A Decoupled Multi-Agent Framework for Complex Text Style Transfer

Lingxi Zhang[1], Yu-Neng Chuang[1], Guanchu Wang[1,2], Ruixiang Tang[3], Xuanting Cai[4], Rajesh Shenoy[4], Xia Hu[1]
[1]Rice University, [2]University of North Carolina at Charlotte, [3]Rutgers University, [4]Meta Platforms, Inc.

Abstract

Text style transfer (TST) modifies a source sentence to match a target style while preserving its semantics. While existing models perform well on simple styles like sentiment and formality, they struggle with complex, entangled styles such as poetry and brand-specific tones, which require advanced operations to disentangle content and style. We propose a multi-agent self-check framework that contains a large language model (LLM) as a planner for disentangling subtasks and expert agents for executing the subtasks. This training-free multi-agent framework decomposes TST into manageable components, enabling iterative refinement through a self-check module that balances style adherence and content preservation. Experiments on both simple and complex style datasets show our framework significantly improves style strength and content preservation, with strong adaptability in few-shot settings.

1 Introduction

The text style transfer (TST) task aims to modify a source sentence to match a target style while preserving its original semantics. This task is essential for making NLP applications more user-centered and is widely applied in areas such as dialogue systems (Li et al., 2016; Kim et al., 2019; Firdaus et al., 2023; Chang et al., 2024; Yuan et al., 2024; Liu et al., 2023), writing assistants (Johnstone, 2009; Ashok et al., 2013), text debiasing (Clark et al., 2018; Nogueira dos Santos et al., 2018; Chuang et al., 2025), and online healthcare systems (Neeley et al., 2025; Wang et al., 2024b). TST models can handle a diverse range of styles, such as sentiment (He and McAuley, 2016; Shen et al., 2017), formality (Rao and Tetreault, 2018), Shakespearean (Xu et al., 2012), and beyond.

Existing approaches (Dai et al., 2019; Han et al., 2023, 2024) have achieved promising results on simple style transfer benchmarks such as sentiment (He and McAuley, 2016), where stylistic attributes are well-defined and can be modified with minimal fixed edits. However, these methods struggle with complex styles—such as poetry, biblical language, or brand-specific tones—which require broader lexical and structural transformations, necessitating more flexible and adaptive edits, as shown in Figure 3. Such style transfer is particularly challenging due to the entanglement of style and content within sentences: a single token often carries both semantic and stylistic meaning, so applying fixed edits focused solely on style can result in the loss of essential information from the original sentence.

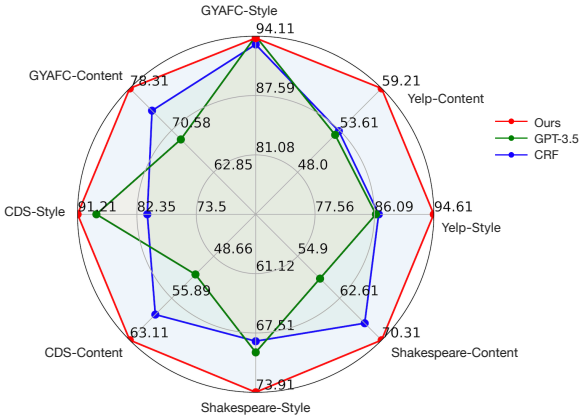

Figure 1: The radar graph of the performance of our framework on both simple and complex styles, showcasing accuracy in content preservation and style strength.

Several efforts have been made to tackle complex style transfer. Some traditional methods (Liu et al., 2021; Li et al., 2020) rely on predefined stylistic rules or attribute templates, but these approaches require extensive manual effort and are difficult to generalize. Meanwhile, some LLM-based methods (Dai et al., 2019; Han et al., 2023, 2024) devise specific loss functions to control the output, and the most recent work (Han et al., 2024) leverages disentangled data generation to enhance the training process. Nonetheless, these methods rely on

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 21393–21403
November 4-9, 2025

single-step generation, which limits their ability to balance style strength and content preservation. When style and content are entangled, enforcing a strong stylistic transformation in a single step often results in semantic drift, as the model lacks the capacity to iteratively adjust and refine the output.

We suggest that decomposition can help address these limitations by breaking the complex style transfer task into a sequence of simpler subtasks. Some subtasks focus on stylistic transformation, while others concentrate on content preservation. This separation enables more fine-grained control and reduces interference between style and meaning. However, applying decomposition to text style transfer raises two key questions: (1) How can we automatically decompose style and content for a new target style without strong supervision signals? and (2) How can we coordinate the subtasks to avoid conflicts and maintain a balance between style strength and content preservation?

To address these challenges, we propose a multi-agent self-check framework that decomposes the style transfer task into subtasks coordinated by a large language model (LLM) planner and executed by multiple expert agents. The training-free planner automatically generates both a subtask plan and an interaction plan, specifying the roles of individual agents and how they communicate. Each subtask is handled by an LLM acting as an expert agent. These agents are interconnected based on the planner's instructions and collaborate through a self-check module, which monitors their outputs for consistency. This module iteratively evaluates content preservation while ensuring that stylistic goals are met, reducing conflicts between agents and improving the overall coherence of the output.

