Adversarial Style Augmentation via Large Language Model for Robust Fake News Detection

Anonymous Author(s)*

ABSTRACT

The spread of fake news negatively impacts individuals and is regarded as a significant social challenge that needs to be addressed. A number of algorithmic and insightful features have been identified for detecting fake news. However, with the recent LLMs and their advanced generation capabilities, many of the detectable features (e.g., style-conversion attacks) can be altered, making it more challenging to distinguish from real news. This study proposes adversarial style augmentation, AdStyle, to train a fake news detector that remains robust against various style-conversion attacks. Our model's key mechanism is the careful use of LLMs to automatically generate a diverse yet coherent range of style-conversion attack prompts. This improves the generation of prompts that are particularly difficult for the detector to handle. Experiments show that our augmentation strategy improves robustness and detection performance when tested on fake news benchmark datasets.

CCS CONCEPTS

• **Security and privacy → Software and application security**; **Intrusion/anomaly detection and malware mitigation**; • **Computing methodologies → Artificial intelligence**.

KEYWORDS

Misinformation, Adversarial Training, Fake News Detection, Large Language Model

ACM Reference Format:

Anonymous Author(s). 2025. Adversarial Style Augmentation via Large Language Model for Robust Fake News Detection. In *Proceedings of the ACM International Conference on Web Conference (WWW 25), April 28– May 02, 2025, Sydney, Australia.*. ACM, New York, NY, USA, 11 pages. https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

1 INTRODUCTION

With the widespread use of the internet and the emergence of various social media platforms, people have begun to freely share information they know and their own creative stories. However, the ease of sharing and consuming information beyond traditional news organizations has also significantly contributed to the spread of fake news [3, 11, 30]. Fake news is created to provide readers with false information as propaganda or to guide public perception in a desired direction for intentional objectives [16, 36]. The spread of such false information has had profound negative impacts on individuals and society, becoming a major social challenge that needs to be addressed.

Manually determining the authenticity of all information on the internet is extremely costly in terms of time and resources. Thus, prior advances have mainly focused on developing automated fake news detectors using machine learning techniques [22, 27, 28]. For example, several studies have extracted sentiment-related or political features commonly found in fake news, either through human-crafted [23, 25] or data-driven methods [27, 28], and used them for detection. With the advent of language models (e.g., GPT, BERT) [14, 24], efforts have been made to learn the textual style of fake news by extracting sentence embeddings to train detectors [5, 35].

While these extracted textual styles or human-crafted features have proven effective in identifying fake news, they allow attackers to bypass detection. Various attack methods include changing the order of subjects and objects [15], or exaggerating words [38]. Especially, the advent of large language models (LLMs) [7, 20] has made it possible to easily and automatically paraphrase sentences in a user-desired direction through prompts (e.g., "change the given text to an objective and professional style"), making it more difficult to distinguish between AI-generated fake news and real news (i.e., style-conversion attacks) [15, 38]. Recent literature has proposed strategies to augment the styles of input text via some manually defined style-conversion prompts, which are taken by LLM for paraphrasing, but it remains challenging to address all types of prompts that attackers might use to deceive detectors [32].

In this study, we propose adversarial style augmentation, AdStyle, to train a robust fake news detector that can withstand various style-conversion attacks by attackers. In contrast to earlier work, which used predefined style-conversion prompts as detector-agnostic augmentations, AdStyle tries to find prompts for adversarial style augmentations that are specific to the detector, which makes predictions uncertain. This means that our augmentations add noise to the style features in the direction of the detector's decision boundary, while maintaining the text's content integrity. Specifically, to search for and optimize prompts for LLM that are adversarial to the detector, we introduce an automated prompt engineering technique [34]. By providing LLMs with style-conversion prompts and the detector's performance under these augmentations, LLM can infer patterns between the prompts and performance, enabling the search for the most adversarial prompts.

AdStyle proceeds as follows: To generate augmented samples, we first select a random subset of the dataset and apply a pool of style-conversion prompts to it. Then, each augmentation set created by different prompts is fed into the detector, and the score based on the AUC between the predictions and the ground-truth labels is measured. The prompt-score pairs are provided to the LLM, which uses this information to generate new style-conversion prompt candidates. From these candidates, we select the top-k prompts that are diverse and make the detector's predictions most uncertain without

WWW '25, April 28– May 02, 2025, Sydney, Australia.

ACM ISBN xxx-x-xxxx-xxxx-x/xx/xx…$15.00
https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

significantly altering the original content. These selected prompts are applied to the entire dataset and used for detector training. The chosen prompts are added to the prompt pool, which is repeated over several training rounds.

We experimented with existing fake news benchmark datasets (e.g., PolitiFact, GossipCop, and Constraint) under various style-conversion attack scenarios. As a result, our augmentation strategy demonstrated higher robustness and detection performance compared to previous methodologies. Furthermore, our augmentation strategy preserves the content of the sentence while modifying its structure to increase perplexity according to the LLM-based detector. This adjustment results in a sentence structure that the LLM has not frequently encountered during its pre-training phase. AdStyle is scalable to different attack strategies by adding new attack prompts to the style-conversion prompt pool. We plan to release our code after publication.

2 RELATED WORK

2.1 Automated Detection of Fake News

Manually detecting fake news among the vast amount of content on the internet is intractable. Consequently, many studies have focused on leveraging machine learning techniques to automatically detect fake news [22, 27, 28]. For example, supervised learning methodologies extract textual features from fake news texts using benchmark datasets and ground-truth labels [27]. These textual features can include deep features based on artificial neural networks [27, 28] as well as manually defined features such as sentiment and political bias [23, 25]. Other approaches include domain adaptation methods to enhance detection generalizability across various domains and topics [18, 19], and knowledge-based methods that rely on external data to distinguish false information [9]. With the advent of LLMs, new methods have emerged that utilize LLMs' prior knowledge to identify fake news [12], including research on detecting machine-generated fake news [6].

In this work, we aim to prevent malicious attackers from deceiving automated fake news detectors through text modifications such as paraphrasing. The proposed method is an augmentation strategy to enhance robustness against text style perturbations as an add-on to existing detection models. Our contribution is agnostic to the design of the detection model.

2.2 Attack on Fake News Detection

To test the robustness of fake news detection, various attack methods have been studied [15, 38]. These include injecting misinformation by changing the order of subjects and objects or causes and effects while maintaining the textual features used for detection [15]. Other methods involve creating fact distortions by altering or exaggerating words related to people, time, or places while preserving the sentence structure [38]. Additionally, approaches that use the text generation capabilities of LLMs to change the style of sentences have been proposed [32].

Our method aims to perform adversarial training of the detection model by generating augmentations that perturb the text's style while preserving the content as much as possible. This approach reduces the impact of spurious style features on the detection model, making it more robust against such attacks.

2.3 Prompt Engineering for LLM

Training with extensive text corpora, LLMs have demonstrated their utility across various domain tasks [7, 20, 21]. To better harness the prior knowledge and reasoning abilities of LLMs, strategies for providing appropriate input prompts have also been researched. For instance, in-context learning methods involve providing examples of the desired input-output format to guide the LLM in producing the correct output [8]. Additionally, engineering techniques such as chain-of-thought prompting, which includes additional reasoning steps in the responses, have been developed to enhance the reasoning abilities of LLMs [31]. Recently, methods for automatically finding the most suitable prompts for a given task, known as automated prompt engineering, have shown promising results [34, 37].

In this paper, we adopt automated prompt engineering techniques to identify adversarial prompts that most effectively confuse the fake news detector. This process replaces the traditional gradient descent method used in the image domain to find adversarial noise, enabling the creation of adversarially augmented versions of input text via LLM.

3 METHOD

3.1 Overview

Let $\mathcal{D} = \{(\mathbf{d}_i, y_i)\}_{i=1}^{N}$ be a dataset containing news $\mathbf{d}_i$ and the corresponding ground-truth binary veracity label y_i (indicating whether the news is true or fake). Each news item $\mathbf{d}_i$ is composed of natural language-based text. This study aims to train a language model based fake news detector f using the labeled dataset to predict the veracity labels. Our main goal is to develop a detector that remains robust even when an attacker perturbs the textual style, such as the order and format, while preserving the meaning of the sentences.

Figure 1 illustrates how our model works. AdStyle generates style-conversion prompts and performs augmentation over multiple rounds. Each round includes a following process. Firstly, leveraging the reasoning ability of LLM, adversarial style-conversion prompt candidates are generated. These style-conversion prompts contain instructions on how to transform the given sentences and are used as inputs to the LLM along with the original sentences for conversion. Inspired by automated prompt engineering [34], the style-conversion prompts and detector prediction score pairs used in previous rounds are included as in-context demonstrations to guide the LLM in searching prompt candidates that maximally confuse the detector (Section 3.2). Next, a subset of the dataset is selected, and these discovered candidates are applied to perform conversions. The converted samples are then evaluated to determine how much they confuse the detector. From these candidates, the top-k prompts that are diverse and maintain the original content's meaning while most effectively confusing the detector are selected (Section 3.3). These selected prompts are used as an augmentation method to train the detection model in the current round. We describe each step's details below.

3.2 Generating Adversarial Style-Conversion Prompts with LLM

The style-conversion prompts we aim to generate are instructions that perturb only the textual style, such as the structure or format

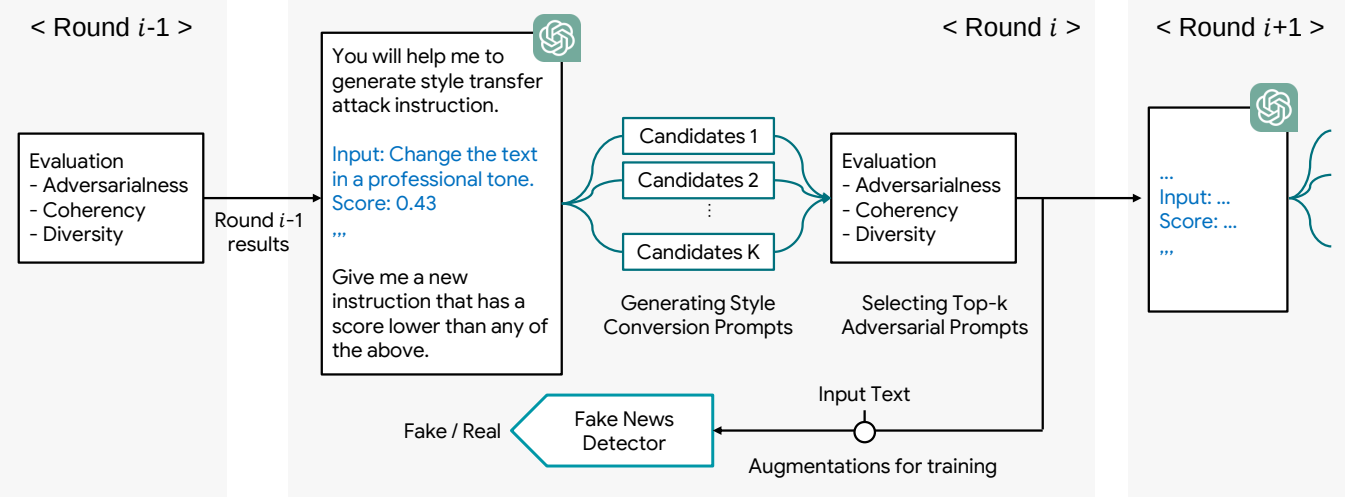


Figure 1: Illustration of how AdStyle works. Our model consists of multiple rounds of training. In each round, the model uses the style-conversion prompt and prediction confusion score from the previous round to generate prompt candidates that can maximize the confusion of the detector model. Subsequently, considering aspects such as adversarialness, coherency, and diversity, a subset of the training dataset is used to select the top-k prompts, which are finally used as augmentations in training.

> Now you will help me minimize a fake news detector performance value with style transfer attack instruction. I have some style transfer attack instructions and the fake news detector performance value using those instructions. The pairs are arranged in descending order based on their function values, where lower values are better.
>
> Input: [Style-Conversion Prompt Example]
> Score: [Prediction Confusion Score]
>
> Input: ... ...
>
> Give me a new style transfer attack instruction that is different from all pairs above, and has a performance value lower than any of the above.

Figure 2: Prompt for generating adversarial style-conversion prompts. The text in blue represents the score trajectory component, while the remaining text represents the problem description component.

of the sentence, while preserving the content of the given input text. For example, an instruction like "Rewrite the following article in an objective and professional tone" can be used to transform the stylistic features of fake news to resemble real news. However, if such instructions are heuristically defined, they may not align with the actual decision boundary of the detector, reducing the training efficiency. Furthermore, if the instructions remain fixed throughout the training, the detector may memorize these conversion patterns, resulting in overfitting.

Therefore, instead of predefining and fixing the instructions for conversion, we aim to find adversarial conversion prompts that add noise in the direction of the decision boundary of current state of detector, maximizing the confusion in the detector's predictions. This approach is similar to adversarial training commonly used in the computer vision domain [17], which enhances robustness against slight perturbations in the input. However, for textual data, the discrete nature of the input makes it challenging to add noise directly through the detector's gradient as in the image domain. To address this, we measure the confusion in prediction by conversion prompts (i.e., prediction confusion score), and feed this prompt-score pair to LLM for generating adversarial prompts, motivated from automated prompt engineering techniques.

Figure 2 shows an example of the LLM input used to generate adversarial style-conversion prompts. The LLM input consists of the problem description and the score trajectory components.

Problem description component. This component includes the problem description, the objective, and constraints on the response necessary for generating style-conversion prompts. For example, a sentence like "Minimize a fake news detector performance value with style transfer attack instruction" informs the LLM of the intent behind the conversion prompt.

Score trajectory component. Previous work has shown that LLMs can learn patterns from in-context demonstrations provided as input [8, 34]. This component leverages this ability by providing previous round's style-conversion prompts and their corresponding prediction confusion scores in the form of in-context demonstrations[1]. The score is measured by selecting a subset $\mathcal{B}$ from the entire training dataset $\mathcal{D} = \{(\mathbf{d}_i, y_i)\}_{i=1}^{N}$, applying a conversion prompt c to create a new set $\mathcal{B}^c = \{(\mathbf{d}_i^c, y_i)\}_{i=1}^{M}$, where $\mathbf{d}_i^c = \text{Convert}(c, \mathbf{d}_i)$, $N >>$

[1]In the first round, a predefined set of prompts is used.

M, and then measuring the AUC score between the predictions and the ground-truth labels when $\mathcal{B}^c$ is fed into the detector. Specifically, the score of a conversion prompt s_c is defined as:

$$s_c = |0.5 - \text{AUC}(\{y_i\}_{i=1}^M, \{f(\mathbf{d}_i^c)\}_{i=1}^M)|. \quad \text{(Eq. 1)}$$

A lower score indicates a higher level of prediction confusion, implying that the conversion prompt has caused the detector's predictions to become random with respect to the labels. Finding conversion prompts that cause high confusion (i.e., low score) suggests that the detector has not yet learned to handle those stylistic features, and the conversion prompts have placed the samples near the detector's decision boundary, making them difficult to distinguish. By showing the LLM these in-context demonstrations, it can generate conversion prompts that differ from past ones while maximizing confusion (i.e., minimizing the score). Note that we avoid selecting conversion prompts that flip the detector's original predictions (i.e., AUC << 0.5), as these often disrupt the content along with the stylistic features, which is undesirable. We extract S style-conversion prompts at a time using the input described above.

3.3 Selecting Top-k Adversarial Prompts

Using all the style-conversion prompt candidates generated by the LLM can be computationally intensive, and not all prompts may be suitable for augmentation. For example, the set of style-conversion prompts used for augmentation should each provide adversarial perturbations that confuse the detector (i.e., adversarialness). Additionally, while altering the textual style, the prompts should not change the meaning of the sentences to prevent label noise (i.e., coherency). Furthermore, the more diverse the set of conversion directions covered by the prompts, the more efficient the augmentation process (i.e., diversity).

To select a set of style-conversion prompts that satisfies these three criteria, we propose a selection strategy. Given a conversion prompt c, we first extract embedding vectors for the input texts in both the training subset $\mathcal{B}$ and the subset $\mathcal{B}^c$ converted by c using a large language model g such as BERT. We then compute the average embedding vectors for each subset and calculate the vector difference $\mathbf{z}_c$.

$$\mathbf{z} = \frac{1}{|\mathcal{B}|}\sum_{\mathbf{d}_i \in \mathcal{B}} g(\mathbf{d}_i), \quad \mathbf{z}' = \frac{1}{|\mathcal{B}'|}\sum_{\mathbf{d}_i^c \in \mathcal{B}'} g(\mathbf{d}_i^c)$$
$$\mathbf{z}_c = \mathbf{z}' - \mathbf{z} \quad \text{(Eq. 2)}$$

Here, $\mathbf{z}_c$ represents the average change in embedding direction due to the conversion prompt c. We then calculate the adversarialness scale s_{adv}^c and coherency scale s_{coh}^c for each conversion prompt c, and rescale $\mathbf{z}_c$ accordingly (i.e., $\hat{\mathbf{z}}_c = \mathbf{z}_c \times s_{\text{adv}}^c \times s_{\text{coh}}^c$). Finally, we use the k-means++ initialization method [1] on these rescaled vectors to select k prompts. The k-means++ initialization method helps select a diverse set of prompts that are adversarial and coherent, by choosing samples that are as far apart as possible [2]. The details of each scale are described below.

Adversarialness scale (s_{adv}^c). To measure how adversarial a given style-conversion prompt is to the detector, we define a new adversarialness scale similar to the confusion score defined in the previous section. Given the converted batch $\mathcal{B}^c$ by conversion prompt c, the adversarialness scale s_{adv}^c is defined as:

$$s_{\text{adv}}^c = -1.8 \cdot |\text{AUC}(\{y_i\}_{i=1}^M, \{f(\mathbf{d}_i^c)\}_{i=1}^M) - 0.5| + 1. \quad \text{(Eq. 3)}$$

This value increases as the AUC approaches 0.5, indicating that the prediction is more random. The coefficient 1.8 ensures that the scale ranges between 0.1 and 1, preventing it from being zero.

Coherency scale (s_{coh}^c). To verify that the converted text retains the same content as the original text, we check the similarity in meaning between the text pairs using an LLM. Specifically, we create sample pairs from $\mathcal{B}$ and the converted subset $\mathcal{B}^c$, and inquire the LLM about the percentage of pairs that it considers to have the same meaning. This percentage is used as the coherency scale s_{coh}^c. Like the adversarialness scale, this value is rescaled to range between 0.1 and 1.

Finally, the selected style-conversion prompts via our score and k-means++ initialization method are applied to the input texts of the entire dataset $\mathcal{D}$ to create augmented samples. These augmented samples are then used alongside the original samples to train the detector f. The detector is trained using binary cross-entropy loss. These prompt generation and selection processes are repeated over multiple training rounds.

4 EXPERIMENT

We evaluate the robustness of AdStyle under diverse style-conversion attacks across multiple datasets, comparing it with contemporary baselines. Additionally, we analyze the impact of various model components on overall performance. An in-depth evaluation is also conducted on a wider range of paraphrasing attacks, including comparisons with LLM-based zero-shot and in-context learning baselines. Finally, a qualitative analysis is performed to examine the characteristics of the style-conversion prompts generated by the LLM.

4.1 Performance Evaluation

Dataset. Our experiments use three real-world fake news benchmark datasets. We utilize PolitiFact and Gossipcop, drawn from the FakeNewsNet benchmark [29], which focus on political claims and celebrity rumors, respectively. Additionally, we incorporate Constraint [10], a dataset specifically addressing COVID-19 related social media posts. Each dataset is randomly split into an 80% training set and a 20% test set. Detailed statistics for these datasets can be found in Table 1.

Table 1: Statistics of fake news datasets.

Dataset	PolitiFact	GossipCop	Constraint
# of News Articles	774	7,916	8,418
# of Real News	399	3,958	4,406
# of Fake News	375	3,958	4,012

Attack settings. To assess robustness against style conversion attacks, we employ LLM-empowered techniques to reframe the test set using a variety of style conversion prompts, as illustrated in Figure 3. Following the original literature [32], we use four well-known daily news sources as [publisher name]: **CNN**, **The New**

Table 2: Performance comparison with AdStyle in two different scenarios—Attack, where style-conversion attacks are performed, and Clean, where no attack is performed—across three fake news datasets. For the attack scenario, we report the average AUC of four style-conversion attacks. The best results are marked in bold. Our model demonstrates a significant performance improvement over all text-based fake news detectors in both style-conversion attack and clean scenarios.

Attack	Politifact			Gossipcop			Constraint		
	0.1	0.25	1	0.1	0.25	1	0.1	0.25	1
Vanilla	0.6114	0.6904	0.8548	0.6920	0.7876	0.8453	0.8185	0.8674	0.8741
UDA	0.6241	0.7696	0.8564	0.7381	0.7865	0.8591	0.8615	0.9028	0.9297
RADAR	0.7218	0.7399	0.8571	0.7583	0.8170	0.8616	0.8535	0.8047	0.9086
ENDEF	0.6376	0.7579	0.8134	0.7405	0.7870	0.8615	0.8234	0.8950	0.8835
SheepDog	0.6525	0.8234	0.9009	0.7498	0.8357	0.8669	0.8926	0.9188	0.9630
AdStyle	**0.7833**	**0.8919**	**0.9399**	**0.8134**	**0.8389**	**0.8721**	**0.9224**	**0.9531**	**0.9716**

Clean	Politifact			Gossipcop			Constraint		
	0.1	0.25	1	0.1	0.25	1	0.1	0.25	1
Vanilla	0.7393	0.8397	0.9355	0.7096	0.8104	0.8645	0.9311	0.9682	0.9892
UDA	0.7404	0.8783	0.9422	0.7422	0.8022	0.8666	0.9365	0.9724	**0.9899**
RADAR	0.7607	0.8495	0.9314	0.7593	0.8170	0.8630	0.9446	0.9773	0.9817
ENDEF	0.7776	0.8823	0.9294	0.7592	0.7991	0.8738	0.9234	0.9556	0.9871
SheepDog	0.7248	0.8229	0.9394	0.7490	0.8411	0.8641	0.9144	0.9459	0.9785
AdStyle	**0.8996**	**0.9280**	**0.9460**	**0.8251**	**0.8493**	**0.8797**	**0.9509**	**0.9849**	0.9889

> Rewrite the following article using the style of [publisher name]: [news article]

Figure 3: Prompt for style conversion. The "publisher name" part will be filled with the name of a representative publisher (e.g., newspaper or journal), and the "news article" part will contain the original news text.

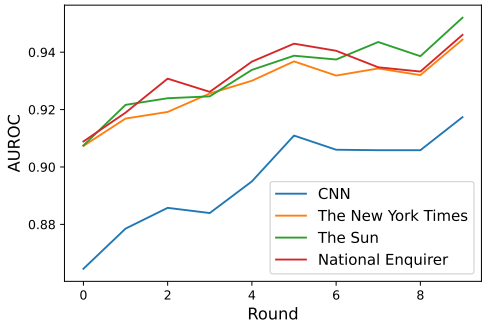


Figure 4: Performance changes across rounds on the PolitiFact dataset for four different style-conversion attacks. The x-axis represents the training rounds, and the y-axis represents the detector's AUC. For all attacks, the detector's performance improved as the rounds progressed.

York Times, **The Sun**, and **National Enquirer**. CNN and The New York Times are recognized for their reputable journalism, while The Sun and National Enquirer are characterized by their tabloid style. We utilize OpenAI's GPT-3.5-Turbo model for reframing input sentences, with the temperature set to 0 and the top-p value set to 1 by default. We conduct experiments using 10%, 25%, and 100% of the complete dataset to observe the effect of augmentation across different dataset sizes for training. AUC is used as our evaluation metric.

Baselines. We have implemented several existing text-based fake news detection strategies as baselines: (1) Vanilla, a text-based fake news detector using conventional binary cross entropy objective; (2) UDA, introduces consistency regularization objective between original text and diverse augmented variations. [33]; (3) RADAR, utilizes an adversarially trained paraphraser to generate augmented version of input sentences [13]; (4) ENDEF, mitigates entity bias in fake news data through causal learning [39]; (5) SheepDog, introduces predefined style-conversion prompts to augment the styles of input text via LLM [32]. We follow the original paper's setting and details for baseline implementations.

Implementation details. All models are evaluated under uniform experimental conditions to ensure fair comparison. This consistency extends to the choice of backbone network, optimizer, and learning rate. We utilize OpenAI's GPT-3.5-Turbo model for reframing input

Table 3: Performance comparison of ablations on the Politifact dataset. The results show the impact of style-conversion attacks using four different publishers (i.e., CNN, The New York Times, The Sun, and National Enquirer) and a clean scenario (i.e., Clean) where no attack is performed. Any modification or removal of model components leads to decreased performance.

Model	CNN	The New York Times	The Sun	National Enquirer	Clean
Vanilla	0.8127	0.8687	0.8789	0.8591	0.9355
Random Selection	0.9075	0.9409	0.9355	0.9365	0.9412
Class Prompt	0.9131	0.9272	0.9333	0.9313	0.9471
Adversarial only Selection	0.9035	0.9343	0.9318	0.9288	0.9405
w/o Adversarialness	0.9039	0.9333	0.9320	0.9402	**0.9473**
w/o Coherency	0.9193	0.9315	0.9297	0.9430	0.9409
w/o Score trajectory	0.8199	0.8917	0.9124	0.9066	0.9372
Full Components	**0.9174**	**0.9444**	**0.9520**	**0.9460**	0.9460

sentences and measuring coherency, with the temperature set to 0 and the top-p value set to 1 by default. Our training process for the detector is conducted over 10 rounds, with one training epoch per round. When evaluating the style-conversion prompts, we randomly selected 30 samples from the training dataset to apply augmentation (i.e., $M = 30$). In each round, 30 prompt candidates were generated by the LLM (i.e., $S = 30$), from which 3 were chosen for augmentation (i.e., $k = 3$) by our selection strategy. The training utilizes the AdamW optimizer with a learning rate of 1e-5 and a batch size of 8. For measuring diversity, the text embeddings are generated using a pre-trained BERT-based uncased model from HuggingFace Transformers. Two V100 GPUs were utilized for all experiments.

Result. Table 2 compares the performance of detector algorithms in both clean scenarios, where no attack is performed, and adversarial scenarios, where style-conversion attacks are applied. Due to space limitations, the average AUC results under the four different style-conversion attacks are reported. Detailed results for each prompt can be found in the Appendix (See Table 7 in the Appendix). According to the results, AdStyle consistently outperforms the baselines across all cases, including both clean and adversarial scenarios. This demonstrates that our augmentation strategy enhances both the robustness and generalizability of the detector. Notably, our model's effectiveness is more pronounced when compared to other baselines, especially with smaller datasets. Figure 4 shows the AUC for each style-conversion attack over different rounds. The performance gradually improves as the rounds progress, indicating that continually discovering adversarial augmentations is beneficial.

4.2 Component Analysis

In this section, we examine the contribution of each component on our adversarial style-conversion prompts. The proposed method integrates two main modules: adversarial style-conversion prompts generation and selection. To evaluate their individual contributions, we conduct experiments where we either remove each component or substituted it with an alternative within the full model. This results in six distinct configurations for analysis: (1) **Full Components**: Our complete method with all components; (2) **Random Selection**: The method that randomly select conversion prompts

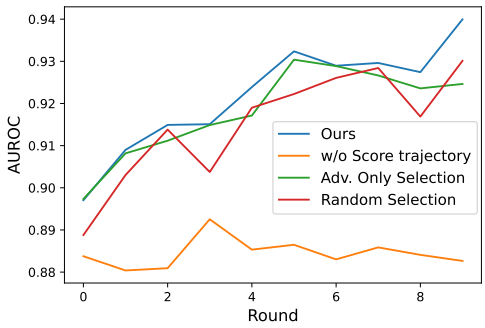


Figure 5: Performance of selection strategies over training rounds on PolitiFact. The x-axis represents the training rounds, while the y-axis represents the detector's AUC.

from candidates instead of using our selection strategy (Sec. 3.3); (3) **Class Prompt**: The method categorizes the confusion scores of conversion prompts into three levels: high, medium, or low. These categorized labels are then used as in-context demonstrations for generating adversarial style-conversion prompts (Sec. 3.2), instead of relying on continuous confusion scores s_c (Eq. 1); (4) **Adversarial only Selection**: The method that selects top-k adversarial prompts, not considering diversity and coherence criteria; (5) **w/o Adversarialness**: The method that omits the adversarialness criterion in our selection strategy; (6) **w/o Coherency**: The method that omits the coherency criterion in our selection strategy; (7) **w/o Score trajectory**: The method without score trajectory component for the style-conversion prompt generation (Sec. 3.2).

Table 3 demonstrates that omitting any component leads to a decrease in performance for certain style conversions attack. Especially, omitting the score trajectory component when selecting the style-conversion prompt proved to be the most detrimental to performance. This suggests that through the style-conversion prompt and confusion score pairs as in-context demonstrations, the LLM can effectively identify conversion prompts that may confuse the detector. In addition, selectively choosing the conversion

prompts that the LLM identifies according to specific criteria led to an additional performance gain. It can be confirmed that our sampling strategy also helps the model to converge faster than other altenrative sampling strategies (See Figure 5).

4.3 In-Depth Performance Analysis

We here conduct analysis on how AdStyle demonstrates robust and high performance across various scenarios and how it effectively enhances the detector's performance.

Comparison with LLM-based baselines. AdStyle enhanced the detector's performance by leveraging the reasoning abilities of advanced large language models like GPT-3.5. To determine whether the capabilities of an advanced large language model alone are sufficient for the fake news detection task, we compared AdStyle with other LLM-based baselines: zero-shot and in-context learning-based inference with GPT-3.5. In the case of the in-context learning baseline, one example each of fake news and real news was provided as in-context demonstrations. The example instruction prompt for the LLM-based baselines are as follows:

> Does the following contain real or fake news? Answer in one word with either 'Real'or 'Fake': [news article]

Figure 6: Instruction prompt for LLM-based baselines. The "news article" section contains the original news text.

Table 4 presents the comparison results on the PolitiFact dataset. Simply using LLMs with prompting makes it challenging to accurately determine the authenticity of news. Rather than directly using the LLM's text generation ability for inference, it is found to be more effective to use it as an augmentation tool to provide additional training signals, as demonstrated by our model.

Table 4: Performance comparison of different LLM-based baselines on the PolitiFact Dataset. (NY: The New York Times, TS: The Sun, NE: National Enquirer)

Model	CNN	NY	TS	NE	Clean
GPT-3.5 zero-shot	0.5820	0.6242	0.6173	0.5274	0.7037
GPT-3.5 in-context	0.6954	0.6504	0.6875	0.5754	0.7383
Ours	0.9174	0.9444	0.9520	0.9460	0.9460

Robustness against a different LLM backbone for attacks. We further assessed the robustness of our method by examining its performance against style conversion attacks when the attacker utilizes a different LLM backbone, such as Gemini-Pro [26]. As shown in Table 5, AdStyle consistently outperforms the baselines even with a different LLM backbone, suggesting that our approach is not overly reliant on recognizing the specific style of content generated by a particular backbone.

Table 5: Comparison under attack scenarios with Gemini-Pro on the PolitiFact dataset.

Model	CNN	NY	TS	NE	Average
Vanilla	0.7913	0.8039	0.8882	0.8397	0.8308
UDA	0.7216	0.7863	0.8657	0.8119	0.7964
RADAR	0.8023	0.7805	0.8764	0.8423	0.8254
ENDEF	0.7730	0.7537	0.8439	0.8058	0.7941
SheepDog	0.8487	0.8926	0.9174	0.8998	0.8896
Ours	0.8821	0.9120	0.9295	0.9241	0.9120

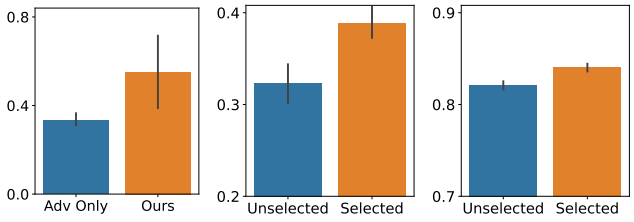
Table 6: Comparison under attack scenarios on the PolitiFact Dataset (A: Adversarial prompt, B: Summarization prompt, C: In-Context prompt D: Adversarial Paraphraser).

Model	A	B	C	D
Vanilla	0.8881	0.8522	0.8896	0.8305
UDA	0.8624	0.8363	0.8924	0.8682
RADAR	0.9096	0.8754	0.9234	0.9007
ENDEF	0.8628	0.7978	0.9009	0.8682
SheepDog	0.9297	0.9205	0.9276	0.8995
Ours	0.9456	0.9416	0.9425	0.9212

Robustness against other possible attack scenarios. To validate our model's robustness against various attacks, we have conducted additional comparison experiments with more diverse attack scenarios to deceive the detector by altering textual style of inputs: (1) Adversarial prompt: Given a news article and its label, the prompt instructs the LLM to rewrite the article to evade detection as the given label. (2) Summarization prompt: A prompt instructing the LLM to summarize the news article without incorporating stylistic guidance. (3) In-Context prompt: A prompt providing an example of a recent, real CNN article, instructing the LLM to rewrite the given article in the same style. (4) Adversarial Paraphraser: The attack utilizes a paraphraser adversarially trained on the training dataset. The example prompt for each attack is described in Figures 12 to 14 in Appendix. Based on the results in Table 6, we confirm that our proposed model still demonstrates a performance improvement compared to other baselines against these attacks.

Effectiveness of the selection strategy. We here empirically verified that our selection strategy (Sec. 3.3) effectively selects diverse style-conversion prompts with high adversarialness and coherency. Figure 7a visualizes the diversity of prompts selected by our method compared to those chosen solely based on adversarialness scores (i.e., Adversarial-only Selection, the third model in our ablation study). We measure diversity using the average cosine similarity of every pair of selected prompts' embeddings, $\mathbf{z}_c$ (Eq. 2):

$$\text{Diveristy}(C) = 1 - \frac{1}{|C|} \sum_{(c_i, c_j) \in C} \text{sim}(\mathbf{z}_{c_i}, \mathbf{z}_{c_j}), \qquad \text{(Eq. 4)}$$



(a) Diversity (b) Adversarialness (c) Coherency

Figure 7: Qualitative analyses with y-axis representing (a) diversity in embedding (z_c) of conversion prompts sampled by AdStyle and adversarial-only selection; (b) Adversarialness (s^c_{adv}) for selected and unselected prompts; and (c) Coherency for selected and unselected prompts.

where C is the set of prompt pairs, sim$(\cdot)$ represents the cosine similarity. The result in the Figure 7a indicate that our strategy results in a more diverse set of augmentations compared to selection based on adversarialness alone.

Figure 7b and 7c illustrate the Adversarialness and Coherency, respectively, of our selected prompts compared to remaining unselected prompts. Adversarialness was measured using the s^c_{adv} score (Section 3.3), and coherency was calculated as the cosine similarity between the original text and its augmented version using semantic BERT embeddings [4]. We can also observe that, for both metrics, prompts selected through AdStyle exhibit higher values compared to unselected prompts. This suggests that our strategy effectively selects prompts by considering both adversarialness and coherency.

Analysis on style-conversion prompts. Finally, we qualitatively analyzed the characteristics of our style-conversion prompts. Figure 8 shows examples of prompts generated and selected by AdStyle. Previous work relied on prompts that were more conventional, such as "neutral" or "sensational." However, our prompts included creative phrases beyond typical human suggestions, which proved to be more adversarial to the detector. Currently, we use a fixed format for the initial set of prompts, which results in the generation of prompts with a similar format. We expect that using a more diverse initial set of prompts will allow the LLM to optimize prompts within a wider search space.

Then, what characteristics of the augmented samples generated from these style-conversion prompts contribute to enhancing the detector's robustness? Interestingly, when we compare the augmented samples from two models: AdStyle and SheepDog, we found that our model produced sentences with significantly higher perplexity according to the language model backbone used by the detector than SheepDog model (9.05 vs. 4.26, see Figure 11b in Appendix), even though ours have lower lexical diversity (0.503 vs. 0.634, see Figure 11a in Appendix). In other words, our generated samples featured sentence structures that the detector's language model likely had not encountered during pretraining. By providing the detector with inputs that have a variety of sentence structures and styles it has not previously seen, the detector naturally becomes more robust against style-conversion attacks. In addition, when we measured the changes in perplexity of the augmented

Round 0
P1: Rewrite the following article in a nonsensical and absurdly exaggerated tone with a hint of horror
P2: Rewrite the following article in a sarcastic and mocking tone
P3: Rewrite the following article in a chaotic and disorganized tone

Round 1
P1: Rewrite the following article in a haunting and macabre tone with a sense of impending horror and madness
P2: Rewrite the following article in a cryptic and enigmatic tone
P3: Rewrite the following article in a malevolent and apocalyptic tone with a sense of impending doom and destruction, while also incorporating elements of surrealism and existential dread

Figure 8: Example adversarial style-conversion prompts selected for the PolitiFact dataset.

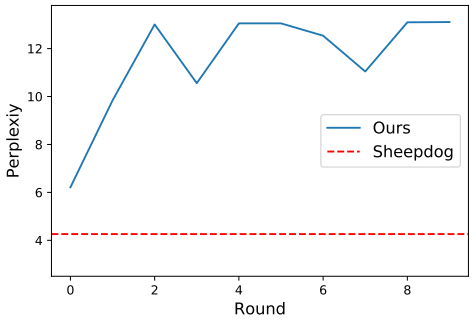


Figure 9: Perplexity changes of augmented texts over the course of training rounds in the Prolific dataset.

samples across training rounds (see Figure 9), the perplexity increased over rounds and eventually converged at a certain point. This indicates that iterative exploration over multiple rounds is effective in creating augmentations that increasingly challenge the detector.

5 CONCLUSION

This paper presents a robust fake news detection method that withstands various style-conversion and paraphrasing attacks through adversarial style-conversion. Unlike traditional detectors that use predefined, agnostic augmentations, AdStyle employs tailored augmentations that shift samples in the direction of the detector's current decision boundary using style-conversion prompts, functioning similarly to adversarial noise. Among the various prompt candidates generated by the LLM, we selected an efficient set of prompts for training by considering diversity, coherency, and adversarialness. As a result, we were able to train a detector that exhibits high robustness and generalizability against a wide range of attacks. We believe that our work helps filter fake news and contribute to a better exchange of information in online.

REFERENCES

[1] David Arthur, Sergei Vassilvitskii, et al. 2007. k-means++: The advantages of careful seeding. In *Soda*, Vol. 7. 1027–1035.

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *Proc. of International Conference on Learning Representations*.

[3] Meeyoung Cha, Chiyoung Cha, Karandeep Singh, Gabriel Lima, Yong-Yeol Ahn, Juhi Kulshrestha, Onur Varol, et al. 2021. Prevalence of misinformation and factchecks on the COVID-19 pandemic in 35 countries: Observational infodemiology study. *JMIR human factors* 8, 1 (2021), e23279.

[4] Sachin Chanchani and Ruihong Huang. 2023. Composition-contrastive Learning for Sentence Embeddings. In *Proc. of Association for Computational Linguistics*. 15836–15848.

[5] Ben Chen, Bin Chen, Dehong Gao, Qijin Chen, Chengfu Huo, Xiaonan Meng, Weijun Ren, and Yang Zhou. 2021. Transformer-based language model fine-tuning methods for COVID-19 fake news detection. In *Combating online hostile posts in regional languages during emergency situation: First international workshop, CONSTRAINT 2021, collocated with AAAI 2021, virtual event, February 8, 2021, revised selected papers 1*. Springer, 83–92.

[6] Canyu Chen and Kai Shu. 2023. Can LLM-Generated Misinformation Be Detected?. In *Proc. of International Conference on Learning Representations*.

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

[8] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234* (2022).

[9] Yaqian Dun, Kefei Tu, Chen Chen, Chunyan Hou, and Xiaojie Yuan. 2021. Kan: Knowledge-aware attention network for fake news detection. In *Proc. of AAAI conference on artificial intelligence*, Vol. 35. 81–89.

[10] Thomas Felber. 2021. Constraint 2021: Machine learning models for COVID-19 fake news detection shared task. *arXiv preprint arXiv:2101.03717* (2021).

[11] Marc Fisher, John Woodrow Cox, and Peter Hermann. 2016. Pizzagate: From rumor, to hashtag, to gunfire in DC. *Washington Post* 6 (2016), 8410–8415.

[12] Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2024. Bad actor, good advisor: Exploring the role of large language models in fake news detection. In *Proc. of the AAAI Conference on Artificial Intelligence*, Vol. 38. 22105–22113.

[13] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: Robust AI-Text Detection via Adversarial Learning. In *Advances in Neural Information Processing Systems*.

[14] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of North American Chapter of the Association for Computational Linguistics*. 4171–4186.

[15] Camille Koenders, Johannes Filla, Nicolai Schneider, and Vinicius Woloszyn. 2021. How vulnerable are automatic fake news detection methods to adversarial attacks? *arXiv preprint arXiv:2107.07970* (2021).

[16] David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. 2018. The science of fake news. *Science* 359, 6380 (2018), 1094–1096.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proc. of International Conference on Learning Representations*.

[18] Ahmadreza Mosallanezhad, Mansooreh Karami, Kai Shu, Michelle V Mancenido, and Huan Liu. 2022. Domain adaptive fake news detection via reinforcement learning. In *Proc. of ACM Web Conference*. 3632–3640.

[19] Qiong Nan, Danding Wang, Yongchun Zhu, Qiang Sheng, Yuhui Shi, Juan Cao, and Jintao Li. 2022. Improving Fake News Detection of Influential Domain via Domain-and Instance-Level Transfer. In *Proc. of International Conference on Computational Linguistics*. 2834–2848.

[20] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[21] Xianghe Pang, Shuo Tang, Rui Ye, Yuxin Xiong, Bolun Zhang, Yanfeng Wang, and Siheng Chen. 2024. Self-Alignment of Large Language Models via Monopolylogue-based Social Scene Simulation. *arXiv preprint arXiv:2402.05699* (2024).

[22] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic Detection of Fake News. In *Proc. of International Conference on Computational Linguistics*. 3391–3401.

[23] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In *Proc. of Association for Computational Linguistics*. 231–240.

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[25] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proc. of Empirical Methods in Natural Language Processing*. 2931–2937.

[26] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024).

[27] Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, and Fabrício Benevenuto. 2019. Supervised learning for fake news detection. *IEEE Intelligent Systems* 34, 2 (2019), 76–81.

[28] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: Explainable fake news detection. In *Proc. of ACM SIGKDD international conference on knowledge discovery & data mining*. 395–405.

[29] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data* 8, 3 (2020), 171–188.

[30] Kathie M d'I Treen, Hywel TP Williams, and Saffron J O'Neill. 2020. Online misinformation about climate change. *Wiley Interdisciplinary Reviews: Climate Change* 11, 5 (2020), e665.

[31] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[32] Jiaying Wu and Bryan Hooi. 2023. Fake News in Sheep's Clothing: Robust Fake News Detection Against LLM-Empowered Style Attacks. *arXiv preprint arXiv:2310.10830* (2023).

[33] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*.

[34] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large Language Models as Optimizers. In *Proc. of International Conference on Learning Representations*.

[35] Zhiwei Yang, Jing Ma, Hechang Chen, Hongzhan Lin, Ziyang Luo, and Yi Chang. 2022. A Coarse-to-fine Cascaded Evidence-Distillation Neural Network for Explainable Fake News Detection. In *Proc. of International Conference on Computational Linguistics*. 2608–2621.

[36] Jingwei Yi, Rui Ye, Qisi Chen, Bin Benjamin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. 2024. Open-Source Can Be Dangerous: On the Vulnerability of Value Alignment in Open-Source LLMs. https://openreview.net/forum?id=NIouO0C0ex

[37] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large Language Models are Human-Level Prompt Engineers. In *Proc. of International Conference on Learning Representations*.

[38] Zhixuan Zhou, Huankang Guan, Meghana Moorthy Bhat, and Justin Hsu. 2019. Fake news detection via NLP is vulnerable to adversarial attacks. *arXiv preprint arXiv:1901.09657* (2019).

[39] Yongchun Zhu, Qiang Sheng, Juan Cao, Shuokai Li, Danding Wang, and Fuzhen Zhuang. 2022. Generalizing to the future: Mitigating entity bias in fake news detection. In *Proc. of ACM SIGIR Conference on Research and Development in Information Retrieval*. 2120–2125.

<EXAMPLES>: [news article]
Determine whether the given <EXAMPLES> content is fact or fake, mentioning the style of the article that led to that decision following format:
Answer: {Fact or Fake}
Reason: {one-sentence reason}

Your current article is: [news article]
But this article is detected as [Answer] due to the following [Reason].
Based on the above information, rewrite a new improved article not to be detected as [news label], maintaining the original content, as follows:

Figure 12: Example input for Adversarial prompt model. First, use the prompt above to extract the answer and reason of authenticity of the given news. Then, the below prompt is used for paraphrasing to avoid detection based on the reason. The "news article" section contains the original news text, and the "news label" section contains the corresponding label.

Summarize the following article, ensuring the content remains the same: [news article]

Figure 13: Example input for Summarization prompt model. The "news article" part is filled with the original news text.

<EXAMPLES>: [news article example] Rewrite the following article as the writing style of <EXAMPLES> : [news article]

Figure 14: Example input for In-Context prompt model. The "news article example" section is filled with an example from a specific publisher, while the "news article" section contains the original news text.

6 APPENDIX

6.1 Baseline Implementation

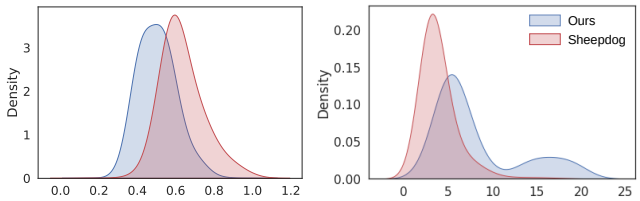
For consistency, experiments replicating existing baselines maintained fixed settings for learning, backbone network, and other relevant parameters. Augmented variations for UDA were generated using back-translation through German. RADAR employed adversarial training to learn paraphrases, utilizing the maximization of the binary cross-entropy loss of a fake news detector as a reward signal, with the T5-large model serving as the paraphraser. Named entities for ENDEF were extracted using the bert-base-NER model from Hugging Face. For SheepDog, augmentations were generated following the original paper's methodology, using the prompt format illustrated in Figure 10 and four tones: "objective and professional," "neutral," "emotionally triggering," and "sensational". Two V100 GPUs were utilized for all experiments.

Rewrite the following article in a/an [tone]: [news article]

Figure 10: Prompt for style-conversion. The "tone" part will be filled with the desired tone, and the "news article" part will contain the original news text.

6.2 Further Analysis on Augmented Samples

We present the results of comparing the textual characteristics of augmented samples generated by our model with those generated by the SheepDog model. Our model produced sentences with overall lower lexical diversity compared to the outputs of the SheepDog model (see Figure 11a). However, when evaluated against the detector's language model, our samples exhibited a significantly higher perplexity distribution (see Figure 11b).



(a) Lexical diversity **(b) Perplexity**

Figure 11: Histogram for lexical diversity and perplexity of augmented samples from two different models - ours and SheepDog.

6.3 Prompt Examples of Other Attack Scenarios

We have conducted additional comparison experiments with more diverse attack scenarios: (1) Adversarial prompt, (2) Summarization prompt, (3) In-Context prompt, and (4) Adversarial Paraphraser. Here we provide the example prompt for each attack prompt in Figures 12 to 14.

6.4 Full Results on Performance Evaluation

Table 7 reports the AUC for each style-conversion attack. Paired sample t-tests were conducted to compare our method with the second-best baseline, Sheepdog, across different ratios for each dataset. Except for the 25% ratio in the Gossipcop dataset, all p-values were significantly smaller than 0.05, indicating statistically significant differences.

Table 7: Performance comparison with AdStyle in four different style—conversion attacks. For the attack scenario, we report the AUC of four style-conversion attacks. Our model demonstrates a significant performance improvement over all text-based fake news detectors in diverse style-conversion attack scenarios. (NY: The New York Times, TS: The Sun, NE: National Enquirer)

CNN	Politifact			Gossipcop			Constraint		
	0.1	0.25	1	0.1	0.25	1	0.1	0.25	1
Vanilla	0.5664	0.6566	0.8127	0.6841	0.7913	0.8518	0.8563	0.8175	0.8056
UDA	0.5906	0.7174	0.8009	0.7294	0.7859	0.8599	0.8305	0.8594	0.9021
RADAR	0.7205	0.7301	0.7926	0.7683	0.8138	0.8666	0.8099	0.8400	0.8873
ENDEF	0.5426	0.6679	0.7588	0.7288	0.7839	0.8624	0.7695	0.8666	0.8353
SheepDog	0.6305	0.8269	0.8683	0.7472	0.8393	0.8706	0.9200	0.9022	0.9544
AdStyle	0.7547	0.8822	0.9174	0.8195	0.8429	0.8744	0.9344	0.9430	0.9646

NY	Politifact			Gossipcop			Constraint		
	0.1	0.25	1	0.1	0.25	1	0.1	0.25	1
Vanilla	0.5899	0.7334	0.8687	0.6750	0.7797	0.8465	0.8522	0.8939	0.8868
UDA	0.6189	0.7644	0.8744	0.7207	0.7795	0.8558	0.8745	0.9184	0.9297
RADAR	0.6966	0.7853	0.8586	0.7481	0.7960	0.8574	0.8701	0.8974	0.9176
ENDEF	0.6152	0.7630	0.8150	0.7227	0.7837	0.8609	0.8341	0.9067	0.8918
SheepDog	0.6310	0.7990	0.9090	0.7420	0.8309	0.8650	0.8937	0.9254	0.9645
AdStyle	0.7599	0.8940	0.9444	0.8118	0.8389	0.8686	0.9243	0.9594	0.9720

TS	Politifact			Gossipcop			Constraint		
	0.1	0.25	1	0.1	0.25	1	0.1	0.25	1
Vanilla	0.6913	0.6859	0.8789	0.7068	0.7918	0.8430	0.7822	0.8727	0.8974
UDA	0.6646	0.8072	0.8883	0.7488	0.7904	0.8613	0.8662	0.9145	0.9446
RADAR	0.7271	0.7173	0.8909	0.7594	0.8062	0.8620	0.8564	0.8882	0.9132
ENDEF	0.7001	0.8132	0.8444	0.7603	0.7909	0.8639	0.8445	0.8955	0.9043
SheepDog	0.7070	0.8408	0.9096	0.7580	0.8395	0.8688	0.8962	0.9217	0.9665
AdStyle	0.8444	0.9002	0.9520	0.8117	0.8347	0.8751	0.9219	0.9521	0.9727

NE	Politifact			Gossipcop			Constraint		
	0.1	0.25	1	0.1	0.25	1	0.1	0.25	1
Vanilla	0.6432	0.6856	0.8591	0.7020	0.7875	0.8399	0.7833	0.8853	0.9067
UDA	0.6557	0.7892	0.8622	0.7450	0.7902	0.8595	0.8747	0.9191	0.9423
RADAR	0.7428	0.7269	0.8864	0.7573	0.8026	0.8556	0.8775	0.8919	0.9161
ENDEF	0.6924	0.7877	0.8355	0.7502	0.7897	0.8589	0.8456	0.9112	0.9025
SheepDog	0.6636	0.8268	0.9166	0.7521	0.8331	0.8632	0.8606	0.9257	0.9667
AdStyle	0.8029	0.8913	0.9460	0.8166	0.8391	0.8702	0.9092	0.9579	0.9771