One Agent to Serve All: a Lite-Adaptive Stylized AI Assistant for Millions of Multi-Style Official Accounts

Xingyu Fan, Feifei Li, Wenhui Que*, Hailong Li
WeChat, Tencent Inc., Beijing, China
{fanxfan, niyali, victorque, aloneli}@tencent.com

Abstract

Conversational agents deployed in industrial-scale WeChat Official Account platforms must generate responses that are both contextually grounded and stylistically aligned—requirements that existing methods struggle to meet. Chain-of-thought (CoT) prompting induces significant latency due to multi-turn reasoning; per-account fine-tuning is computationally prohibitive; and long prompt-based methods degrade the model's ability to grasp injected context and style. In this paper, we propose WeStar, a lite-adaptive framework for stylized contextual question answering that scales to millions of WeChat Official Accounts. Our contributions are fourfold: (1) We introduce WeStar, a unified framework capable of serving large volumes of official accounts with minimal overhead. (2) We propose a multi-dimensional, cluster-based parameter sharing scheme that enables compact style representation while preserving stylistic diversity. (3) We develop a style-enhanced Direct Preference Optimization (SeDPO) method to optimize each style cluster's parameters for improved generation quality. (4) Experiments on a large-scale industrial dataset validate the effectiveness and efficiency of WeStar, underscoring its pracitical value in real-world deployment.

1 Introduction

Inspired by the outstanding capabilities of large language models in question-answering tasks, the industry has adopted conversational agents that dialogue with users in many tasks, such as AI in games (van Stegeren and Mysliwiec, 2021), voice assistants (Arakawa et al., 2024), and WeChat Official Account assistants. WeChat Official Accounts serve as a powerful communication channel for individuals, media outlets, enterprises, government bodies, and other organizations, enabling them to disseminate information in article form within our ecosystem. Users can interact with these articles in WeChat Official Accounts by leaving comments, which are often replied to directly by the author. Additionally, users may pose questions to WeChat Official Account's intelligent assistant via the chat interface, expecting responses that are both context-grounded and style-aware which are grounded in the author's published articles and reflective of the author's personal communication style. While the articles provide rich, factual content, they are typically formal and do not capture the author's conversational tone. In contrast, the author's replies to user comments offer a more authentic and fine-grained reflection of their stylistic preferences in interactive settings. Motivated by this observation, we treat articles as the source of question-specific knowledge and leverage the author's historical comment replies as the basis for style-specific knowledge to address the **Stylized Contextual Question Answering** problem.

To tackle the problem, existing approaches can be broadly categorized into three paradigms: **fine-tuning-based** methods, which directly adapt the model on style-specific data; **CoT-based** methods, which employ multi-step prompting to decompose and solve the task; and **prompt-based** methods, which inject both knowledge-specific and style-specific knowledge into a single prompt. However, each of these comes with limitations in scalability, efficiency, or effectiveness when deployed at industrial scale.

In recent years, fine-tuning-based approaches have shown strong performance in the domain of stylized text generation (Liu et al., 2024). A widely adopted strategy involves supervised fine-tuning (SFT) on customized style-specific corpora, enabling large language models (LLMs) to adjust their output distributions by updating model parameters accordingly. However, fine-tuning remains a significant bottleneck in many real-world appli-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 14258–14275
July 2-7, 2026

cations. For instance, in WeChat Official Account assistant tasks, a distinct model must be fine-tuned and maintained for each individual author to ensure stylistic consistency. This process incurs substantial time and computational costs, severely limiting scalability.

Chain-of-thought (CoT)-based methods offer a promising direction for mitigating the computational and deployment burden associated with stylized text generation. A straightforward solution is to decompose the stylized contextual question answering task into two sequential sub-tasks: (1) generating a contextually relevant answer, followed by (2) applying a text style transfer model to adapt the response to the target style. While conceptually simple, this two-stage pipeline introduces practical limitations. Specifically, invoking a large language model (LLM) twice not only increases computational overhead but also introduces latency, which can significantly degrade the user experience.

Therefore, prompt-based methods presents a more practical solution to the stylized contextual question answering task. By incorporating retrieved question-specific articles and customized style-specific corpora into the input of an end-to-end LLM, the system injects external stylistic and semantic knowledge directly into the model's context window, enabling dynamic adaptation without parameter updates. However, injecting knowledge from multiple sources via input prompts inevitably leads to increased context length. This extended input not only introduces additional computational overhead and latency during inference, but also degrades the LLM's ability to effectively understand and utilize the injected information (Levy et al., 2024).

To address these challenges, we propose **WeStar**, a novel framework to build one lite-adaptive style-ware and context-grounded agent capable of serving millions of multi-style WeChat Official Accounts. Before online inference, WeStar first performs fine-grained style labeling over each account author's corpus across multiple stylistic dimensions. Authors with similar style are then grouped into clusters, and each cluster is associated with a shared set of stylized model parameters trained via style-enhanced Direct Preference Optimization (SeDPO). This cluster-based parameter sharing enables compact storage of stylistic knowledge and supports scalable deployment across millions of account authors. During inference, WeStar incorporates question-specific knowledge (i.e., retrieved articles) into the input prompt to enrich the model's domain understanding and enhance its question-answering capabilities. In parallel, instead of relying exclusively on prompt-based knowledge injection, WeStar adopts a parameter injection approach, where style-specific knowledge is embedded directly into the model parameters. This dual-channel design not only enhances stylistic consistency but also substantially reduces prompt length, thereby mitigating context overflow and improving inference efficiency.

To evaluate the effectiveness of WeStar in real-world industrial scenarios, we conducted experiments on a large-scale WeChat official account dataset. WeStar achieves state-of-the-art performance across four customized evaluation dimensions, including contextual alignment, question relevance, stylistic strength and fluency. These results demonstrate the practical applicability of our framework in stylized contextual generation tasks.

Our main contributions are summarized as follows:

- We propose WeStar, a novel framework to build one lite-adaptive stylized contextual question-answering agent capable of serving millions of WeChat Official Accounts.
- We introduce a multi-dimensional style-specific cluster-based parameter sharing approach that enables compact parameter storage while preserving a broad spectrum of stylistic knowledge, facilitating scalable deployment across diverse WeChat official account authors.
- We leverage a style-enhanced Direct Preference Optimization (SeDPO) strategy to train the parameter representations for each style cluster, thereby improving the model's capacity for style-aware generation.
- We conduct evaluations on a large-scale industrial dataset and validate the proposed method across four key metrics, demonstrating its practical value and profitability in real-world applications.