Our experiments cover both simple TST datasets (Yelp (Shen et al., 2017) and GYAFC (Rao and Tetreault, 2018)) and complex, entangled style datasets (CDS (Krishna et al., 2020) and Shakespeare (Xu et al., 2012)). Leveraging a multi-agent self-check strategy powered by LLMs, our framework achieves substantial improvements in both style strength and content preservation. Specifically, on the most challenging CDS dataset, we observe a 2.8% increase in style strength and a 4.4% improvement in content preservation. Furthermore, as a training-free method, our framework shows strong adaptability to diverse styles, even in challenging few-shot scenarios.

In summary, our contributions are as follows:

- We introduce an automatically decoupled multi-agent framework for tackling complex, entangled style transfer.
- We incorporate a self-check strategy that enables iterative refinement by each agent, balancing style strength and content preservation.
- We conduct extensive experiments showing that our framework not only improves style transfer performance on both simple and complex styles but also adapts effectively to diverse styles with only a few example samples.

2 Related Work

Traditional Methods. Traditional approaches to text style transfer have primarily focused on simple style transfer tasks, some approaches (Willams, 1992; Jang et al., 2022; Luo et al.) propose style-oriented losses, and some approaches (Fu et al., 2018; Romanov et al., 2019; Tikhonov et al., 2019) leverage an attribute classifier on representations. Recently, StyleTrans (Dai et al., 2019), RACoLN (Han et al., 2023), and DIRR (Liu et al., 2021) have achieve promising result on sentiment transfer. StyleTrans employs style embeddings and incorporates three specific loss functions to provide supervision signals for effective style transformation. DIRR adopts a reinforcement learning approach, using a semantic similarity metric as a reward to preserve content during training. RACoLN leverages a reverse attention mechanism to implicitly remove style tokens while integrating content information into style representations through conditional layer normalization. However, it is important to note that these traditional methods depend on labeled training data for model development, limiting their adaptability. Our comparisons with these methods primarily focus on tasks involving simple style transfer.

Generation Based Methods with LLM. In contrast, LLM-based methods have begun to leverage LLMs to achieve competitive performance on complex style transfer tasks (Wang et al., 2024a). For instance, DisenTrans (Han et al., 2024) introduces a disentangled CoT prompting mechanism to synthesize parallel data along with corresponding attribute components for supervision. The model design two custom loss functions to enhance attention to attribute properties and constrain the semantic space, resulting in improved performance on more intricate style transfer tasks (Luo et al., 2025). However, despite leveraging the concept of

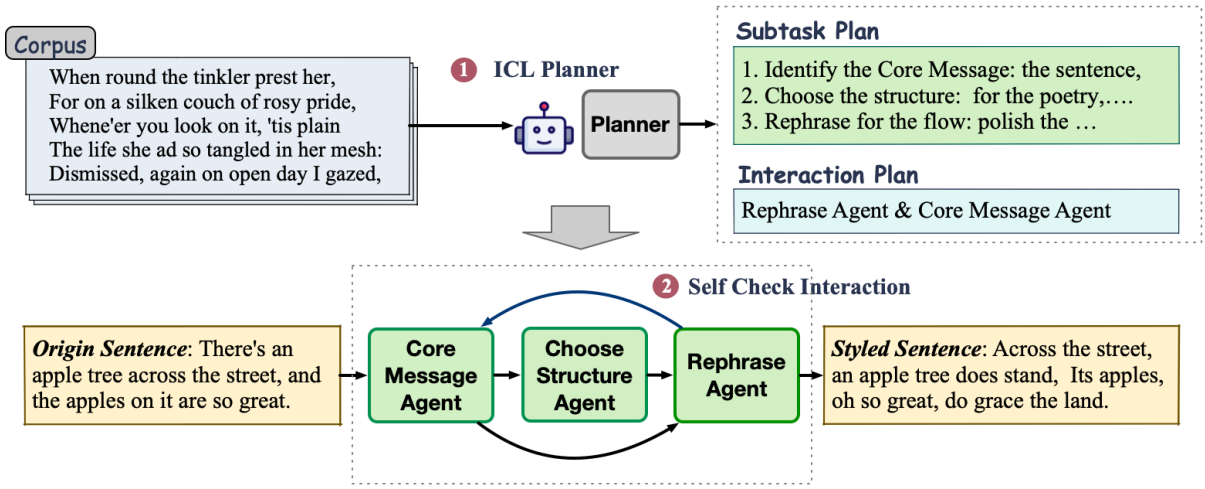

Figure 2: **Overview of Our Multi-Agent Self-Check Framework.** We first leverage a planner to decompose the entangled styles by generating both a subtask plan and an interaction plan (Step 1), then we assigns LLMs as agents to manage each subtasks and leverage self-check strategy to avoid conflict caused by entanglement (Step 2).

disentanglement, it employs a single-step generation process, which can result in conflicts between style strength and content preservation during generation.

