Coherence
Definition: The clarity of the structure, and the logical flow of ideas and transitions between paragraphs and sentences.
Standards:
10: Clear structure, strong logical progression, and smooth flow of content with natural transitions.
7–9: Mostly coherent, with minor lapses or jumps in logic.
4–6: Structure is somewhat unclear, and there are noticeable gaps or jumps in logic.
1–3: Disorganized and confusing, with no clear structure or logical connections.

1.3 Clarity
Definition: How clear and easy the content is to understand, and whether it includes sufficient detail.
Standards:
10: Clear expression, rich in detail, and easy to understand.
7–9: Mostly clear, but may have some lengthy or unclear parts.
4–6: Expression is somewhat muddled, lacks detail, and is difficult to understand.
1–3: Extremely vague, minimal information, and difficult to comprehend.

2. Special Criteria (Applicable to Specific Genres)
2.1 Creativity and Uniqueness
Definition: Whether the content is innovative, offering new perspectives, or showcasing original expression.
Standards:
10: Highly creative and unique, presenting entirely new or unconventional ideas or perspectives.
7–9: Creative, but some parts are more conventional or lack originality.
4–6: Limited creativity, mostly traditional content with little innovation.
1–3: Lacks creativity, offering conventional or uninspired content.
.....

`format_eval` Prompt

The final output should be in JSON format, structured as follows: ```json
{
"Criterion 1": {
"Analysis": "Analysis content",
"Score": X
},
"Criterion 2": {
"Analysis": "Analysis content",
"Score": X
},
...
}
```

**`format_query` Prompt**

After think give final answer in JSON format. Structured as follows: ```json
{
"Selected Criterion 1": {
"Definition": "Criterion definition",
"Standards": "Scoring standards for the query, give a simple rubric from 1 to 10, 10: ..., 7-9: ..., 4-6: ..., 1-3: ..."
},
"Selected Criterion 2": {
"Definition": "Criterion definition",
"Standards": "..."
},
...
}
```

prompt

Please refer to the evaluation criteria outlined below: {evaluation_criteria}
Task:
You are tasked with evaluating the query: `{query}`. From the "Special Criteria" section, select 3 relevant criteria, and from the "General Criteria" section, select all criteria (Relevance, Coherence, Clarity), for a total of 6 criteria.
Be sure to:
1. Think step-by-step about why each criterion is relevant to the query.
2. Think step-by-step through the query and how each criterion applies.
3. Provide a brief analysis for each selected criterion on how it applies to the query.
4. Integrate the above reasoning into the Definition and Standards sections of each criterion.
{format_query}

prompt

Query: {query}
Result: <start> {clean_res} <end>
Evaluation Standard: {json.dumps(evaluate_standard, ensure_ascii=False)}
Based on the provided info, perform a rigorous evaluation. {format_eval}

A.5 Evaluation Prompt for Win-Rate Judgment

SYSTEM_PROMPT for GPT-4.1 Win-Rate Evaluation

Please act as an impartial judge and evaluate the quality of the written responses provided by two AI assistants to the user's writing prompt below. You will be given Assistant A's response and Assistant B's response. Your job is to determine which assistant's writing is superior.
Evaluate them on the following criteria:
1. Relevance and Completeness: Does the assistant fully respond to the writing prompt? Does the length meet the user's query expectations? Is the content relevant to the topic, and does it provide sufficient depth, length, and detail, rather than drifting off-topic or simplistic?
2. Writing Quality: Evaluate whether the assistant's writing is clear, fluent, and free of obvious grammatical errors. The overall quality of the writing is high, with elegant.
3. Creativity and Originality: If applicable, assess the creativity of the response. Does the assistant offer fresh perspectives, unique insights, or demonstrate a certain level of originality?
4. Specificity and Detail: Determine whether the assistant provides concrete examples or detailed explanations. Properly justified repetition is permissible.
5. Tone and Style: Is the tone appropriate for the writing prompt? Is the writing style consistent throughout? Consider whether it aligns with the expectations of the intended audience or writing purpose.

After evaluating each response, determine which one is superior based on the factors above. Provide your explanation and then select one of the following final verdicts:

- Assistant A is significantly better: [[A»B]]
- Assistant A is slightly better: [[A>B]]
- Tie, relatively the same: [[A=B]]
- Assistant B is slightly better: [[B>A]]
- Assistant B is significantly better: [[B»A]]

Example output: My final verdict is tie: [[A=B]].