2 Related Work

2.1 Text Style Transfer

Text style transfer (TST) is a different but related task compared to our work, which aims to alter the stylistic attributes of a given text while preserving its original content. Early research focused on rule-based and statistical methods. With the rise of deep learning, neural methods became dominant, particularly unsupervised frameworks leveraging content-style disentanglement or reinforcement learning (Liu et al., 2022; Luo et al., 2019). More recent work investigates the potential of contrastive learning and pattern mining (Han et al., 2023). LLMs also bring new paradigms to TST. Reif et al.(Reif et al., 2022) and Mukherjee et al.(Mukherjee et al., 2024) explored zero-shot and few-shot style transfer via prompt engineering, revealing LLMs' surprising generalization ability. Ostheimer et al. (Ostheimer et al., 2024) further investigated LLM-based evaluation, demonstrating strong correlation with human judgments.

2.2 Stylized Answer Generation

Several studies tackle stylized answer generation using fine-tuned large language models or style-controlled decoding strategies (Zheng et al., 2021; Gao et al., 2019). To better ground stylized generation in external knowledge, Sun et al.(Sun et al., 2022) introduce disentangled template rewriting in knowledge-grounded scenarios. Additionally, Feature-Guided Knowledge Augmentation(Li et al., 2023) retrieves stylistic sentences to guide content planning and uses contrastive learning to enhance fluency and style control. Despite these advances, most prior work operates on relatively small-scale or synthetic style corpora, which limits their applicability in real-world, large-scale scenarios. Recent efforts have begun to address scalability (Li et al., 2024a; Jing et al., 2024), but none match the scale and complexity of the deployment setting considered in our work.

3 Method

Given a user-issued question Q, a retrieved context-specific knowledge C relevant to Q, and a retrieved style-specific knowledge S representing a target response style, the objective is to generate a style-aware and context-grounded answer A such that A correctly addresses the information need expressed in Q, grounded in the content of C, and A adheres to the stylistic characteristics exemplified by S.

To address the challenges of stylized contextual question answering, WeStar adopts a dual-injection approach: it injects question-specific knowledge into the prompt and style-specific knowledge into the model parameters. In this section, we first describe how WeStar clusters authors with similar style and trains shared stylize-specific parameters for each style cluster. We then present the online inference mechanism of WeStar, which dynamically composes responses by combining the retrieved content and the appropriate style-specific parameters.

3.1 CQA Construction

To construct high-quality CQA (*Context, Question, Answer*) triplets for training, we adopt two complementary strategies: a forward-thinking method that prompts a large language model M to generate questions and answers based on given article segments, and a bottom-up method that prompts M to simulate realistic user roles and queries based on the domain of each public account, followed by context retrieval and answer generation with M. The forward-thinking approach offers high scalability, while the bottom-up strategy introduces domain-relevant and user-intent-aligned queries, ensuring a better match to real-world QA workflows. Together, these methods yield a CQA dataset that balances diversity, difficulty, and contextual grounding. Full prompt templates and examples are provided in the **Appendix**.

3.2 Style Labeling

We design twelve style classification standards, organized across four stylistic dimensions: (1)**Semantic level:** intention type, degree of authority; (2)**Grammatical level:** omitted features, use of inversion, use of passive voice; (3)**Syntactic level:** sentence complexity, rhetorical features, cohesion mechanisms; (4)**Lexical level:** lexical complexity, emotional polarity, frequency of emojis, and degree of formality.

For each official account author, we perform fine-grained labeling of their style corpus using LLM M, based on the twelve predefined classification standards. Specifically, for every QA pair in the corpus, M generates candidate labels for each stylistic dimension. We then aggregate these annotations by taking the majority label for each standard across all QA pairs within the author's corpus. These aggregated labels serve as the author's stylistic profile and provide the foundation for subsequent style tree building. The detailed prompt templates are provided in the **Appendix**.

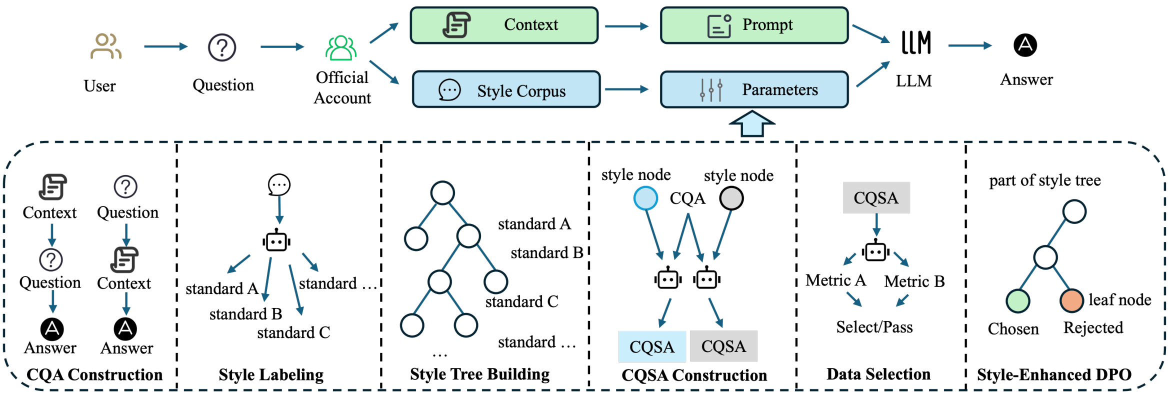

Figure 1: Overview of WeStar.

Algorithm 1 style tree building

Input: $\mathcal{C}$ – a set of style corpora
Input: $\mathcal{S}$ – a set of style classification standards
Output: $\mathcal{T}$ – a hierarchical style tree

Initialize tree $\mathcal{T}$ with a root node containing all corpora $\mathcal{C}$
Initialize queue $Q \leftarrow$ {root node}
Set corpus size threshold k
For each style standard $s \in \mathcal{S}$:
  While Q is not empty:
    $n \leftarrow Q.front()$
    $Q.pop()$
    If node n can be split into child nodes $\{n_1, ..., n_m\}$ by s **and** each child n_i satisfies $|\mathcal{C}_{n_i}| > k$, **then**:
      Expand node n in $\mathcal{T}$ into children $\{n_1, ..., n_m\}$
  Push all new leaf nodes in $\mathcal{T}$ into queue Q
Return $\mathcal{T}$

3.3 Style Tree Building

Following the style labeling stage, we construct a hierarchical style clustering tree to group the authors with similar style into the same cluster. The construction process, outlined in Algorithm 1, proceeds by traversing the predefined set of style labeling standards $\mathcal{S}$ in a specified hierarchical order. For each leaf node in the current tree, if the associated style corpus can be partitioned by a label standard $s \in \mathcal{S}$—and each resulting subset contains more than k examples—then the node is expanded into multiple child nodes, each corresponding to a distinct stylistic subgroup defined by s. Upon completion of the algorithm, each leaf node in the resulting style clustering tree represents a stylized cluster of the original corpus $\mathcal{C}$. The path from a given leaf node to the root captures its cumulative stylistic characteristics.