3 Preliminary

Given a set of styles $S = \{s_1, s_2, ..., s_n\}$ and a corpus $D = \{(x_{ij}, s_i)\}$, where x_{ij} is a natural sentence and si is its corresponding style label. The text style transfer task is to acquire a model M which takes a natural language sentence x along with a desired style $s \in S$ as input, and then generates a new sentence $x^{'}$ that aligns with the desired style s while maintaining the semantic information of the original input sentence x. To be noted, our proposed framework is training-free, requiring only a small corpus D with some example sentences for the target style, therefore, no parallel training data is needed in our framework.

Entangled Styles. A sentence can be stylistically transformed through atomic lexical edits, such as adding, removing, or replacing individual words. We categorize styles into simple and complex entangled based on the nature and extent of these changes. Simple styles can typically be transferred using a small number of fixed edits—often fewer than three per sentence. In contrast, complex entangled styles require more substantial transformations, including sentence restructuring and multiple coordinated edits, often exceeding three modifications per sentence and involving more diverse types of edits. While our work primarily focuses on complex entangled styles, the proposed framework is broadly applicable to a range of style transfer tasks.

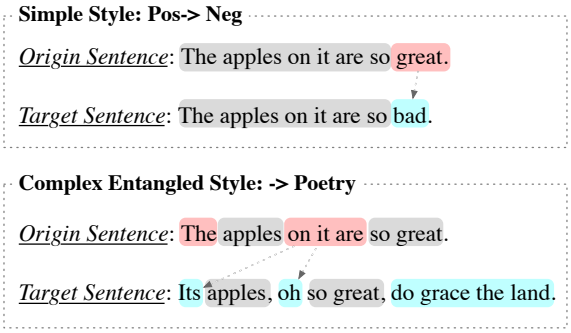

Figure 3: Illustration of Sentence Transfer Examples for Simple and Complex Entangled Styles.

4 Approach

We introduce a multi-agent self-check framework designed for entangled text style transfer tasks. The framework employs an LLM as a planner (Section 3.1) to generate style transfer plans and assigns expert LLM agents (Section 3.2) to execute each subtask, all powered by GPT-3.5. As illustrated in Figure 2, given an input sentence and a target style, the planner produces both subtask and interaction plans, decomposing the entangled style transfer into simpler, manageable steps. Each subtask agent operates sequentially according to the plan and is interconnected through a self-check module, which ensures consistency and prevents conflicts across subtasks. This process is recursive, as the framework iteratively coordinates the agents to address each subtask, ensuring that the final out-

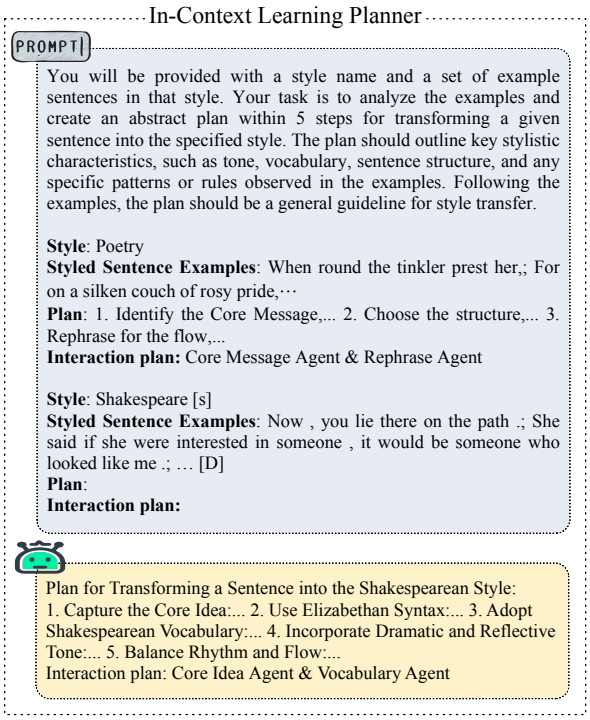

Figure 4: Illustration of prompting LLM to generate subtask plan with in-context learning.

put adheres to the target style while preserving the original meaning.

4.1 LLM as a Planner

The goal of the planner is to decompose complex entangled style transfer tasks into subtasks that can each be addressed through a small number of lexical or structural edits. To achieve this, we prompt the LLM to generate a concise yet comprehensive plan consisting of a few key steps.

Formally, given an input sentence x and a target style $s_i \in S$, along with a set of example sentences $D_s = \{x_{ij} \mid (x_{ij}, s_i) \in D\}$, we construct a prompt P that includes a human-written instruction, the textual description of s_i, and examples from D_s, as shown in the upper part of Figure 4. Using this prompt and in-context learning examples, we query the LLM and treat its output O as the high-level plan for style transfer. We then parse the output into two components based on its format: the subtask plan O_{plan} and the interaction plan $O_{interact}$. The planner generates distinct subtask plans for different target styles, as different styles require attention to different aspects of the input. For example, transferring to a poetic style may emphasize syntactic restructuring, while transferring to a Shakespearean style may prioritize lyrical and expressive phrasing.