Annotation Details. We hired 3 human annotators (2 female, 1 male), all with undergraduate education or above, to perform manual evaluation. Each instance was labeled by all three annotators, and final labels were determined via majority voting. Annotators were compensated at a rate of 300 RMB per day.

A.6 Case Study

Research on the Impact of Digital Inclusive Finance on Household Financial Asset Allocation

Query: Research on the Impact of Digital Inclusive Finance on Household Financial Asset Allocation.

***SuperWriter*-LM Response:** Digital inclusive finance (DIF) has become a cornerstone of global financial innovation, transforming access to financial services for millions. Over 70% of Sub-Saharan Africa now uses mobile money platforms like Kenya's M-Pesa, while India's Digital India initiative has spurred a 40% rise in crypto adoption since 2020. These advancements highlight DIF's transformative potential, yet existing research has fallen short in analyzing three critical dimensions: the mechanisms driving shifts in household asset allocation, the equity trade-offs between access and exclusion, and the risk mitigation required to balance innovation with stability. Most studies focus on access expansion or behavioral finance theories in isolation, neglecting how DIF's design—such as gamified investing apps or frictionless cross-border transfers—affects portfolio composition, risk tolerance, or wealth gaps. Methodological gaps persist too: cross-

sectional data often conflate correlation with causation, and equity analyses rarely account for cultural or infrastructural moderators like trust in institutions or gender norms. This study addresses these gaps by rigorously examining how DIF influences household financial asset allocation, including diversification, risk-return preferences, and equity access. Grounded in behavioral finance (e.g., FUDT), financial inclusion theory, and institutional economics, it employs a mixed-methods approach combining quantitative surveys (n=10,000 households in Nigeria and Indonesia), qualitative interviews with users and regulators, and ethnographic observations. Grounded in these theories, the study's five hypotheses guide the analysis: (1) DIF adoption increases formal market participation, (2) behavioral nudges shift portfolios toward riskier assets, (3) DIF reduces wealth gaps for underserved groups, (4) geographic/cultural factors moderate these effects, and (5) DIF amplifies exposure to volatile/high-risk products. By bridging interdisciplinary insights—from behavioral biases to institutional trust—the study offers actionable strategies for policymakers and fintech firms to design equitable, risk-aware systems. Its findings aim to inform global debates on how to harness DIF's potential while mitigating its pitfalls, ensuring financial innovation serves as a ladder out of poverty rather than a trap into risk.
Empirical studies have highlighted DIF's role in expanding financial access. For instance, Kenya's M-Pesa reduced informal savings reliance by 20% among rural households (Ratha, 2012), while India's Digital India initiative spurred a 40% rise in crypto adoption since 2020 (Reserve Bank of India, 2023). These findings underscore DIF's potential to bridge wealth gaps and democratize financial inclusion. However, such studies focus predominantly on access metrics—such as account ownership or transaction frequency—rather than how this access translates to **changes in household financial behavior**, such as asset allocation or risk tolerance. By including unregulated peer-to-peer platforms in DIF metrics, prior studies may overstate the service's equity impact, as informal tools often lack safeguards against predatory practices or volatility. This conflation also masks how DIF's design—such as gamified apps or cross-border transfers—affects risk tolerance or wealth gaps, limiting insights into its dual role as both an equity tool and a risk amplifier. This focus on access metrics neglects the mechanisms through which DIF alters household financial behavior—such as shifts in asset allocation or risk tolerance—and fails to assess how these changes interact with equity trade-offs or systemic risks, such as volatility exposure.
To address these dual challenges of equity and risk, theoretical frameworks like Financial Decision-Making Under Uncertainty (FUDT) and Modern Portfolio Theory (MPT) provide critical lenses for understanding how DIF's design shapes household financial behavior.....)