In this way, authors with similar stylistic characteristics are grouped into the same clusters. This hierarchical organization facilitates style-specific parameter training by enabling parameter sharing among authors within the same cluster, thus reducing both training costs and storage overhead. Moreover, for authors with limited stylistic data, this clustering mechanism allows them to benefit from the shared parameters of stylistically similar authors, thereby enhancing generalization and mitigating the data scarcity issue.

3.4 Data Construction

Many recent studies have demonstrated the strong capabilities of LLMs in text style transfer(TST) (Mukherjee et al., 2024). Building on this foundation, we leverage LLMs to transform standard CQA instances into CQSA *(Context, Question, Stylized Answer)* instances that align with the target style for each cluster.

For each corpus within the style cluster derived from the style tree, we prompt LLM M to rewrite the original answer—while preserving the factual correctness of the response—into a form that conforms to the style of the target corpus. The detailed prompts are presented in the **Appendix**.

Inspired by metric-based RL (Wang et al., 2025), we apply metric-based constraints during data construction to better align the model's outputs with the target stylistic expectations, while simultaneously reducing the likelihood of hallucinations. Following previous works (Wang et al., 2025; Ostheimer et al., 2024), we conduct automated evaluation on each CQSA instance using LLM M across four key dimensions: Contextual Alignment (**C–A**), Question Relevance (**Q–A**), Stylistic Strength (**S–A**) and **Fluency**, which is presented in the **Appendix**. We will further discuss these metrics in Section **Metrics**. We aggregate the scores across these four metrics and select the top 10,000 CQSA instances as high-quality samples for subsequent style-specific parameter training.

3.5 Style-Enhanced DPO

Prior to this stage, we have obtained high-quality CQSA instances for each style cluster to support style-specific parameter training. In this step, we adopt style-enhanced DPO (SeDPO) to train the parameters. Specifically, for a given style cluster, the top 10,000 CQSA instances are used as chosen samples. We construct the corresponding rejected sample for each chosen sample by retrieving answers to the same question with high stylistic similarity—such as sibling nodes in the style tree—while ensuring that these answers differ in a certain style label. This construction aligns with the principle of controlled variable experimentation: when the negative sample shares most contextual and semantic features with the positive sample, the model is encouraged to focus more on the fine-grained stylistic distinctions. Consequently, this training setup facilitates more effective learning of style-specific behaviors within each stylistic cluster.

We employ LoRA (Xu et al., 2024) as our fine-tuning and parameter storage strategy. This design enables each style cluster to be associated with an independently trained set of low-rank adaptation parameters, which allows the model to encode style-specific behaviors in a parameter-efficient manner, enabling scalable and flexible deployment across a wide range of stylistic clusters without the need to train or deploy the full base model.

3.6 Online Inference

After training a set of style-specific parameters for each style cluster, WeStar applies them during online inference to deliver style-aware, context-grounded responses at scale.

WeStar performs inference by jointly injecting question-specific and style-specific knowledge into the generation process. Specifically, question-specific article segments are inserted into the input prompt, providing contextual grounding. Simultaneously, style-specific LoRA parameters corresponding to the author's style cluster are retrieved and injected into the model's parameter space. This dual-injection strategy enables WeStar to generate responses that are both contextually relevant and stylistically aligned, while maintaining scalability across a large number of public account authors.

4 Experiments

4.1 Experimental Settings

4.1.1 Dataset

To the best of our knowledge, there exists no public dataset that simultaneously provides articles, user queries, and large-scale stylized replies from millions of authors. Therefore, we directly evaluate our method using proprietary data from a widely-used, real-world industrial official accounts platform, which contains multiple language corpora such as Chinese, English, Japanese, etc. We deployed the WeStar framework in this environment and conducted evaluations across ten representative style clusters, constructed using the methodology detailed in Section **Style Tree Building**. Each cluster contains real user comments and the corresponding author replies collected before July 2025, serving as style reference for stylized question answering. The contextual retrieval corpus consists of all historical articles published by the authors prior to the same date. To perform evaluation, we constructed a test set of 2,000 instances using the same methodology described in Section **CQA Construction** and Section **CQSA Construction**. To better simulate real-world deployment scenarios, we further supplemented the test set with 3,000 user-generated information-seeking questions collected from live interactions. In total, this dataset comprises 5000 queries and will be further open-sourced for the community.

4.1.2 Metrics

To ensure consistency and comparability, we employ the same four evaluation metrics as described in Section **Data Selection** to assess model performance. These metrics have been widely adopted in previous studies (Ostheimer et al., 2024; Wang et al., 2025): (1)**Contextual Alignment (C–A)**: the degree of semantic consistency between the generated answer and the retrieved context; (2)**Question Relevance (Q–A)**: the extent to which the answer accurately addresses the core intent of the input question; (3)**Stylistic Strength (S–A)**: the degree to which the answer adheres to the target stylistic attributes; (4)**Fluency**: the grammaticality and naturalness of the generated response.

To obtain consistent and scalable evaluations, we employed DeepSeek-R1 (DeepSeek-AI et al., 2025) to rate each output along the four dimensions, using standardized evaluation prompts. The exact prompt formulations are provided in the **Appendix**. In case of human evaluations, the annotators are instructed to use the same criteria, as presented in the **Appendix**.

4.1.3 Baselines

As discussed in the **Introduction**, stylized contextual question answering in industrial settings—such as official account platforms—requires a fully end-to-end LLM pipeline. This constraint renders multi-step or sequential prompting approaches infeasible due to latency and system complexity. To the best of our knowledge, there exists no prior method capable of supporting scalable stylized contextual question answering across millions of official account authors with distinct stylistic preferences.

To evaluate the effectiveness of WeStar under this challenging setting, we compare it against five baseline methods, covering diverse methodological paradigms: (1) two prompting-based approaches, (2) two SFT variants from WeStar, and (3) one variant of DPO:

R1-Prompt. This baseline uses DeepSeek-R1 (DeepSeek-AI et al., 2025). We construct the prompt by combining the user's question, retrieved articles, style-exemplar author replies, and a system instruction, enabling the model to generate a context-grounded and style-aware answer. It represents the prompt-based paradigm, relying solely on prompt engineering without fine-tuning.

SFT-Prompt(StyleChat). We leverage the state-of-the-art technique StyleChat (Li et al., 2024b) as the second baseline, named SFT-Prompt in this paper, based on the Qwen3-32B model (Yang et al., 2025). This setting represents the **SFT-then-prompting paradigm** under similar model scale constraints and serves as a strong baseline for evaluating methods that enhance LLMs via lightweight supervised adaptation followed by prompting.

LoRA-SFT. Performs LoRA-based SFT on randomly sampled CQSA per cluster.