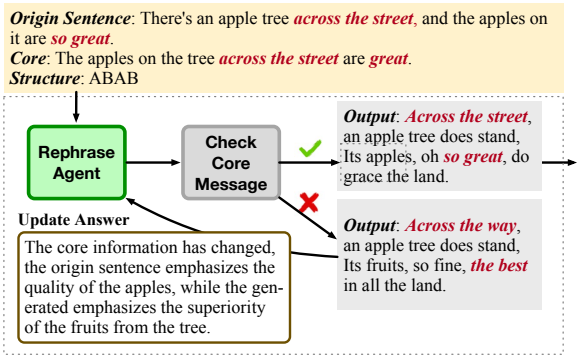

Figure 5: Illustration of Self-Check Interaction.

We retain only outputs that meet the expected format: : O must consist of two paragraphs, with the first beginning with "Plan" and the second with "Interaction Plan." The subtask plan O_{plan} must contain k paragraphs (where $3 \leq k \leq 5$), with each paragraph representing a distinct subtask. If the output does not match this format, we regenerate it until a valid plan is obtained. We opt for shorter plans, as plans with too many steps are likely to cascading errors caused by incorrect assumptions or overly rigid decomposition.

4.2 Multi-agent Framework

Given the planner's output $O_{plan} = D_1 D_2 \ldots D_k$, we extract subtask names and descriptions to construct prompt hints for each subtask. Specifically, we treat the title of each paragraph as the subtask name D_i^{name}, and the accompanying text as the subtask description D_i^{desc}.

For each identified subtask, we assign a powerful LLM as an expert agent, resulting in k agents for k subtasks. For each agent $Agent_i$ corresponding to subtask D_i, we generate the following prompt template:

"You are an expert in D_i^{name}. Given the original sentence and the target style, please transfer the sentence according to D_i^{desc}, using only the information provided below."

Subtasks are executed sequentially in the order specified by the planner. Each agent receives three inputs: (1) the prompt hint specific to its subtask, (2) the original sentence, and (3) the outputs of all preceding agents. This pipeline ensures that each LLM expert focuses on its designated task while building on prior outputs, maintaining consistency with the planner's overall structure and objectives.

Self-Check Interaction. Given the subtask plan and an expert agent for each subtask, a natural ap-

proach is to execute these agents sequentially in a step-by-step manner. However, this naive pipeline can lead to conflicts between agents. For example, a stylistic refinement agent might modify a token or phrase to enhance the sentence's style, then a subsequent content-preservation agent may revert that modification in an effort to maintain the original semantics, rendering the stylistic refinement ineffective.

To address this issue, we introduce a self-check interaction module, illustrated in Figure 5. Based on the interaction plan $O_{interact}$ generated by the planner, we extract a set of interaction tuples:

$$S^{Iter} = \left\{ \left(T_i, T_j, D_{ij}^{Iter} \right) \right\}$$

where T_i and T_j denote the interacting subtasks, and D_{ij}^{Iter} specifies the interaction protocol between them. For each tuple, we instantiate an interaction model LLM_{ij}^{Iter} to manage the coordination.

During execution, if a subtask T_i appears in the interaction plan (i.e., $\exists k \mid (T_i, T_j) \in S^{Iter}$), we activate the self-check module to ensure proper coordination between T_i and T_j.

The self-check module operates as follows: (i) Intermediate Result Generation: Given the current input, the module first calls the agent responsible for T_i to produce an intermediate output. (ii) Conflict Detection: The interaction model LLM_{ij}^{Iter} then evaluates whether the output of T_i conflicts with the requirements or expectations of T_j. If a conflict is detected, the self-check module uses LLM_{ij}^{Iter} to revise the output and feeds the corrected version back to T_i. This revision process is repeated recursively until no conflict is detected or a maximum of five iterations is reached. Figure 5 illustrates an example step in the control flow of the self-check interaction process.

5 Experiment

In this section, we present the empirical evaluation of our proposed multi-agent framework. We begin by introducing the evaluation metrics, datasets, baselines, and experimental setups. Next, we present the main results, followed by a detailed ablation analysis.

5.1 Experimental Setting

Datasets. We evaluate our framework on both simple and complex style datasets. For simple styles, we use the **GYAFC**(Rao and Tetreault, 2018), a parallel formal/informal corpus from Yahoo Answers, and the **Yelp**(Shen et al., 2017) dataset, a non-parallel sentiment-labeled review corpus. For complex styles, we use the **CDS**(Krishna et al., 2020), a non-parallel dataset with 11 diverse styles (e.g., poetry, Biblical), and the **Shakespeare**(Xu et al., 2012) dataset, a parallel corpus translating modern to Shakespearean English. Since CDS lacks sentence pairs, we use it only for style strength and content preservation evaluation.

Evaluation Metrics. We adopt automatic evaluation to assess the effectiveness of our method, focusing on two widely used criteria in style transfer: style transfer strength and content preservation accuracy, following prior work (Xiao et al., 2021).

Style Transfer Strength This metric evaluates whether the generated sentence successfully adopts the target style and measures the degree of stylistic transformation. For sentiment style, we follow (Xiao et al., 2021) and use a fine-tuned SBERT (Reimers, 2019) model for sentiment classification to compute style accuracy. For formality and complex styles, we train a classification model based on a fine-tuned RoBERTa-Large (Liu, 2019) to assess style transfer. For the CDS dataset, which lacks parallel data, we construct a binary classification dataset to train the evaluation model. Specifically, we sample neutral sentences from Wikipedia (Vrandečić and Krötzsch, 2014) as negative examples and use CDS sentences as positive examples. This setup enables the classifier to distinguish styled text from neutral text and provides a proxy for evaluating style accuracy.