艾琳达民俗志

Query: 请帮助写出一个有趣的故事，2000字左右。开头是在遥远平行世界的一角，你，作为一名异世界的民俗学家，踏上了一片充满异域风情的土地。

***SuperWriter*-LM Response:** 实验室的白炽灯在《艾琳达民俗志》泛黄的书页上投下斑驳光影。林小夏的指尖悬在笔记本边缘，激光笔的光斑正沿着“岩灵祭祀需三色岩盐”的注释游移。突然，那道光斑在纸面某处骤然凝固，符号边缘泛起幽蓝荧光——这光芒并非单纯视觉现象，而是《艾琳达民俗志》中记载的“裂隙激活符”——当三色岩盐的符号被激光笔的光谱激发时，会形成跨维度的能量通道。符号边缘的蓝光在纸面形成类似DNA链的缠绕结构，每个节点都与冰晶裂纹的走向完全重合。
“奇怪.....”她轻声呢喃，凑近观察时，书页竟开始自行翻动。那些发光的萤火藤实为岩灵祭祀的灵脉指示器——当能量通道开启时，被封印的祭祀参与者会以虚影形式显现。倒影在符号中的冰封峡谷骤然清晰，冰面下封印着艾琳达大陆的岩灵祭祀残卷。雾气中浮现出雾之族少女的轮廓。竹哨的呜咽声混着实验室的电子嗡鸣，那频率与《艾琳达民俗志》第17章记载的“岩灵共鸣频率”完全吻合。林小夏的耳膜传来刺痛——这声音的波长恰好是三色岩盐在特定温度下共振的频率。她下意识握紧父亲遗留的铜牌，铜牌表面的纹路正

与冰晶共鸣。指尖突然发烫，她想起父亲临终前将一枚刻有三色岩盐纹的铜牌塞进她手心——那枚铜牌此刻正贴着掌心，与冰晶共鸣的频率完全一致。
“等等。”她后退半步，符号光芒暴涨，将整张书桌吞没。当视线再次聚焦时，她握着的笔记本已变成一片冰晶，而窗外的竹林深处，传来与童年祭祖时如出一辙的竹哨声。实验室的通风管道开始渗出细小的雾气，凝结成与冰晶相同的六棱结晶。这些雾气在地面形成微小的漩涡，如同被撕裂的时空裂缝在现实世界投下的投影。
冰凉的触感爬上后颈——那不是汗珠，而是雾之族少女的雾气渗入实验室的证明。林小夏低头，发现所有笔记都被双螺旋符号覆盖，唯有窗台残留着竹哨的余韵，与三百年前父亲唤醒祖先灵位的调子完美重合。那些符号并非静态图案，而是以书页为基底不断重组——就像艾琳达大陆的岩灵祭祀在重演。 雾气从莉娅指尖溢出时，林小夏闻到了竹子的清香。少女的皮肤如流动的云雾，随着她向前走动，那些雾气在冰面下浮现出与《艾琳达民俗志》第12章记载的"岩灵印记"相同的螺旋纹路——这是雾之族萨满在举行祭祀时，通过雾气凝结形成的古老符号。
"您父亲用的竹哨，与三百年前萨满唤醒土地的仪式完全一致。"莉娅的声音裹着雾气，如同三百年前岩灵祭祀的咒文："信仰不是跪拜的姿态，而是让岩盐结晶时，能听见土地在冰层下低语的耳朵。"她跪坐在冰封的溪流旁，指尖划过冰面，泛起的涟漪竟让整片区域的雾气凝结成冰晶。
林小夏盯着那些冰晶中的纹样，突然意识到它们与《艾琳达民俗志》中"季节仪式"的记载完全重合——那不是魔法符号，而是岩灵祭祀在冰层下刻下的伤疤。她举起激光笔对准冰面，试图用光斑测量裂纹角度："这不过是自然魔法的伪科学现象。仪式失败的原因，是你们的信仰不够纯粹。"
"信仰？"莉娅的雾气突然收缩成剑锋形状，皮肤在雾气收缩时泛起蛛网状裂纹，如同被暴雨击碎的瓷器。她的声音混着雾气的震颤："您父亲用竹哨唤醒祖先灵位时，是否听见了土地的低语？"
林小夏的指尖猛地刺痛——那不是幻觉。冰面下传来竹哨的呜咽，混着雾气的震颤，那声音与父亲临终前对她说的"有些伤痕需要文化之光来治愈"重叠。父亲站在竹林深处的画面突然在冰晶中闪现：他闭着眼睛，让竹哨声与山风共鸣，而她却总在反驳"竹哨只是乐器"。
"爸爸，竹哨只是竹子做的乐器……"八岁的林小夏曾这样反驳。父亲却闭上眼睛，让竹哨声与山风共鸣："林小夏，听，土地在说话。"
"你父亲的竹哨，"莉娅的声音突然尖锐如岩浆："唤醒了不该被惊醒的沉睡。仪式的火焰需要木柴，但你的火种早已枯萎——三色岩盐在你手中，不过是实验室的化学试剂。"她的雾气化作冰针刺入溪流，冰面下浮现的岩盐结晶在蓝光中闪烁。
……
林小夏握紧被莉娅血脉之力染蓝的玉璧，听见窗外传来竹哨声——那声音既熟悉又陌生，如同被时光折叠的回响，正穿透实验室的玻璃，唤醒另一个维度的古老契约。

A.7 Predicting User's Length Requirement with DeepSeek-R1-distill-Qwen-32B

We describe a method for predicting the length of a user's input requirement using the R1-distill-Qwen-32B model for few-shot learning. The process involves two main steps: predicting whether the input exceeds 2,000 words, and predicting the exact length requirement based on the first prediction.