LoRA-SFT-S. Same as LoRA-SFT but uses top-ranked CQSA from metric-based selection.

WeStar$_{MDPO}$. Replaces our SeDPO with metric-guided DPO (Wang et al., 2025) using base LLM distillates as rejected samples.

4.1.4 Implementation Details

We adopt Qwen3-32B as the base language model. Additionally DeepSeek-R1 is chosen as the the auxiliary LLM M throughout our framework. During the **CQA Construction** phase, we prompt M to generate **3** representative user roles for each official account domain, and subsequently generate **3** domain-relevant questions per role to construct a diverse question set. In the **Style Tree Building** stage, we set the corpus size threshold **k=100** to ensure sufficient stylistic representation in each node before further partitioning. For **Style-enhanced DPO**, we employ LoRA with a rank of 16 and train for one epoch, enabling efficient parameter adaptation for each style corpus while maintaining training scalability.

4.2 Main Results

4.2.1 WeStar vs. Prompt-Based Methods

Figure 2 illustrates the comparative performance between WeStar and prompt-based methods across all four evaluation metrics mentioned above. Specifically, the top-left, top-right, bottom-left, and bottom-right subfigures respectively show the per-cluster comparison results between WeStar, R1-Prompt and SFT-Prompt along each metric. The x-axis represents the identifiers of different style corpus clusters, while the y-axis corresponds to the metric scores achieved by each method on those clusters. More detailed numerical results are presented in Table 1. The first column lists the style corpus clusters and the associated evaluation metrics. Columns 2, 3, and 7 report the performance of R1-Prompt, SFT-Prompt, and WeStar, respectively.

From the results, we observe that WeStar consistently outperforms both prompting-based approaches on Q–A, C–A , and Fluency metrics on average. This improvement stems from the limitations of prompt-based methods when dealing with long input sequences. These approaches inject both the retrieved articles and the full style corpus into the prompt, leading to significant context length expansion, which in turn challenges the LLM's attention window. As a result, the model's ability to accurately comprehend and utilize the injected information is weakened. For the Stylistic Strength (S–A) metric, only R1-Prompt achieves performance comparable to WeStar on average. We attribute this to two main reasons: (1) Prompt-based methods can flexibly incorporate the original author's entire style corpus directly into the prompt, whereas WeStar utilizes corpora from style-similar authors identified via the style tree; (2) The base model used in R1-Prompt, DeepSeek-R1, contains significantly more parameters than our base model,

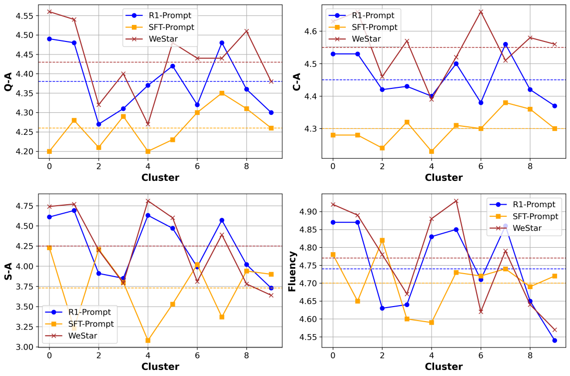

Figure 2: Results of WeStar vs. Prompt-Based Methods

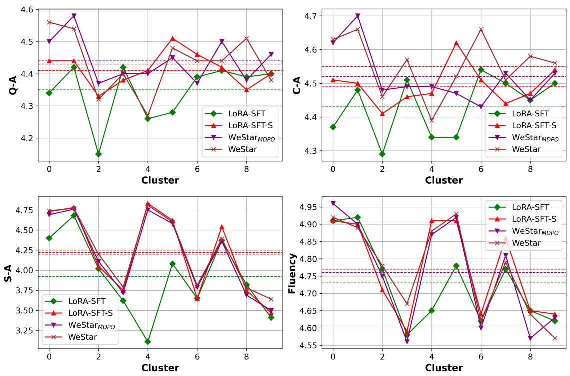

Figure 3: Results of WeStar vs. Variants of WeStar

Qwen3-32B, which may contribute to its stronger stylistic generation performance. Consequently, WeStar matches but does not surpass R1-Prompt on the S–A dimension.

4.2.2 WeStar vs. Variants of WeStar

Figure 3 presents a comparative analysis of WeStar and its three training variants: LoRA-SFT, LoRA-SFT-S, and WeStar$_{MDPO}$—across all four evaluation metrics. The detailed results can be found in Table 1, where the 4th to 7th columns correspond to the respective performances of LoRA-SFT, LoRA-SFT-S, WeStar$_{MDPO}$, and WeStar.

Among these, LoRA-SFT underperforms the other three methods across all metrics on average. This is primarily due to the lack of metric-based data filtering in its training phase. In contrast, the remaining three approaches benefit from the metric-based data selection pipeline, which ensures higher-quality CQSA instances and leads to more accurate style-aware and context-grounded responses.

Overall, WeStar achieves the best average performance, outperforming both LoRA-SFT-S and WeStar$_{MDPO}$, except on the Q–A metric, where WeStar$_{MDPO}$ leads by a small margin of 0.01. Across all four metrics, the performance differences among these three approaches remain relatively small (average gap is smaller than 0.06), indicating that data quality plays a major role, while training objectives provide more fine-grained benefits.

Notably, WeStar achieves the highest score on the S–A (Stylistic Strength) metric, validating the effectiveness of using style-specific rejected samples during DPO training. In contrast, WeStar$_{MDPO}$ shows slightly inferior performance on S–A. A likely explanation is that its rejected samples differ significantly from the chosen ones across all four dimensions, allowing the model to easily distinguish them and optimize logits without focusing specifically on stylistic nuances. This diluted objective may have weakened the model's ability to capture fine-grained stylistic preferences.

4.3 Online A/B Test

To validate real-world impact, we deployed WeStar (with SeDPO) against WeStar$_{MDPO}$ in a live production environment over 14 consecutive days, involving millions of active users. The results are statistically significant at $p < 0.01$: WeStar achieves a +0.42% lift in Daily Active Users (DAU) and a +2.62% lift in average dialogue turns per user. Although the offline metric gap between SeDPO and MDPO appears modest (0.05 avg), these online gains translate to millions of additional user interactions, demonstrating that the stylistic fidelity improvement from SeDPO yields substantial engagement benefits at industrial scale.

4.4 Time Cost Analysis

We focus our runtime comparison on SFT-Prompt and WeStar, as SFT-Prompt employs a comparable backbone model. The average end-to-end inference time per sample is 2.08s for WeStar vs. 2.47s for SFT-Prompt, yielding a 1.19× speedup. Table 2 provides a per-component breakdown of WeStar's pipeline latency.

The bottleneck lies in LLM inference (82%), while style tree lookup and LoRA loading introduce negligible overhead (0.01s and 0.15s respectively). In contrast, SFT-Prompt's longer input length—caused by injecting style-related tokens into the prompt—imposes substantial decoding overhead. This confirms the efficiency of parameterized style injection over prompt-based alternatives.

Table 1: Main Results

	Prompt-Based Methods		**SFT-Based Methods**		**DPO-Based Methods**	
Dataset & Metrics	R1-Prompt	SFT-Prompt	LoRA-SFT	LoRA-SFT-S	WeStar$_{MDPO}$	WeStar
cluster 0 Q-A	4.49	4.20	4.34	4.44	4.50	**4.56**
cluster 0 C-A	4.53	4.28	4.37	4.51	4.62	**4.63**
cluster 0 S-A	4.61	4.23	4.40	4.73	4.69	**4.74**
cluster 0 Fluency	4.87	4.78	4.91	4.91	**4.96**	4.92
cluster 1 Q-A	4.48	4.28	4.42	4.44	**4.58**	4.54
cluster 1 C-A	4.53	4.28	4.48	4.50	**4.70**	4.66
cluster 1 S-A	4.69	3.23	4.68	**4.78**	4.76	4.77
cluster 1 Fluency	4.87	4.65	**4.92**	4.90	4.90	4.89
cluster 2 Q-A	4.27	4.21	4.15	4.33	**4.37**	4.32
cluster 2 C-A	4.42	4.24	4.29	4.41	**4.48**	4.46
cluster 2 S-A	3.91	**4.21**	4.02	4.05	4.11	4.20
cluster 2 Fluency	4.63	**4.82**	4.77	4.71	4.75	4.78
cluster 3 Q-A	4.31	4.29	**4.42**	4.38	4.40	4.40
cluster 3 C-A	4.43	4.32	4.51	4.46	4.49	**4.57**
cluster 3 S-A	**3.85**	3.80	3.62	3.75	3.72	3.79
cluster 3 Fluency	4.64	4.60	4.58	4.59	4.56	**4.67**
Due to the space limit, we put the detailed results of other six clusters in the **Appendix**.						
average Q-A	4.38	4.26	4.35	4.41	**4.44**	4.43
average C-A	4.45	4.30	4.43	4.49	4.52	**4.55**
average S-A	**4.25**	3.73	3.92	4.22	4.20	**4.25**
average Fluency	4.75	4.70	4.73	**4.77**	4.76	**4.77**

Bold indicates the best result; Underline indicates the second-best.

Component	**Time (s)**
Article retrieval	0.21
Style tree lookup	0.01
LoRA loading	0.15
LLM inference	1.71
Total	**2.08**

Table 2: Per-component latency breakdown of WeStar.

Q：如何通过艺术风格判断不同文明间的交流证据？

WeStar: 你看那爱琴海壁画，工匠们跋山涉水留下画作，跟江湖游侠在各地酒楼题诗差不多。前些日子见着叙利亚象牙家具上刻的战斗场景，分明是腓尼基商船带来的东方风格……

SFT-Prompt: 你看那些东地中海的壁画工匠，带着手艺到处跑，留下的图像风格能横跨巴比伦到尼罗河。美索不达米亚的浮雕技法会出现在波斯湾，印度河的珠宝工艺能影响到安纳托利亚……

R1-Prompt: 壁画兽首长了波斯眼、埃及角，那就夹带了商路的私货。工匠画壁的时候混进两三种技法，国际风格便溅出泥点。锅碗瓢盆埋进古船，沿岸全是异国物件……

Figure 4: Cases of WeStar vs. Prompt-Based Methods

4.5 Memory Analysis

In industrial deployment practices, we observed that using fewer than 200 style clusters can effectively characterize the stylistic features of millions of official account authors. According to our experimental setup, each cluster corresponds to approximately 0.2GB of parameter storage, with the total storage overhead significantly lower than the space occupied by the model itself. This additional storage cost is entirely acceptable in industrial-scale application scenarios.

4.6 Case Study

We present a case study in Figure 4, comparing responses generated by WeStar and two prompt-based methods for the same input question. The question is designed to reference three article snippets from an Official Account. To make the comparison intuitive and reader-friendly, we choose a stylistic exemplar from Chinese literature—the fictional character Huang Rong (wikipedia, 2025) from Jin Yong's novels—as the target persona. Due to space limits, we omit the referenced article snippets and only display the opening portion of each generated response. Phrases, words, or sentence fragments that align with Huang's style are underlined. As shown, WeStar produces responses that more consistently reflect the target style. Notably, the last sentence generated by WeStar showed in Figure 4 omits the subject, which mirrors a grammatical-level stylistic trait of Huang's speech patterns. In contrast, both prompt-based methods underperform in stylistic consistency. This is likely due to the injection of lengthy article segments into the prompt, which increases the context length and challenges the model's attention window. This case highlights the effectiveness of WeStar's parameterized style representation: by encoding style-specific knowledge directly into the model's parameter space, WeStar avoids the limitations imposed by prompt size and better preserves target style during generation.

4.7 Error Propagation Analysis

To understand how retrieval errors propagate, we manually annotated 400 queries for context and style retrieval correctness, then measured downstream metrics under each condition. Table 4.7 shows that context and style errors exhibit *disjoint* propagation patterns: context failure alone collapses C–A to 2.13 but preserves S–A (4.45); style failure alone drops S–A to 3.11 but leaves C–A intact (4.57). Fluency remains stable (∼4.77–4.79), indicating it is decoupled from retrieval quality.

Condition	C–A	Q–A	S–A	Fluency
Both correct	4.55	4.43	4.26	4.77
Only context correct	4.57	4.43	3.11	4.79
Only style correct	2.13	4.44	4.45	4.77
Both incorrect	1.98	4.43	3.23	4.79

Table 3: Error propagation from context and style retrieval (n=400).

Detailed human evaluation results are provided in Appendix 6.3.

5 Conclusion

In this work, we tackle the underexplored yet practical task of stylized contextual question answering for official accounts, where responses must be both style-aware and context-grounded at scale. Existing fine-tuning, CoT, and prompt-based methods struggle with efficiency or scalability in industrial settings. We propose WeStar, a lite-adaptive AI assistant designed for millions of multi-style Official Accounts. WeStar employs a dual-injection approach: it injects retrieved context into the prompt and style-specific parameters into the model. It builds high-quality training data through hierarchical style clustering, LLM-based stylized rewriting, and metric-driven selection, and further enhances stylistic alignment via Style-enhanced DPO. Experiments on large-scale industrial data show that WeStar outperforms strong baselines in contextual relevance, style fidelity, and fluency—offering an effective, scalable solution for real-world deployments.

Limitations

Data sufficiency. This framework assumes each style cluster contains sufficient representative author data for robust LoRA training. For authors with limited historical responses or underrepresented styles, the model may fail to reproduce their stylistic nuances accurately. We alleviate this via style clustering trees that ensure each category has adequate corpus.

Input length constraint. While the dual-injection design removes style corpus from the prompt, the retrieved article context still occupies the input window. For exceptionally long articles, the prompt length bottleneck remains, potentially affecting generation quality. Standard mitigation strategies such as document summarization or chunk selection can be applied and are orthogonal to WeStar.

Reproducibility. Our evaluation is conducted on a proprietary industrial dataset. Although we will release an anonymized subset of 5,000 instances covering 10 style clusters at `https://github.com/FxyQwQ/WeStar`, the lack of an established public benchmark limits reproducibility. Additionally, we do not compare against closed-source models (e.g., GPT-4) due to reproducibility considerations.

Ethical considerations. The ability to mimic an author's style raises potential misuse concerns, such as impersonation or generating misleading content. In this work, we have taken a preliminary step to mitigate these risks by employing an LLM-based filter to screen and remove offensive content or personally identifiable information from the training data. However, this filter may not be exhaustive, and the model could still be prompted to generate content that aligns with the learned style but is factually inaccurate or ethically problematic. Future deployments should incorporate more robust safeguards—such as style watermarking, real-time content moderation, and clear user consent mechanisms—to ensure responsible and ethical use of the technology.

Style dimensionality. Our style representation is built on 12 stylistic dimensions that capture the majority of salient variance (>90% via PCA) with low inter-dimension redundancy. However, this set may not exhaustively cover all aspects of style.

References

Riku Arakawa, Jill Fain Lehman, and Mayank Goel. 2024. Prism-q&a: Step-aware voice assistant on a smartwatch enabled by multimodal procedure tracking and large language models. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 8(4):180:1–180:26.

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 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Xiang Gao, Yizhe Zhang, Sungjin Lee, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2019. Structuring latent spaces for stylized response generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1814–1823. Association for Computational Linguistics.

Jingxuan Han, Quan Wang, Licheng Zhang, Weidong Chen, Yan Song, and Zhendong Mao. 2023. Text style transfer with contrastive transfer pattern mining. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7914–7927. Association for Computational Linguistics.

Liqiang Jing, Xuemeng Song, Xuming Lin, Zhongzhou Zhao, Wei Zhou, and Liqiang Nie. 2024. Stylized data-to-text generation: A case study in the e-commerce domain. *ACM Trans. Inf. Syst.*, 42(1):25:1–25:24.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15339–15353. Association for Computational Linguistics.

Jinpeng Li, Zekai Zhang, Xiuying Chen, Dongyan Zhao, and Rui Yan. 2023. Stylized dialogue generation with feature-guided knowledge augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 7144–7157. Association for Computational Linguistics.

Jinpeng Li, Zekai Zhang, Quan Tu, Xin Cheng, Dongyan Zhao, and Rui Yan. 2024a. Stylechat: Learning recitation-augmented memory in llms for stylized dialogue generation. *CoRR*, abs/2403.11439.

Jinpeng Li, Zekai Zhang, Quan Tu, Xin Cheng, Dongyan Zhao, and Rui Yan. 2024b. Stylechat: Learning recitation-augmented memory in llms for stylized dialogue generation. *CoRR*, abs/2403.11439.

Ruibo Liu, Chongyang Gao, Chenyan Jia, Guangxuan Xu, and Soroush Vosoughi. 2022. Non-parallel text style transfer with self-parallel supervision. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Xinyue Liu, Harshita Diddee, and Daphne Ippolito. 2024. Customizing large language model generation style using parameter-efficient finetuning. In *Proceedings of the 17th International Natural Language Generation Conference, INLG 2024, Tokyo, Japan, September 23 - 27, 2024*, pages 412–426. Association for Computational Linguistics.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5116–5122. ijcai.org.

Sourabrata Mukherjee, Atul Kr. Ojha, and Ondrej Dusek. 2024. Are large language models actually good at text style transfer? In *Proceedings of the 17th International Natural Language Generation Conference, INLG 2024, Tokyo, Japan, September 23 - 27, 2024*, pages 523–539. Association for Computational Linguistics.

Phil Sidney Ostheimer, Mayank Kumar Nagda, Marius Kloft, and Sophie Fellenz. 2024. Text style transfer evaluation using large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 15802–15822. ELRA and ICCL.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 837–848. Association for Computational Linguistics.

Qingfeng Sun, Can Xu, Huang Hu, Yujing Wang, Jian Miao, Xiubo Geng, Yining Chen, Fei Xu, and Daxin Jiang. 2022. Stylized knowledge-grounded dialogue generation via disentangled template rewriting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3304–3318. Association for Computational Linguistics.

Judith van Stegeren and Jakub Mysliwiec. 2021. Fine-tuning GPT-2 on annotated RPG quests for NPC dialogue generation. In *FDG'21: The 16th International Conference on the Foundations of Digital Games 2021, Montreal, QC, Canada, August 3-6, 2021*, pages 2:1–2:8. ACM.

Yihang Wang, Bowen Tian, Yueyang Su, Yixing Fan, and Jiafeng Guo. 2025. MDPO: customized direct preference optimization with a metric-based sampler for question and answer generation. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 10660–10671. Association for Computational Linguistics.

wikipedia. 2025. Huang rong.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. 2024. Qa-lora: Quantization-aware low-rank adaptation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Yinhe Zheng, Zikai Chen, Rongsheng Zhang, Shilei Huang, Xiaoxi Mao, and Minlie Huang. 2021. Stylized dialogue response generation using stylized unpaired texts. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14558–14567. AAAI Press.

6 Appendix

6.1 Prompt

In this section, we outline the exact prompts for all experiments related to WeStar.

- forward-thinking question generation: Figure 5.
- bottom-up user role generation: Figure 6.
- bottom-up question generation: Figure 7.
- answer generation: Figure 8.
- style labeling: Figure 9.
- stylized answer rewriting: Figure 10.
- answer evaluation: Figure 11.
- generation for prompt-based methods: Figure 12.

6.2 Full Table

In this section, we show the detailed results of all 10 style clusters in Table 4.

6.3 Human Evaluation

To validate the reliability of our LLM-as-a-judge metrics, we conducted a comprehensive human evaluation study.

Multi-judge agreement. To ensure our conclusions are not dependent on any single LLM judge, we employ three SOTA models—Qwen3-235B, DeepSeek-R1, and DeepSeek-V3—as independent evaluators. For each judge, we compute performance differences between every baseline and WeStar, then apply Wilcoxon signed-rank tests across all baseline–method pairs. All p-values fall below 0.05, confirming statistically significant agreement in method rankings across judges.

Annotation Setup. Three expert annotators (with backgrounds in computational linguistics and AI product management) evaluated n=200 randomly sampled instances (40 per cluster from 5 clusters) along the same four dimensions used for automatic evaluation: Contextual Alignment (C–A), Question Relevance (Q–A), Stylistic Strength (S–A), and Fluency. Each dimension was scored on a 1–5 Likert scale. The inter-annotator agreement measured by Fleiss' Kappa reaches 0.74 (substantial).

Table 6.3 reports the mean human evaluation scores, which confirm the same ranking observed in automatic evaluations: WeStar leads across all metrics.

Method	C–A	Q–A	S–A	Fluency
R1-Prompt	4.42	4.42	4.38	4.72
SFT-Prompt	4.21	4.18	3.89	4.65
WeStar	**4.58**	**4.47**	**4.52**	**4.81**

Table 5: Human evaluation results (mean scores, 1–5 scale, n=200).

Correlation with LLM judges. We calculated the Spearman rank correlation coefficient between the human ratings and the corresponding LLM-as-a-judge scores. The correlation is ρ=0.81 ($p <$ 0.01), indicating a statistically significant positive correlation and supporting the consistency of our automated metrics.

Qualitative disagreement analysis. Qualitative analysis of disagreements between human and LLM judgments reveals two primary patterns: (1) subtle stylistic nuances (e.g., author-specific sentence-level habits) are occasionally missed by LLMs, accounting for 12.5% of divergent cases; (2) LLMs tend to penalize minor factual inconsistencies that human annotators deem acceptable, representing 8.0% of disagreements. These findings suggest that while LLM-based evaluation serves as a reliable scalable proxy, human judgment remains indispensable for capturing fine-grained stylistic distinctions.

forward-thinking question generation

System: You are an expert generator of data.

You are a research scientist. You want to make data to train a context question answering system.

Generate a question from these given articles:
{COLLECTION_OF_ARTICLES}

Answer in the following format:
Question:<Question>

Figure 5: Prompt Used for Forward-Thinking Question Generation

bottom-up user role generation

System: You are an expert role generator.

You are a research scientist. You want to create data to simulate user questions for a WeChat Official Account assistant.

Generate {K} possible user roles who might ask questions based on the following WeChat Official Account domain:
Domain: {WECHAT_ACCOUNT_DOMAIN}

Answer in the following format:
Roles:<Role 1>, <Role 2>, ..., <Role {K}>

Figure 6: Prompt Used for Bottom-Up User Role Generation

bottom-up user question generation

System: You are an expert question generator.

You are a research scientist. You want to create data to simulate realistic user questions for a WeChat Official Account assistant.

Given a WeChat Official Account domain and a list of user roles, generate one representative question for each role. Each question should reflect the interests, needs, or concerns of the role within the context of the given domain.
Domain: {WECHAT_ACCOUNT_DOMAIN}
Roles: <Role 1>, <Role 2>, ..., <Role {K}>

Answer in the following format:
Questions:
<Role 1>: <Question 1>
<Role 2>: <Question 2>
...
<Role {K}>: <Question 10>

Figure 7: Prompt Used for Bottom-Up User Question Generation

answer generation

System: You are an expert generator of data.

You are a research scientist. You want to make data to train a context question answering system.

Generate an answer to the question strictly based on the information provided in the following articles. Do not use any external knowledge or assumptions. Only include information that is explicitly stated or can be directly inferred from the articles.
Articles: {COLLECTION_OF_ARTICLES}
Question: {QUESTION}

Answer in the following format:
Answer:<Answer>

Figure 8: Prompt Used for Answer Generation

style labeling

System: You are an expert style annotator.

You are a research scientist. You want to label the stylistic attributes of author replies on a WeChat Official Account.

Given a user comment and a corresponding author reply, assign style labels to the reply based on the specified style dimensions and their candidate labels. Your labeling should consider the content, tone, and intention of the author reply in the context of the user comment.

Style Dimensions and Candidate Labels:
{DIMENSION_1}: {LABEL_A, LABEL_B, ...}
{DIMENSION_2}: {LABEL_C, LABEL_D, ...}
...

Input Pair:
User Comment: {COMMENT}
Author Reply: {REPLY}

Answer in the following format:
{DIMENSION_1}: <LABEL>
{DIMENSION_2}: <LABEL>
...

Figure 9: Prompt Used for Style Labeling

stylized answer rewriting

System: You are an expert in answer rewriting.

You are a research scientist. You want to rewrite a factual answer in the style of a specific author, for use in a WeChat Official Account assistant.

You are given:

- A set of articles providing context.
- A user question.
- A factual answer to the question.
- An author's style profile (multi-dimensional style labels).
- Several example user-comment and author-reply pairs written in the same author's style.

Your task is to rewrite the answer so that it matches the author's style, based on both the style labels and the example pairs. You may rephrase, restructure, or add stylistic elements, but you must **preserve the factual content** from the original answer.

Input:
Articles: {COLLECTION_OF_ARTICLES}
Question: {QUESTION}
Answer: {ANSWER}
Author Style Labels:
{DIMENSION_1}: {LABEL_1}
{DIMENSION_2}: {LABEL_2}
...
Style Examples:
User: {EXAMPLE_QUESTION_1}
Author: {EXAMPLE_REPLY_1}
User: {EXAMPLE_QUESTION_2}
Author: {EXAMPLE_REPLY_2}
...

Answer in the following format:
Stylized_Answer:<Your rewritten answer here>

Figure 10: Prompt Used for Stylized Answer Rewriting

answer evaluation

System: You are an expert in evaluation.

You are a research scientist. You want to evaluate a stylized answer generated for a WeChat Official Account assistant.
You are given: Articles (context), a user question , a stylized answer , author's style profile (multi-dimensional labels) , and several example replies by the same author.
Evaluate the answer on four dimensions, each from 1 (poor) to 5 (excellent), using the criteria below:
1. Contextual Alignment
1 – Unrelated or wrong
2 – Partly relevant, key errors
3 – Basically right, some key points missing
4 – Accurate, covers main ideas
5 – Fully accurate, detailed, article-grounded
2. Question Relevance
1 – Off-topic or misinterprets
2 – General match, vague or weak support
3 – Covers some points, misses key ones
4 – Covers most points clearly
5 – Fully answers with structure and detail
3. Stylistic Strength
1 – No style match
2 – Scattered style traits only
3 – Style theme clear, but inconsistent details
4 – Style is coherent and consistent
5 – Fully matches the intended style
4. Fluency
1 – Serious grammar issues
2 – Awkward, hard to follow in parts
3 – Mostly smooth, minor awkwardness
4 – Natural, few issues
5 – Very fluent and native-like
Input:
Articles: {COLLECTION_OF_ARTICLES}
Question: {QUESTION}
Stylized_Answer: {STYLIZED_ANSWER}
Author Style Profile: {STYLE_DIMENSION_1: LABEL, STYLE_DIMENSION_2: LABEL, ...}
Author Reply Examples:
User: {EXAMPLE_QUESTION_1}
Author: {EXAMPLE_REPLY_1}
User: {EXAMPLE_QUESTION_2}
Author: {EXAMPLE_REPLY_2}
...
Answer in the following format:
Scores:
Contextual Alignment: <1–5>
Question Relevance: <1–5>
Stylistic Strength: <1–5>
Fluency: <1–5>

Figure 11: Prompt used for answer evaluation

generation for prompt-based methods

System: You are an expert stylized response generator.

You are a research scientist. You want to generate a stylized answer for a WeChat Official Account assistant that matches a specific author's writing style.

You are given:

- A set of articles providing context
- A user question
- The author's style profile (multi-dimensional labels)
- A few example replies from this author

Use the author style profile and examples to infer the tone, structure, emotion, and linguistic patterns preferred by the author. Then, write a response that:

- Accurately reflects the content from the articles
- Directly answers the user's question
- Strongly aligns with the author's style
- Is fluent and natural-sounding

Input:
Articles: {COLLECTION_OF_ARTICLES}
Question: {QUESTION}
Author Style Profile: {STYLE_DIMENSION_1: LABEL, STYLE_DIMENSION_2: LABEL, ...}
Author Reply Examples:
User: {EXAMPLE_QUESTION_1}
Author: {EXAMPLE_REPLY_1}
User: {EXAMPLE_QUESTION_2}
Author: {EXAMPLE_REPLY_2}
...

Output Format:
Stylized_Answer: <Your generated answer here>

Figure 12: Prompt Used for Generation in Prompt-Based Methods

Table 4: Full Results

	Prompt-Based Methods		**SFT-Based Methods**		**DPO-Based Methods**	
Dataset & Metrics	R1-Prompt	SFT-Prompt	LoRA-SFT	LoRA-SFT-S	WeStar$_{MDPO}$	WeStar
cluster 0 Q-A	4.49	4.20	4.34	4.44	<u>4.50</u>	**4.56**
cluster 0 C-A	4.53	4.28	4.37	4.51	<u>4.62</u>	**4.63**
cluster 0 S-A	4.61	4.23	4.40	<u>4.73</u>	4.69	**4.74**
cluster 0 Fluency	4.87	4.78	4.91	4.91	**4.96**	<u>4.92</u>
cluster 1 Q-A	4.48	4.28	4.42	4.44	**4.58**	<u>4.54</u>
cluster 1 C-A	4.53	4.28	4.48	4.50	**4.70**	<u>4.66</u>
cluster 1 S-A	4.69	3.23	4.68	**4.78**	4.76	<u>4.77</u>
cluster 1 Fluency	4.87	4.65	**4.92**	<u>4.90</u>	<u>4.90</u>	4.89
cluster 2 Q-A	4.27	4.21	4.15	<u>4.33</u>	**4.37**	4.32
cluster 2 C-A	4.42	4.24	4.29	4.41	**4.48**	<u>4.46</u>
cluster 2 S-A	3.91	**4.21**	4.02	4.05	4.11	<u>4.20</u>
cluster 2 Fluency	4.63	**4.82**	4.77	4.71	4.75	<u>4.78</u>
cluster 3 Q-A	4.31	4.29	**4.42**	4.38	<u>4.40</u>	<u>4.40</u>
cluster 3 C-A	4.43	4.32	<u>4.51</u>	4.46	4.49	**4.57**
cluster 3 S-A	**3.85**	<u>3.80</u>	3.62	3.75	3.72	3.79
cluster 3 Fluency	<u>4.64</u>	4.60	4.58	4.59	4.56	**4.67**
cluster 4 Q-A	4.37	4.20	4.26	**4.41**	<u>4.40</u>	4.27
cluster 4 C-A	4.40	4.23	4.34	<u>4.47</u>	**4.49**	4.39
cluster 4 S-A	4.63	3.08	3.11	**4.83**	4.75	<u>4.81</u>
cluster 4 Fluency	4.83	4.59	4.65	**4.91**	4.87	<u>4.88</u>
cluster 5 Q-A	4.42	4.23	4.28	**4.51**	4.45	<u>4.48</u>
cluster 5 C-A	4.50	4.31	4.34	**4.62**	4.47	<u>4.52</u>
cluster 5 S-A	4.47	3.53	4.08	**4.62**	4.58	<u>4.60</u>
cluster 5 Fluency	4.85	4.73	4.78	4.91	<u>4.92</u>	**4.93**
cluster 6 Q-A	4.32	4.30	4.39	**4.46**	4.37	<u>4.44</u>
cluster 6 C-A	4.38	4.30	<u>4.54</u>	4.51	4.43	**4.66**
cluster 6 S-A	<u>3.99</u>	**4.02**	3.65	3.65	3.79	3.81
cluster 6 Fluency	<u>4.71</u>	**4.72**	4.62	4.64	4.60	4.62
cluster 7 Q-A	<u>4.48</u>	4.35	4.41	4.42	**4.50**	4.44
cluster 7 C-A	**4.56**	4.38	4.50	4.44	<u>4.53</u>	4.51
cluster 7 S-A	**4.57**	3.37	4.37	<u>4.54</u>	4.36	4.39
cluster 7 Fluency	**4.86**	4.74	4.77	**4.86**	<u>4.81</u>	4.79
cluster 8 Q-A	4.36	4.31	<u>4.39</u>	4.35	4.38	**4.51**
cluster 8 C-A	4.42	4.36	4.45	<u>4.47</u>	4.45	**4.58**
cluster 8 S-A	**4.02**	<u>3.94</u>	3.82	3.75	3.69	3.78
cluster 8 Fluency	<u>4.65</u>	**4.69**	<u>4.65</u>	<u>4.65</u>	4.57	4.64
cluster 9 Q-A	4.30	4.26	<u>4.40</u>	<u>4.40</u>	**4.46**	4.38
cluster 9 C-A	4.37	4.30	4.50	<u>4.54</u>	4.53	**4.56**
cluster 9 S-A	<u>3.73</u>	**3.90**	3.41	3.47	3.50	3.64
cluster 9 Fluency	4.54	**4.72**	4.62	<u>4.64</u>	4.63	4.57
average Q-A	4.38	4.26	4.35	4.41	**4.44**	<u>4.43</u>
average C-A	4.45	4.30	4.43	4.49	<u>4.52</u>	**4.55**
average S-A	**4.25**	3.73	3.92	<u>4.22</u>	4.20	**4.25**
average Fluency	4.75	4.70	4.73	**4.77**	<u>4.76</u>	**4.77**

Bold indicates the best result; <u>Underline</u> indicates the second-best.