Content Preservation Accuracy This metric evaluates how well the generated sentence preserves the original meaning while adapting to the target style. We use three evaluation metrics for this purpose. First, we employ a pre-trained SBERT (Reimers, 2019) model to compute the semantic similarity between the original and generated sentences. Second, for datasets with parallel references, we calculate BLEU scores (Papineni et al., 2002) using the Natural Language Toolkit (Bird et al., 2009), including both Self-BLEU (measuring similarity between the generated output and the input) and Ref-BLEU (measuring similarity to the ground-truth reference). The final content preservation score is computed as the average of the SBERT score, Self-BLEU score, and

Ref-BLEU score.

Baselines. We evaluate several state-of-the-art TST methods, including both traditional approaches—primarily designed for simple style transfer—and recent LLM-based methods that leverage large language models to achieve competitive performance on complex style transfer tasks. For traditional baselines, we include StyleTrans (Dai et al., 2019), RACoLN (Han et al., 2023), and DIRR (Liu et al., 2021), which have demonstrated strong performance on standard TST benchmarks, such as sentiment transfer. It is important to note that these traditional methods rely on supervised training with parallel data and can only be evaluated on simple style transfer tasks.

For LLM-based methods, we first compare our approach with the raw GPT-3.5 (OpenAI, 2024), used via simple prompting. GPT-3.5 also serves as the base model for our framework. Additionally, we compare our method with DisenTrans (Han et al., 2024), which leverages LLMs by introducing a disentangled Chain-of-Thought prompting strategy to synthesize parallel data with corresponding attribute components for supervised training.

Implementation Details. We use GPT-3.5-turbo (OpenAI, 2024) for both the planner and subtask agents. For each dataset, 10 target style examples are randomly selected—preferably from the training set, or from the test set (e.g., CDS) with no evaluation overlap. Planner prompts are manually crafted, while subtask agent prompts are auto-generated from the planner's output.

5.2 Overall Result

The automatic evaluation results are presented in Table 1. Our framework demonstrates competitive overall performance compared to both state-of-the-art traditional baseline methods and LLM-based approaches. Specifically, our proposed method outperforms on complex styles, achieving higher average scores for CDS (style +2.8%, content +4.4%) and Shakespeare(style +4.3%, content +0.3%). It also delivers comparable results on simple style tasks, with strong performance on Yelp (style +1.4%, content +0.5%) and GYAFC benchmarks(content +3.1%).

Our training-free approach can surpass traditional fine-tuned methods on both complex and simple styles. Outperforming SOTA methods including StyleTrans, RACoLN, and DIRR, highlights the effectiveness of large language models in text style transfer tasks. The superior performance of our method can be attributed to the stronger understanding and generalization capabilities of large language models compared to smaller ones, like BERT (Devlin et al., 2019). Furthermore, unlike these traditional methods, which require complex training processes and large datasets, our approach is training-free, making it significantly more efficient and easier to use for inference.

Our method also outperforms LLM-based approaches, particularly on complex styles, including both raw GPT-3.5 and the recent LLM-based approach DisenTrans, which is also powered by LLMs. This superior performance demonstrates that the success of our framework is not solely due to the power of the LLM. Instead, our disentangled multi-agent framework enhances the LLM's ability to understand complex styles, while the division of tasks into simpler subtasks effectively boosts the style strength of the generated sentences. Although raw GPT achieves a higher style strength score on the simple style GYAFC dataset, it suffers from a lower content preservation score. In contrast, our self-check strategy ensures that our model maintains a high content preservation score while achieving a comparable style strength score, striking a better balance between style and content.

5.3 Ablation Study

We conduct ablation studies on two representative styles: the simple style sentiment, using data from the Yelp dataset, and the complex style poetry, using data from the CDS dataset. For both styles, we sample 1,000 instances for evaluation, balancing experimental rigor with the cost of API calls. The ablation study investigates the impact of key components in our framework, specifically the multi-agent strategy and the self-check mechanism. In addition, we perform robustness tests to assess the influence of the base model and prompt design.

Impact of Multi Agent Strategy. We experiment with three variations to analyze the impact of the planner and subtask agents in our multi-agent strategy. In the **"Raw LLM"** variation, the LLM is directly prompted with the style name and a few in-context learning examples, performing the style transfer based solely on this basic information without any decoupling or planning. The **"w/o Multi Agents"** variation extends the "Raw LLM" setup by providing the LLM with a plan in addition to the basic information; however, the style transfer is

Table 1: Overall Accuracy on Text Style Transfer Datasets. (%)

	Yelp		GYAFC		CDS Average		Shakespeare	
	Style	Content	Style	Content	Style	Content	Style	Content
Input Copy	1.4	21.8	5.1	70.1	8.0	60.9	9.6	67.1
StyleTrans (Dai et al., 2019)	90	46	86.3	70.8	75.0	53.0	62.1	69.1
DGST (Li et al., 2020)	88	54.5	79.4	70.1	70.1	51.6	-	-
DIRR (Liu et al., 2021)	92.8	52.3	86.7	75.2	86.9	53.5	63.0	70.0
RACoLN (Han et al., 2023)	86.9	56.3	-	-	-	-	-	-
CRF (Shuo, 2022)	86.7	53.5	93.2	74.2	80.8	58.7	68.4	67.2
DisenTrans (Han et al., 2024)	93.2	58.7	-	-	-	-	61.3	66.5
GPT-3.5 (OpenAI, 2024)	86.3	53.0	**94.1**	68.9	88.4	51.8	69.6	59.0
Ours	**94.6**	**59.2**	93.9	**78.3**	**91.2**	**63.1**	**73.9**	**70.3**

Table 2: Ablation Study on Sentiment and Poetry. (%)

Methods	Sentiment		Poetry	
	Sty.	Cont.	Sty.	Cont.
Ours	92.3	54.3	58.9	43.2
Effect of Plan and Multi Agents Strategy				
Raw LLM	88.5	51.2	54.8	37.8
w/o Multi Agents	88.7	51.9	54.3	41.0
w/o Specific Plan	92.0	58.1	59.0	41.5
Effect of Self Check Interaction Strategy				
w/o Self Check	91.0	50.3	59.5	40.7
w/o Specific Check Plan	92.4	54.0	58.2	42.7

Table 3: Robust Study of Base Model on Poetry. (%)

	Simple Prompt		Our Framework	
	Content	Style	Content	Style
Input Copy	-	3.4	-	-
Style Transformer	49.1	82.1	-	-
LLaMA-8B	28.6	39.7	29.8	53.2
LLaMA-70B	43.8	74.1	48.3	82.1
GPT-3.5	39.0	79.6	**50.3**	**85.9**

executed in a single step, without breaking it into multiple subtasks. Finally, in the **"w/o Specific Plan"** variation, the framework employs a general human-designed abstract plan that keep same across all styles, rather than generating a tailored plan for each specific style. The subtask agents in this case remain consistent with those in the original framework.

The results in Table 2 demonstrate that all these variations lead to a decrease in both style strength and content preservation accuracy. The "Raw LLM" approach shows the most significant drop, highlighting the critical importance of incorporating a decoupling plan in our framework. For complex and entangled styles, the absence of a decoupled plan makes it challenging even for a powerful LLM to generate sentences that balance style strength and content preservation effectively. The "w/o Multi Agents" approach also results in a performance decline, but to a lesser extent than "Raw LLM", suggesting that multi-step execution further enhances performance beyond simply having a plan. The "w/o Specific Plan" variation also reduces accuracy, though the impact is smaller compared to the other variations. This indicates that even a general plan improves the LLM's ability to handle complex styles. However, the remaining performance gap shows that a specific plan is essential for achieving optimal results.

Impact of Self-Check Interaction Strategy. We conduct experiments with two variations to evaluate the impact of the self-check interaction module in our framework. In the **"w/o Self Check"** variation, the self-check strategy is entirely removed. In this case, the multiple agents execute their tasks sequentially according to the subtask plan, and the output of the last agent is taken as the final transferred sentence without further verification or refinement. Instead of generating a tailored interaction check plan for each specific style, the **"w/o Specific Check Plan"** variation approach uses a simple, human-designed plan that is uniformly applied to all styles. All other components and steps remain consistent with the origin framework.

The results in Table 2 reveal that "w/o Self Check" leads to a decline in content preservation accuracy but does not much affect style strength. This suggests that the style strength in text style transfer primarily depends on the LLM's understanding and decoupling of complex styles to make sentences more aligned with a target style. In the datasets used for this experiment, most subtask

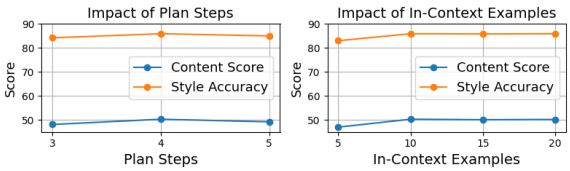

Figure 6: Impact of Plan Steps and In-Context Examples on Performance for the Poetry Style.

agents focus on style, and conflicts between agents often result in high style strength at the cost of altering the original semantic meaning.

The results also show that "w/o Specific Check Plan" has minimal impact, yielding nearly the same performance as the original framework. This indicates that, for most styles, the key interaction to avoid conflicts occurs between the core message agents and the final polishing agents. Since this interaction is covered in the human-designed abstract plan, a general plan can deliver comparable results. However, we argue that for certain complex styles, a specific interaction plan may be necessary to achieve optimal performance.

Impact of Base model. To evaluate the robustness of our framework, we replace the base model, GPT-3.5, with smaller LLMs such as LLaMA-8B (Touvron et al., 2023) and vary the prompts to assess the impact of both the base model and prompt design. As shown in Table 3, the simple prompt refers to a basic prompt containing only a few in-context learning examples, without the multi-agent framework. Results show that the choice of base model also plays a critical role, as our training-free approach relies on the model's inherent capabilities. Using a smaller model like LLaMA-8B leads to a clear drop in both style strength and content preservation. Nevertheless, our multi-agent framework consistently outperforms simple prompting across all models, demonstrating its robustness and adaptability for TST.

Impact of Robust and Hyperparameter. We investigate the robustness of our framework by varying two key factors: the number of decomposition steps in the planner's output and the number of in-context learning examples, as shown in Figure 6. First, we evaluate plans with 3, 4, and 5 subtasks to assess whether different granularities of the plan affect performance. All configurations show performance improvements over the base setting, with 4-step plans yielding the best trade-off between style strength and content preservation. Second, we examine the effect of varying the number of in-context examples used by the planner (5, 10, 15, 20 examples). We observe that increasing the number of examples generally improves performance, with diminishing returns after 10 examples.

Style: InFormal -> Formal
Input: think about what good it brought about. **GPT-3.5**: Consider the benefits it has brought about. **Ours**: Please consider the good it has brought about.
Style: Negative -> Positive
Input: so far i'm not really impressed **GPT-3.5**: So far, I believe there's still room for improvement. **Ours**: So far, I'm looking forward to being more impressed.
Style: Neutral -> Poetry
Input: There's an apple tree across the street, and the apples on it are so great. **GPT-3.5**: Across the way, an apple tree does stand, Its fruits, so fine, the best in all the land. **Ours**: Across the street, an apple tree does stand, Its apples, oh so great, do grace the land.

Figure 7: Case Study on GPT-3.5 and Our Framework.

5.4 Case Study

Figure 7 presents examples of generated transferred sentences from our framework alongside those produced by the powerful base LLM for both simple styles (sentiment and formality) and the complex style poetry. The results illustrate that our framework effectively balances style strengthening and content preservation. In contrast, the raw LLM can perform well on simple styles, but may omit critical content on the complex style transferring; for instance, in the poetry example, it fails to retain "so great" and "across the street" In comparison, our framework preserves all essential content while successfully transferring it into the poetic style.

6 Conclusion

We propose a multi-agent self-check framework for text style transfer, using an LLM planner and expert agents for subtasks. Unlike prior models that struggle with complex styles, our training-free approach decomposes the task and enables iterative refinement through self-checking, balancing style and content. Experiments on both simple and complex datasets demonstrate that our framework achieves improvements in both style strength and content preservation. Moreover, our approach showcases strong adaptability in few-shot settings, underscoring its potential as a robust and efficient solution for diverse TST tasks.

7 Limitations

While our proposed framework achieves improved performance across multiple datasets and outperforms several baseline methods, it comes with certain limitations. First, the framework relies on the inference of multiple large language models, which leads to significantly higher API call costs compared to single-agent approaches. This increased cost is a known challenge in multi-agent systems. Although we prioritize accuracy in this work—as it is often the most critical factor in style transfer—reducing computational and monetary costs remains an important direction for future research. Second, due to the involvement of multiple agents and recursive refinement steps, repeated experimentation may become time-consuming. In future work, we aim to improve efficiency and make the behavior of the framework more predictable, thereby minimizing the need for extensive tuning or repeated trials.

8 Ethics Statement

The text style transfer model is versatile and can be applied to various styles; however, this flexibility also poses potential risks. The model could be misused to generate sentences in styles containing offensive or even illegal content. In our framework, GPT-3.5 may occasionally produce toxic outputs. To address this, we plan to incorporate a detoxification module in future iterations to better control and filter the generated content.

9 Acknowledgements

This paper follows to the new ACL Policy on AI Writing Assistance, utilizing AI tools solely to assist with language refinement.

References

Vikas Ganjigunte Ashok, Song Feng, and Yejin Choi. 2013. Success with style: Using writing style to predict the success of novels. In *Conference on Empirical Methods in Natural Language Processing*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Chia-Yuan Chang, Zhimeng Jiang, Vineeth Rakesh, Menghai Pan, Chin-Chia Michael Yeh, Guanchu Wang, Mingzhi Hu, Zhichao Xu, Yan Zheng, Mahashweta Das, et al. 2024. Main-rag: Multi-agent filtering retrieval-augmented generation. *arXiv preprint arXiv:2501.00332*.

Yu-Neng Chuang, Leisheng Yu, Guanchu Wang, Lizhe Zhang, Zirui Liu, Xuanting Cai, Yang Sui, Vladimir Braverman, and Xia Hu. 2025. Confident or seek stronger: Exploring uncertainty-based on-device llm routing from benchmarking to generalization. *arXiv preprint arXiv:2502.04428*.

Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, IUI '18, page 329–340, New York, NY, USA. Association for Computing Machinery.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuan-Jing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mauajama Firdaus, Arunav Shandilya, Asif Ekbal, and Pushpak Bhattacharyya. 2023. Being polite: Modeling politeness variation in a personalized dialog agent. *IEEE Transactions on Computational Social Systems*, 10(4):1455–1464.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jingxuan Han, Quan Wang, Zikang Guo, Benfeng Xu, Licheng Zhang, and Zhendong Mao. 2024. Disentangled learning with synthetic parallel data for text style transfer. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15187–15201.

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)*, pages 7914–7927.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Eric Jang, Shixiang Gu, and Ben Poole. 2022. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Barbara Johnstone. 2009. Stance, style, and the linguistic individual. *Stance: sociolinguistic perspectives*, pages 29–52.

Soomin Kim, Joonhwan Lee, and Gahgene Gweon. 2019. Comparing data from chatbot and web surveys: Effects of platform and conversational style on survey response quality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA. Association for Computing Machinery.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany. Association for Computational Linguistics.

Xiao Li, Guanyi Chen, Chenghua Lin, and Ruizhe Li. 2020. Dgst: a dual-generator network for text style transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7131–7136.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Yixin Liu, Graham Neubig, and John Wieting. 2021. On learning text style transfer with direct rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4262–4273.

Zirui Liu, Guanchu Wang, Shaochen Henry Zhong, Zhaozhuo Xu, Daochen Zha, Ruixiang Ryan Tang, Zhimeng Stephen Jiang, Kaixiong Zhou, Vipin Chaudhary, Shuai Xu, et al. 2023. Winner-take-all column row sampling for memory efficient adaptation of language model. *Advances in Neural Information Processing Systems*, 36:3402–3424.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Hoang Anh Duy Le, Shaochen Zhong, Hongyi Liu, Jiayi Yuan, Yang Sui, Vladimir Braverman, Vipin Chaudhary, et al. 2025. Autol2s: Auto long-short reasoning for efficient large language models. *arXiv preprint arXiv:2505.22662*.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. A dual reinforcement learning framework for unsupervised text style transfer.

Matthew Neeley, Guantong Qi, Guanchu Wang, Ruixiang Tang, Dongxue Mao, Chaozhong Liu, Sasidhar Pasupuleti, Bo Yuan, Fan Xia, Pengfei Liu, et al. 2025. Survey and improvement strategies for gene prioritization with large language models. *arXiv preprint arXiv:2501.18794*.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194, Melbourne, Australia. Association for Computational Linguistics.

OpenAI. 2024. Chatgpt: Language model by openai. `https://openai.com/chatgpt`. Accessed: 2024-12-13.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Alexey Romanov, Anna Rumshisky, Anna Rogers, and David Donahue. 2019. Adversarial decomposition of text representation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 815–825.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *Advances in neural information processing systems*, 30.

Yang Shuo. 2022. Tagging without rewriting: A probabilistic model for unpaired sentiment and style transfer. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 293–303.

Alexey Tikhonov, Viacheslav Shibaev, Aleksander Nagaev, Aigul Nugmanova, and Ivan P Yamshchikov. 2019. Style transfer for texts: Retrain, report errors, compare with rewrites. 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)*, pages 3936–3945.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Guanchu Wang, Yu-Neng Chuang, Ruixiang Tang, Shaochen Zhong, Jiayi Yuan, Hongye Jin, Zirui Liu, Vipin Chaudhary, Shuai Xu, James Caverlee, et al. 2024a. Taylor unswift: Secured weight release for large language models via taylor expansion. *arXiv preprint arXiv:2410.05331*.

Guanchu Wang, Junhao Ran, Ruixiang Tang, Chia-Yuan Chang, Yu-Neng Chuang, Zirui Liu, Vladimir Braverman, Zhandong Liu, and Xia Hu. 2024b. Assessing and enhancing large language models in rare disease question-answering. *arXiv preprint arXiv:2408.08422*.

Ronald J Willams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Fei Xiao, Liang Pang, Yanyan Lan, Yan Wang, Huawei Shen, and Xueqi Cheng. 2021. Transductive learning for unsupervised text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2510–2521.

Wei Xu, Alan Ritter, William B Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914.

Jiayi Yuan, Hongyi Liu, Shaochen Zhong, Yu-Neng Chuang, Songchen Li, Guanchu Wang, Duy Le, Hongye Jin, Vipin Chaudhary, Zhaozhuo Xu, et al. 2024. Kv cache compression, but what must we give in return? a comprehensive benchmark of long context capable approaches. *arXiv preprint arXiv:2407.01527*.

A Appendix

A.1 Datasets

Table 4: Statistics of each Style Transfer dataset.

Dataset	Style	Test Num.	Style Type
GYAFC	Formality	1,082	Simple
Yelp	Sentiment	1,000	Simple
CDS	Literature	14,079	Complex
Shak.	Shakespeare	1,293	Complex

We evaluate our framework on two types of datasets: those targeting simple styles and those targeting complex styles. For simple styles, we use the **Grammarly's Yahoo Answers Formality Corpus (GYAFC)**(Rao and Tetreault, 2018) and the **Yelp Review Dataset**(Shen et al., 2017). GYAFC is a parallel corpus containing formal and informal sentence pairs collected from the Yahoo Answers forum. The Yelp dataset is a non-parallel corpus labeled with binary sentiment (positive or negative), consisting of user reviews from various businesses and services on Yelp.

For complex styles, we use the **CDS**(Krishna et al., 2020) and **Shakespeare**(Xu et al., 2012) datasets. CDS is a non-parallel corpus containing 11 distinct and stylistically rich categories, such as poetry and Biblical text. The Shakespeare dataset is a parallel corpus designed to convert modern English into Shakespearean-style language. Since CDS lacks parallel sentence pairs, we use it exclusively to evaluate style strength and content preservation, without measuring content accuracy.