1. **Step 1: Predicting Length Exceedance (Prompt 1)**: The first prediction is made by checking whether the input exceeds 2,000 words. A carefully crafted prompt (Prompt 1) is provided to the model to predict if the content's expected word count will surpass the 2K threshold. The model utilizes few-shot learning with example inputs to classify the task into either "above 2K" or "below 2K" based on the nature of the input.
2. **Step 2: Predicting Exact Length Requirement (Prompt 2)**: Once the model predicts whether the task exceeds 2,000 words, a second prediction is made to determine the exact length category. Based on the result from Step 1, Prompt 2 is designed to predict whether the content is in the 2K-4K, 4K-8K, 8K-16K, or 16K+ category. The model provides the final prediction by analyzing the contextual hints and the input length characteristics.

Prompt-1

Guidelines:
To determine whether the expected output will exceed 2000 words, consider the following factors:

1. **Depth and Complexity:** Does the task require detailed explanations, in-depth analysis, or comprehensive coverage of complex topics?
2. **Scope and Breadth:** Does the task cover multiple subtopics, perspectives, or extensive subject matter?
3. **Structure and Sections:** Does the output need to include multiple sections such as introductions, literature reviews, methodologies, results, discussions, and conclusions?
4. **Research and References:** Does the task require extensive research, citations, and referencing of multiple sources?

Response Format:

- Answer with either "#*# Yes" or "#*# No".
- Provide a concise justification based on the guidelines above.

Example 1:
Query: Is Sanskrit the oldest language?
Answer: This question requires a concise factual answer, not an extensive output. #*# No *** **END**

Example 2:
Query: Create a detailed business plan for a new cat litter product.
Answer: Creating a detailed business plan involves multiple sections such as market research, product development, financial projections, marketing strategy, and competitive analysis, all of which require in-depth exploration and explanation. #*# Yes *** **END**

.....
.....

Assess the following statement and decide whether the expected response is likely to require more than 2000 words. Answer with either "#*# Yes" or "#*# No," and include a brief justification, like above example.
Query: **User Query**
Answer:

Prompt-2

Guidelines:
To estimate the expected length of the output, consider the following factors:

1. **Depth and Complexity:** Does the task require detailed explanations, in-depth analysis, or complex reasoning?
2. **Scope and Breadth:** Does the task cover multiple subtopics, perspectives, or an extensive subject matter?
3. **Structure and Sections:** Does the output require multiple sections (e.g., introduction, literature review, methodologies, results, discussions, conclusions)?
4. **Research and References:** Does the task require significant research, citations, or references to multiple sources?
5. **Detail Level:** Is the task expected to be highly detailed, or can it be summarized concisely?

Response Format:
- Choose the most likely word count category: "Less than 2000 words", "2000 words", "4000 words", "8000 words", or "16000 words". Using (### Category: "Chosen category") as the response format.
- Provide a brief justification based on the guidelines above.

Example 1: Less than 2000 words
Query: Is Sanskrit the oldest language?
Answer: This is a factual question requiring a brief answer with no complex analysis or subtopics. Likely to be less than 2000 words. ### Category: Less than 2000 words
Explanation: Similar to a short blog post or brief news article, this task needs minimal detail and is concise. *** **END**

Example 2: 2000 words (2000 to 4000 words)
Query: Describe the key differences between classical and quantum computing.
Answer: This question requires moderate detail, comparing classical and quantum computing without exhaustive technical exploration. Likely to be around 2000 words. ### Category: 2000 words
Explanation: Similar to a moderate-length essay or a detailed blog post, this task covers key points with enough depth but remains manageable. *** **END**

.....
.....
Example 5: More than 16000 words
Query: Write a full-length book on the history of the Industrial Revolution, covering all major events, technological innovations, and global impacts.
Answer: This would require an in-depth exploration of the entire history of the Industrial Revolution, with detailed analysis across multiple chapters. Likely to be more than 16000 words. ### Category: 16000 words
Explanation: Similar to a book-length content, such as a thesis or encyclopedia entry, requiring substantial detail and coverage over multiple sections or chapters. *** **END**
Assess the following statement and decide what the expected output length is. Answer with the appropriate word count category and provide a brief justification.
Query: **User Query**
Answer: