Kill two birds with one stone: generalized and robust AI-generated text detection via dynamic perturbations

Anonymous ACL submission

Abstract

The growing popularity of large language models has raised concerns regarding the potential to misuse AI-generated text (AIGT). It becomes increasingly critical to establish an excellent AIGT detection method with high generalization and robustness. While, existing methods either focus on model generalization or concentrate on robustness. The unified mechanism, to simultaneously address the challenges of generalization and robustness, is less explored. In this paper, we first empirically reveal an intrinsic mechanism for model generalization and robustness of AIGT detection task. Then, we proposed a novel AIGT detection method (DP-Net) via dynamic perturbations introduced by a reinforcement learning with elaborated reward and action. Experimentally, extensive results show that the proposed DP-Net significantly outperforms some state-of-the-art AIGT detection methods for generalization capacity in three cross-domain scenarios. Meanwhile, the DP-Net achieves best robustness under two text adversarial attacks.

1 Introduction

Recently, the emergence of large language models (LLMs) has significantly enhanced the capabilities of natural language generation. With continuous improvements in model parameters, data scale, and AI-human alignment techniques, these LLMs are now capable of generating text that is grammatically correct, semantically coherent, and very similar to human-written text, making it difficult for humans to distinguish between machine-generated and human-written content. As powerful tools for streamlining content creation, LLMs are widely used across various domains, including journalism, academia, social media, and so on. However, the threats posed by AI-generated text (AIGT), such as academic dishonesty (Wu et al., 2023; Zeng et al., 2024), fake news (Su et al., 2024; Hu et al., 2024), and false comments (Mireshghallah et al., 2024) have raised significant concerns. To prevent the LLM abuse with malicious purpose, numerous AIGT detection methods have been proposed, utilizing the specific feature differences between human-written text and AI-generated text.

Mainstream AIGT detection methods typically fall into one of two categories: white-box or black-box detection, depending on whether the generator's parameters are accessible. While some white-box methods (Sebastian et al., 2019; Wang et al., 2023; Yang et al., 2024; Li et al., 2023) achieve high-precision detection by extracting textual features from the generator's output logits, black-box AIGT detection methods (Corizzo and Leal-Arenas, 2023; Soto et al., 2024; Tian et al., 2024) are gaining increasing attention for their ability to operate without accessing the model's parameters, making them particularly relevant in the context of many closed-source commercial LLMs.

Current black-box AIGT detection methods learn deep features on large annotated datasets, preforming well when training and testing samples are independent and identically distributed (IID). However, the detection of out-of-distribution (OOD) samples is inevitable in real-world application scenarios. Therefore, some researchers are concentrated on enhancing generalization capacity for cross-domain detection (Liu et al., 2023b; Bhattacharjee et al., 2023). These methods aim to enhance generalization in unknown target domains through feature alignment to improve detection performance. Some studies (Mitchell et al., 2023; Zhu et al., 2023) introduce zero-shot learning methods to achieve domain-generalization AIGT detection in unpredictable target distributions by extracting effective domain-invariant features.

To evade detection, users may attempt to modify the generated text when using AI tools. However, current AIGT detectors face challenges in detecting minor adversarial perturbations (Zhou et al., 2024; Krishna et al., 2023). Therefore, some research has

explored robust AIGT detection methods (2023; 2024). Despite the good generalization ability of existing AIGT methods, they are vulnerable for text adversarial attacks. Meanwhile, some robust detection method can not achieve good generalization for OOD data. An interesting and challenging problem is how to construct a comprehensive framework that exhibits high generalization and robustness against text adversarial attack.

In this paper, we argue that model generalization and robustness are intrinsically linked. Both concepts can be viewed as responses to variations introduced in the source domain. Specifically, robustness is often validated by introducing minor perturbations within the source domain, while OOD data can be understood as changes in the distribution resulting from more substantial and specific disturbances. The analysis spontaneously induce an evident thinking: *Can adding elaborated perturbations simultaneously improve generalization and robustness of AIGT detection network?*

In order to exploring the rationality of the above thinking, we first empirically reveal an intrinsic mechanism for model generalization through making quantitative and qualitative experiments. Then, in light of the intrinsic mechanism, we propose a novel AIGT detection method via dynamic perturbations (called DP-Net). The crux of DP-Net is that dynamic perturbations, added into embedding matrix in training phase, are yielded by a reinforcement learning with elaborated reward and action. Extensive experimental the proposed PD-Net can significantly improve the generalization and robustness for AIGT detection task. Our contributions are as follows:

- We empirically reveal an intrinsic mechanism for generalization capacity of AIGT detection. i.e., adding slight perturbations into source domain can effectively simulate the domain shift between source and target domains.

- We propose a novel AIGT detection method (DP-Net) via dynamic perturbations which are adaptively yielded by the reinforcement learning with elaborated reward and action.

- Extensive experiments demonstrate that the proposed DP-Net achieves state-of-the-art generalization in three cross-domain scenarios and achieves best robustness under two text adversarial attacks.

2 Related Work

2.1 Supervised Detector

Supervised classifiers are trained on large labeled datasets to extract representations that can effectively distinguish between two classes of text samples. For instance, Zhong et al. (2020) combined the actual structure of text with a classifier based on RoBERTa (Liu et al., 2020) to improve detection accuracy. Tian et al. (2024) introduce a length-sensitive Multiscale Positive-Unlabeled Loss, which enhances the detection performance of short texts while maintaining the detection efficacy for long texts. These methods achieve strong detection performance in detecting datasets belonging to the same domain as the training set, but usually fail when faced with datasets that are not in the domain of the training set. To enhance the model's generalization ability in known target domain, Liu et al. (2023b) combined maximum mean discrepancy with contrastive learning to obtain domain-invariant representations, facilitating adaptation of classifiers from source to target generators. Verma et al.(2024) proposed Ghostbuster, a method that processes documents through a series of weaker language models, conducts a structured search over possible combinations of their features, and then trains a classifier on the selected features to predict whether the documents are AI-generated.

2.2 Zero-shot Detector

With the widespread use of LLMs, AIGT detectors often encounter unknown domain detection in practical applications, which has sparked a growing interest in zero-shot domain generalization detectors. Mitchell et al. (2023) proposed a text perturbation method to measure the log probabilities difference between original and perturbed texts. Su et al.(2023) proposed a zero-shot method that measures the log-probability difference between original text and perturbed text using text perturbation techniques, significantly improving AIGT detection performance. Venkatraman et al.(2024) argue that humans tend to evenly distribute information during language production, whereas AI-generated text may lack this uniformity. Therefore, they introduce uniform information density features to quantify the smoothness of token distribution, aiding in the identification of AI-generated text.

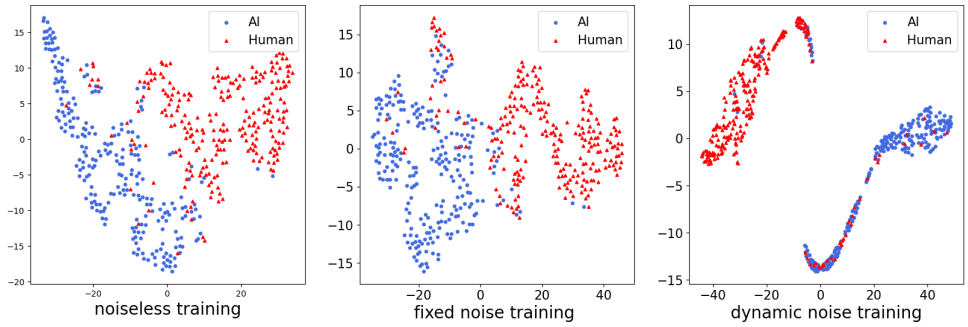

Figure 1: Visualization results of features extracted by detectors with different training strategies.

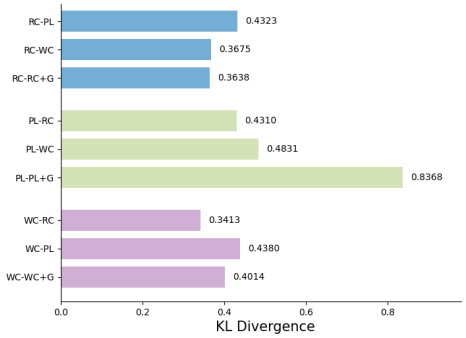

Figure 2: KL divergence between different distributions.X-Y represents the distance between the two distributions X and Y, while "RC", "PL", and "WC" refer to Reddit ChatGPT, PeerRead Llama, and WikiHow ChatGPT, respectively. Additionally, "+G" indicates the addition of Gaussian noise.

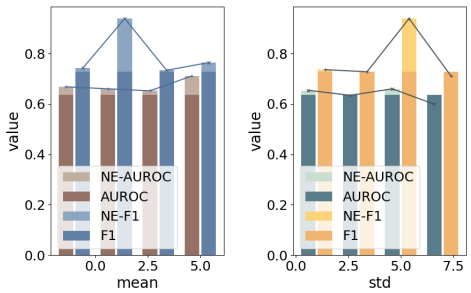

Figure 3: The impact of adding noise on the generalization performance of model.

3 Methodology

3.1 intrinsic mechanism analysis

To evaluate the rationality of the intrinsic mechanism. i.e.: adding slight perturbations into source domain can effectively simulate the domain shift between source and target domains, we make some quantitative and qualitative experiments. For the quantitative experiment, we leverage the Roberta network to extract semantic spaces of different domains, and then Kullback-Leibler Divergence (KLD) is used to measure domain shift of the different semantic spaces. As shown in Figure 2, we can find that the KLD of “RC-RC+G” is similar to those of other two cross-domain scenarios, i.e. “RC-WC” and “RC-PL” from the blue lines. The trend are also exhibited by the experimental results from the pink lines. While, the results shown by green lines have obvious KLD gap, which is different from other two. The experimental results demonstrate that adding slight perturbations to the source domain effectively simulates the domain shift between different domains. In addition, since different cross-domain scenarios have different domain shift, the network can learn excellent domain-invariant features, adding dynamic perturbations in the training phase.

For the qualitative experiment, we validate the impact of noise on the detection performance of the model in unknown domains, and the experimental results are shown in Figure 3. The experiments demonstrate that introducing dynamic perturbations during the training process can effectively improve model generalization. Additionally, a single type of noise cannot guarantee optimal performance of the model in unknown target domain. We also obtain similar results by visualizing the features extracted by detectors using different training strategies, as shown in Figure 3. It is found that a single type of noise cannot guarantee an improvement in the detector’s performance in unknown domains. Therefore, we continuously adjust the noise distribution during training through reinforcement learning to encourage the model to learn more generalized feature representations. As shown in Figure 3, by introducing a reinforcement learning joint training strategy, the model is able to achieve clearer decision boundaries in unknown target domains.

According to the above quantitative and qualita-

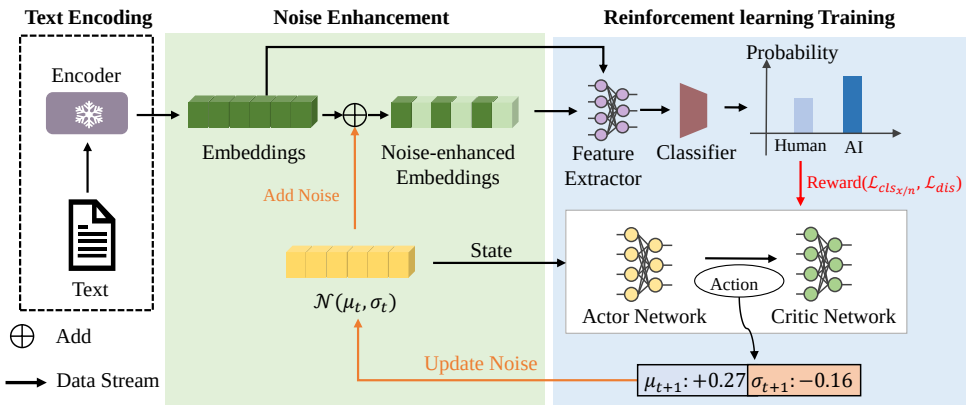

Figure 4: The overall framework of DP-Net. This framework includes three modules: text encoding, noise enhancement, and reinforcement learning training.

tive analysis, we proposed a novel AIGT detection method via dynamic perturbations (DP-Net) mainly consists of three modules: Text Encoding, Noise Enhancement and Reinforcement Learning Training. The detail architectures of the DP-Net are introduced as following sections.

3.2 Text Encoding and Noise Enhancement

We introduce noise-enhanced samples during training to improve the generalization and robustness of the detector. The overall process is as follows.

First, text x consisting of n tokens is encoded by a pre-trained RoBERTa (Liu et al., 2020) with fixed parameters to obtain text embedding $\mathbf{E}_x$ in text encoding module.

Next, we randomly generate a Gaussian noise $\mathcal{N}(\mu, \sigma^2)$ with its mean μ and variance σ^2 within a specified range. Then the noise is added to the embedding of input text $\mathbf{E}_x$ to obtain the noise-enhanced embedding $\mathbf{E}_n$:

$$\mathbf{E}_n = \mathbf{E}_x + \mathcal{N}(\mu, \sigma^2) \tag{1}$$

After that, the original embedding and the noise-enhanced embedding are input into a feature extractor to obtain low-dimensional classification representations $\mathbf{z}_x$ and $\mathbf{z}_n$, respectively:

$$\mathbf{z}_{x/n} = Extractor(\mathbf{E}_{x/n}) \tag{2}$$

To improve the ability of detector to align noisy samples with original samples, we introduce a distance loss L_{dis} to minimize the distance between the two embeddings:

$$L_{dis} = \|\mathbf{z}_x - \mathbf{z}_n\|_2^2 \tag{3}$$

Then $\mathbf{z}_x$ and $\mathbf{z}_n$ are processed by a $softmax$ function to obtain the predicted probabilities $\hat{\mathbf{y}}_x$ and $\hat{\mathbf{y}}_n$, respectively, After that, two cross-entropy losses L_{cls_x} and L_{cls_n} are incorporated to increase prediction accuracy.

$$L_{cls_{x/n}} = -\left[\mathbf{y}\log(\hat{\mathbf{y}}_{x/n}) + (1-\mathbf{y})\log(1-\hat{\mathbf{y}}_{x/n})\right] \tag{4}$$

where $\mathbf{y}$ is the true label of text x. The final model loss L is defined as:

$$L = \lambda_1 \cdot L_{cls_x} + \lambda_2 \cdot L_{cls_n} + \lambda_3 \cdot L_{dis} \tag{5}$$

where λ_1, λ_2 and λ_3 are adjustable hyper-parameters.

3.3 Reinforcement Learning Training

To enhance sample diversity and guide the model in finding more suitable noise parameters for the unknown target domain, we introduce reinforcement learning. Inspired by Liu et al. (2023a), we treat the mean and variance of Gaussian noise during training as a continuous control task. We employ an actor-critic algorithm called Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2019), which leverages deep neural networks to learn deterministic policies in continuous action spaces to control the noise distribution, aiming to maximize cumulative rewards through a combination of off-policy learning and target network stabilization.

We design a reward function based on the training loss of original samples and noise-enhanced samples to guide this process. The reward function is defined as:

$$reward(L) = \begin{cases} -log(L-\epsilon) & L-\epsilon > 0 \\ e^{\epsilon - L} & L-\epsilon \leq 0 \end{cases} \tag{6}$$

where ϵ is an adjustable threshold hyper-parameter.

We store the tuple (s, a, r, s') into the experience replay buffer $\mathcal{R}$, where s represents the current

noise distribution state, a is the action taken, s' is the subsequent state resulting from the action, and r is the reward received. Next, DDPG is used to optimize the policy by maximizing the Q-function, aiming to maximize the long-term cumulative reward for actions taken in all states.

For each sampled tuple, the target Q-value is computed using the target networks:

$$y_i = r_i + \gamma Q'(s'_i, \mu'(s'_i|\theta^{\nu'})|\theta^{Q'}) \tag{7}$$

where γ is the discount factor, Q' and ν' are the target Q and policy networks, respectively. The parameters θ^Q of the Q-network are then updated by minimizing the loss function $L(\theta^Q)$:

$$L(\theta^Q) = E_{(s,a,r,s')}[(Q(s_i, a_i|\theta^Q) - y_i)^2] \tag{8}$$

The policy network ν is updated using the deterministic policy gradient:

$$\nabla_{\theta^\nu} J(\theta^\nu) = E_{s\sim\mathcal{R}}\left[\nabla_a Q(s, a \mid \theta^Q)\Big|_{a=\nu(s)} \nabla_{\theta^\nu}\mu(s \mid \theta^\nu)\right] \tag{9}$$

This update seeks to maximize the expected return by adjusting the parameters θ^ν of the policy network.

The target networks' parameters are updated using a soft update strategy:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \tag{10}$$

$$\theta^{\nu'} \leftarrow \tau\theta^\nu + (1-\tau)\theta^{\nu'} \tag{11}$$

where $\tau \ll 1$ is a small parameter that ensures slow updates to the target networks, thereby providing stability to the training process.

We illustrate the general process in pseudo Algorithm 1.

Algorithm 1 DP-Net Algorithm

Input: source domain training data $D = \{X, Y\}$
Parameter: Maximum number of iterations $max_episode$, exploration steps per episode max_step, loss weight $\lambda_1, \lambda_2, \lambda_3$, reward threshold ϵ
Output:Trained detector $Detector$

```
env ← Env() # Initialize environment
agent ← DDPG() # Initialize reinforcement learning agent
for episode in max_episode do
    Initial current state s
    for step in max_step do
        Retrieve the current action a based on s
        Act based on the s to obtain the next state s'.
        for all x sampled from D do
            Predict all x based on s' to obtain the loss L
            with Eq.5 and the reward r with Eq.6
            Update encoder Detector based on L
        end for
        Store s, a, r and s' in experience replay buffer R
        update politic network ν and critic network Q
        s ← s' # update state
    end for
end for
return Trained detector Detector
```

Corpus	Human	Davinci003	ChatGPT	FlanT5
Wikipedia	3,000	3,000	2995	0
Reddit	3,000	3,000	3,000	3,000
WikiHow	3,000	3,000	3,000	0
peerread	5,798	2,344	2,344	0
arXiv	3,000	3,000	3,000	3,000

Table 1: Sample numbers from different sources in M4.

4 Experiments

4.1 Dataset

We use the M4 (Wang et al., 2024) as the benchmark dataset to construct various cross-domain scenarios. M4 is a large-scale, multi-domain corpus generated by three LLMs for the AIGT detection task. We utilize a subset of datasets from M4, with detailed information provided in Table 1.

In the following experiments, we denote the data sources by combining the corpus name with the generator name. For example, 'Arxiv ChatGPT' signifies that the corpus consists of Arxiv abstracts (arXiv.org submitters, 2024), and the AI-text generator used for this corpus is ChatGPT (Ouyang et al., 2022).

4.2 Baseline Methods

We compared five benchmark methods for black-box AIGT detection:

Naive Classifer is a supervised method, which consist of a fixed RoBERTa model with an additional MLP classification head. The training process is consistent with Roberta.

Roberta (Liu et al., 2020) is a supervised method. In this work, we fine-tune the parameters on the labeled training set and perform detection in the target domain.

SCRN (Huang et al., 2024) employs a reconstruction network to add and remove noise from text, extracting a semantic representation that is robust to local perturbations. It uses two random noise sources to enhance the representation and learns through twin networks. In contrast, our DP-Net approach uses reinforcement learning to continuously modify the noise distribution during training.

GLTR (Sebastian et al., 2019) proposes three simple tests to assess whether the text is generated in a specific assumed manner. In this work, we use the most powerful Test-2 feature, which is the

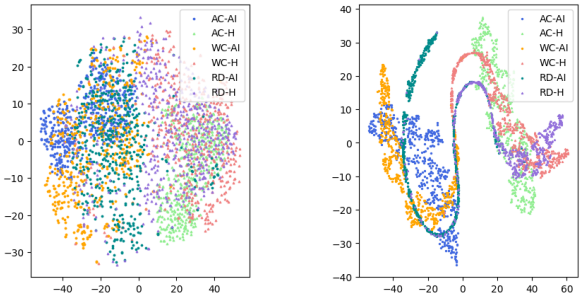

Figure 5: Visualization of Domain Generalization Detection Results. The left figure represents the distribution of the original texts, while the right figure shows the distribution of the texts obtained through the trained encoder.

absolute rank of a word, consistent with Guo et al. (Guo et al., 2023).

Fast-DetectGPT (Bao et al., 2024) is an optimized zero-shot detector, which utilize conditional probability curvature to elucidate discrepancies in word choices between LLMs and humans within a given context. In this study, both the sampling model and the scoring model used in the method are GPT-2 (Radford et al., 2019).

4.3 Experimental Setting

During the training process, the agent continuously adjusts the noise distribution based on the reward at each step and optimizes both the actor and critic networks. Simultaneously, the detector is optimized based on the model's final loss. The optimizers for all networks in the model are Adam, with a learning rate of $3e^{-4}$ for the actor and critic networks and $8e^{-5}$ for the encoder. A learning rate decay strategy is introduced to gradually reduce the learning rate, promoting model convergence. The model is trained for 300 epochs on the source domain data, which is Arxiv ChatGPT, and tested on different unseen target domains in this experiment. The weight parameters used during training are $\lambda_1 = 0.5$, $\lambda_2 = 0.5$, $\lambda_2 = 0.01$ and the threshold $\epsilon = 1$. In the experiments involving reinforcement learning, the reported results are averaged across five random seeds. The GPU used for the experiments is an NVIDIA RTX 4060.

4.4 Domain Generalization Results

We test the generalization ability of our proposed DP-Net on seven target domains, which include three different cross-domain scenarios: same generator across different corpora, same corpus across different generators, and different corpora across different generators.

From the Table 2, DP-Net achieves an average detection accuracy of 86.10% across seven unseen target domains, surpassing the other baseline methods by at least 5.55% and demonstrating state-of-the-art domain generalization performance. This is because zero-shot detectors like Fast-detectGPT heavily rely on the selection of proxy models, and different proxy models can significantly impact model performance. However, supervised detectors based on training data, such as Roberta, suffer from the lack of target domain data. Training solely on source domain data prevents the model from extracting effective domain-invariant features, resulting in poorer detection performance on unseen domains.

As seen from the visualization results in Figure 5, DP-Net effectively pulls together texts of the same class from different domains while pushing apart texts from different classes. This indicates that DP-Net has the generalization ability to perform detection in new, unseen domains.

4.5 Cross-Domain Adversarial Robustness

We further evaluate the model's robustness under Adversarial attacks in cross-domain scenarios. We perform synonym replacement with a replacement ratio of 0.2 and paraphrase attacks on the test data from the unknown target domain. In the paraphrasing attack, we use Flan-T5-base (Chung et al., 2022) to rewrite AI-generated text.

Based on Table 3, our proposed method demonstrates superior robustness against attacks. It achieves the highest cross-domain detection accuracy under both types of attacks compared to other baseline methods. Specifically, under the synonym substitution attack, our method achieves an average detection accuracy of 91.25% by introducing Gaussian noise, surpassing other baseline models. For the paraphrase attack, which significantly alters text distribution, our method attains an average cross-domain detection accuracy of 68.73%, representing a 9.46% improvement over the second-best baseline model.

4.6 Ablation Study

To examine the impact of noise types on model generalization, we compared the average detection accuracy across seven unseen domains under different noise settings, both with and without Joint Reinforcement Learning Training Strategy. The results shown in Figure 6 indicate that the model's gen-

Method	Naive Classifier		Roberta		SCRN		GLTR		Fast detectGPT		DP-Net +U(ours)		DP-Net +G(ours)	
Target Domain	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Wikihow-C	64.00	72.89	54.90	68.35	51.10	44.67	44.30	55.48	**79.90**	**79.08**	68.52	61.19	66.62	57.34
Wikipedia-C	64.80	73.96	53.90	68.45	53.90	49.58	70.00	76.45	95.40	95.59	**96.88**	**96.97**	96.04	96.18
Reddit-C	63.10	73.05	51.90	67.51	44.92	45.98	54.90	68.62	**91.90**	**91.91**	88.10	88.41	89.62	90.17
Arxiv-D	94.70	94.10	98.00	97.96	51.50	45.37	78.10	76.87	36.00	35.48	90.94	89.89	89.52	88.27
Arxiv-F	98.40	98.37	**99.80**	**99.80**	51.04	63.72	83.30	83.18	82.50	80.83	91.66	90.82	90.84	89.91
Peerread-D	60.62	64.83	47.28	57.93	49.80	45.98	46.64	54.70	**97.12**	97.17	76.15	75.35	77.62	75.93
Reddit-D	63.10	73.05	51.90	67.52	51.00	44.60	55.00	68.71	82.00	81.93	86.10	86.14	**87.72**	**88.04**
Average	72.67	78.61	65.38	75.36	51.04	48.56	61.75	69.14	80.55	80.28	85.48	84.11	**86.10**	**83.69**

Table 2: Comparison of Models across Different Domains. "+U" indicates adding uniform distribution noise, and "+G" indicates adding Gaussian noise. "-C","-D","-F" indicate that AIGT is generated by ChatGPT, Davinci, and FlanT5, respectively.

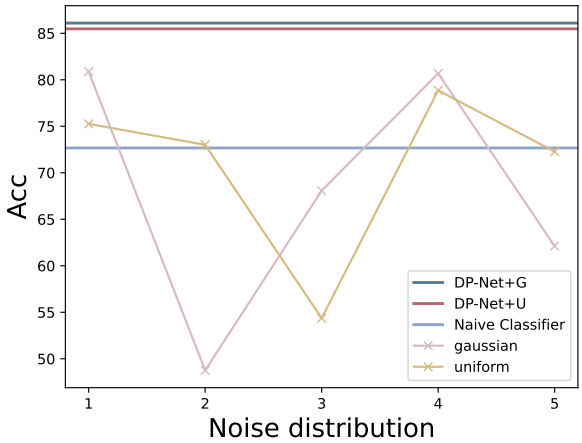

Figure 6: The impact of different noise distribution types on model generalization. × represents the average detection accuracy across 7 unkown domains when the model is trained with noise-augmented samples without using RL.

eralization performance varies significantly across different noise settings. By introducing reinforcement learning, the detection accuracy of the model trained with noise-enhanced samples improved significantly. Additionally, as shown in Table 2 and Table 3, the model trained with Gaussian noise shows better generalization and robustness against attacks compared to uniform distribution noise. We believe this is because Gaussian noise causes significant changes in the original sample distribution, thereby increasing the diversity of the training sample distribution. This, in turn, enhances the model's generalization and robustness after reinforcement training.

Next, we demonstrate the effectiveness of the DP-Net structure by adding Gaussian noise through two variants:

DP-Net+G$_{-rl}$: Only noise-enhanced training is performed.

DP-Net+G$_{-encoder}$: The encoder is not trained simultaneously with reinforcement learning. Instead, a two-stage approach is adopted. In this approach, the encoder parameters are fixed and used as the environment at first. Reinforcement learning is employed to find the optimal distribution, after which the encoder parameters are trained on the source domain data based on the optimal distribution.

We trained DP-Net and its variants on data from Arxiv ChatGPT, with the results shown in Table 4.

We find that directly adding noise to train the classifier results in a relatively low average cross-domain detection accuracy. This may be due to the significant differences between domains, making it difficult for a single noise sample to handle detection in various domain shift scenarios. For instance, in our experiments, DP-Net+G$_{-rl}$ achieves the highest detection accuracy on the test data sampled from Arxiv Davinci but performs poorly in other domains. Additionally, the two-step method is also affected by a single noise distribution. Although the optimal noise for the current environment is obtained through reinforcement learning initially, the model's generalization remains lower than that of DP-Net due to the continuous changes in the training environment and the fixed noise distribution.

We also explore the data efficiency of DP-Net. We use 100% and 75% of the training data to train the model, and compare it with naive classifier,

Target domain	Wikipedia ChatGPT		Reddit ChatGPT		Reddit Davinci		Average	
Attack Method	Synonym replacement (ratio=0.2)							
Method	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1
Roberta	52.70	67.89	90.00	89.29	79.10	82.21	73.93	79.80
SCRN	49.58	44.92	49.55	45.90	48.88	44.82	49.34	45.21
GLTR	79.30	76.66	64.39	50.38	63.57	67.11	69.09	64.72
Fast-DetectGPT	90.39	82.72	83.01	76.30	68.96	64.72	80.79	74.58
DP-Net + U (ours)	91.58	91.22	83.39	62.02	83.44	79.83	86.98	84.81
DP-Net + G (ours)	**94.60**	**94.61**	**90.32**	**89.51**	**88.84**	**87.74**	**91.25**	**90.62**
Attack Method	Paraphrase							
Method	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1
Roberta	53.90	68.45	64.90	65.82	51.70	**67.39**	56.83	**67.22**
SCRN	49.88	45.16	49.20	45.37	48.63	44.44	49.24	44.99
GLTR	56.40	66.84	62.61	**66.71**	58.81	66.71	59.27	66.75
Fast-DetectGPT	61.76	64.35	31.32	35.26	23.01	28.09	38.70	42.57
DP-Net + U (ours)	75.50	69.62	**68.44**	62.02	61.58	50.54	68.51	60.73
DP-Net + G (ours)	**76.50**	**71.93**	67.92	62.02	**61.76**	51.37	**68.73**	61.77

Table 3: Evaluation of cross-domain adversarial robustness. The source domain is Arxiv ChatGPT. The optimal results are indicated in bold.

Target Domain	Wikipedia ChatGPT		Reddit ChatGPT		Arxiv Davinci		Peerread Davinci		Average	
	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1
DP-Net+G (ours)	**96.04**	**96.18**	**89.62**	**90.17**	89.52	**88.27**	**82.36**	**75.39**	**89.39**	**87.50**
DP-Net+G$_{-rl}$	67.50	75.47	66.00	74.63	**94.00**	93.62	70.23	65.58	74.43	77.33
DP-Net+G$_{-encoder}$	71.00	77.52	75.00	80.00	88.50	87.01	73.79	68.49	77.07	78.26

Table 4: Ablation Experiment Results. The source domain is Arxiv ChatGPT. The optimal results are indicated in bold.

which is the DP-Net variant after removing noisy enhanced samples and RL, also trained with only 75% of the training data. The results indicate that using only 75% of the training data still achieves results comparable to those obtained using the entire training set. In contrast, reducing the training samples significantly degrades the performance of the Naive Classifier, demonstrating that our proposed DP-Net has high data efficiency. Specific experimental results can be found in Appendix A.

Additionally, we compare DP-Net with a different RL method, Deep Q-Network (DQN). In this approach, we treat the changes of mean and variance in noise distribution as a discrete control task while keeping all other settings the same as the DDPG method. The experiments show that DQN, with its discretized actions, exhibits more stable rewards after multiple iterations than DDPG. The results demonstrate that the generalization performance of both methods on the six unseen domains is comparable. Specific experimental results can be found in Appendix B.

5 Conclusion

We have proposed a novel AIGT detection method via dynamic perturbations (DP-Net) to simultaneously improve the model generalization and robustness against text adversarial attack. The DP-Net combines RL with a noise-enhanced training strategy. It adaptively introduces noise-enhanced samples to train the encoder, enhancing model generalization and robustness. Experiments show that the DP-Net performs excellently in robustness tests, maintaining high average detection accuracy under attacks in cross-domain situations. By continuously adjusting the distribution through RL, the introduction of Gaussian noise enables the model to achieve better generalization and robustness compared to uniform noise.

6 Limitation

The revealed intrinsic mechanism dose not be proved by explicit mathematical expression, which unable to accurately guide the perturbation design, hindering the performance improvement. In addition, for the case of multiple source domains, the proposed DP-Net has not comprehensively explored.

References

arXiv.org submitters. 2024. arxiv dataset.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv preprint*.

Roberto Corizzo and Sebastian Leal-Arenas. 2023. One-gpt: A one-class deep fusion model for machine-generated text detection. In *2023 IEEE International Conference on Big Data (BigData)*, pages 5743–5752.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *Preprint*, arXiv:2301.07597.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(20):22105–22113.

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*, volume 36, pages 15077–15095. Curran Associates, Inc.

Guanhua Huang, Yuchen Zhang, Zhe Li, Yongjian You, Mingze Wang, and Zhouwang Yang. 2024. Are ai-generated text detectors robust to adversarial perturbations? *Preprint*, arXiv:2406.01179.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. *Preprint*, arXiv:2307.11729.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Frederick Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. 2023. Origin Tracing and Detecting of LLMs. *arXiv e-prints*, arXiv:2304.14072.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2019. Continuous control with deep reinforcement learning. *Preprint*, arXiv:1509.02971.

Sicong Liu, Xi Sheryl Zhang, Yushuo Li, Yifan Zhang, and Jian Cheng. 2023a. On the data-efficiency with contrastive image transformation in reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2023b. CoCo: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16167–16188, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Niloofar Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. 2024. Smaller language models are better zero-shot machine-generated text detectors. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 278–293, St. Julian's, Malta. Association for Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023.

Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

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):9.

Gehrmann Sebastian, Strobelt Hendrik, and Rush Alexander. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Rafael Alberto Rivera Soto, Kailin Koch, Aleem Khan, Barry Y. Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-shot detection of machine-generated text using style representations. In *The Twelfth International Conference on Learning Representations*.

Jinyan Su, Claire Cardie, and Preslav Nakov. 2024. Adapting fake news detection to the era of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1473–1490, Mexico City, Mexico. Association for Computational Linguistics.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412, Singapore. Association for Computational Linguistics.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, QINGHUA ZHANG, Ruifeng Li, Chao Xu, and Yunhe Wang. 2024. Multiscale positive-unlabeled detection of AI-generated texts. In *The Twelfth International Conference on Learning Representations*.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2024. GPT-who: An information density-based machine-generated text detector. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 103–115, Mexico City, Mexico. Association for Computational Linguistics.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717, Mexico City, Mexico. Association for Computational Linguistics.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1144–1156, Singapore. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. LLMDet: A third party large language models generated text detection tool. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2113–2133, Singapore. Association for Computational Linguistics.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. In *The Twelfth International Conference on Learning Representations*.

Zijie Zeng, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guangliang Chen. 2024. Towards automatic boundary detection for human-ai collaborative hybrid essay in education. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(20):22502–22510.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2461–2470, Online. Association for Computational Linguistics.

Ying Zhou, Ben He, and Le Sun. 2024. Humanizing machine-generated content: Evading AI-text detection through adversarial attack. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8427–8437, Torino, Italia. ELRA and ICCL.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. 2023. Beat LLMs at their own game: Zero-shot LLM-generated text detection via querying ChatGPT. In *Proceedings of the*

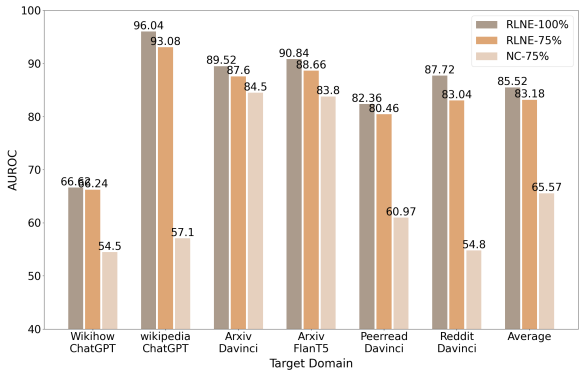

Figure 7: Data efficiency experiment. NC-75% indicates using 75% of the training set to train the Naive classifier, which is the DP-Net variant after removing noisy enhanced samples and RL. The source domian is Arxic ChatGPT.

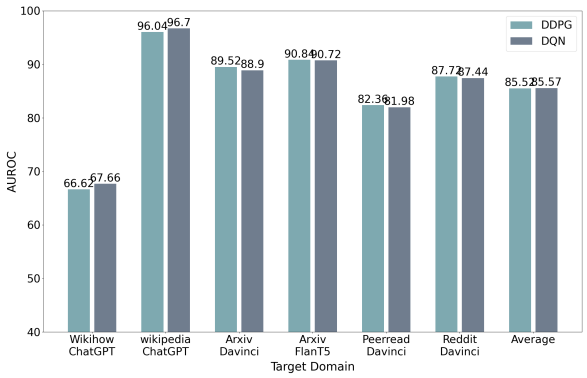

Figure 8: Detection result on 6 different domains using different reinforcement learning methods to train models. The source domian is Arxic ChatGPT.

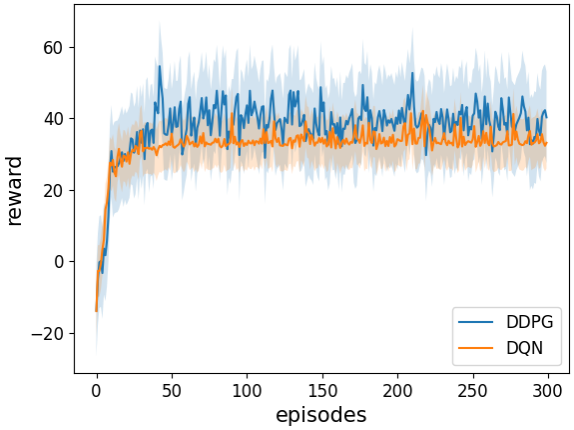

Figure 9: Comparison of reward between DDPG and DQN methods.

2023 Conference on Empirical Methods in Natural Language Processing, pages 7470–7483, Singapore. Association for Computational Linguistics.

A Data Efficiency

We trained the model using both 100% and 75% of the training data and compared its performance to that of a naive classifier. The naive classifier, which is an DP-Net variant without the noisy enhanced samples and reinforcement learning, is also trained using only 75% of the data. The experimental results are shown in the figure 7. In our DP-Net implementation for this experiment, we introduce Gaussian noise.

The results indicate that using only 75% of the training data still achieves results comparable to those obtained using the entire training set, with only a 2.34% decrease in performance. In contrast, reducing the training samples significantly degrades the performance of the naive classifier, which drops from 85.52% with the full training set to 65.57%. This result indicates that our proposed DP-Net demonstrates high data efficiency.

B Comparison of Different Reinforcement Learning Methods.

We compare DP-Net with a different reinforcement learning method, Deep Q-Network (DQN), which does not use an actor-critic framework. Deep Q-Network (DQN) is a reinforcement learning algorithm that combines Q-learning with deep neural networks. It approximates the action-value function, which estimates the potential future rewards of taking certain actions in given states. Additionally, DQN employs a target network, which is periodically updated, to provide more consistent target values during the learning process.

As shown in the figure 8, The models trained using the two reinforcement learning methods exhibit comparable generalization performance. Additionally, as shown in Figure 9, the use of discrete action distributions simplifies the environment, resulting in more stable rewards over 300 iterations when using DQN compared to DDPG.