



北京理工大学
信息系统及安全对抗实验中心
信息安全与对抗技术研究所
Beijing Forest Studio



机生文本检测中的扰动增强

www.isclab.org.cn

硕士研究生 李梓皓

2026年10月25日

- 相关内容

- 邢倚康《从生成机制探索机生文本检测新方法》——2026.01.03
- 邢倚康《机生文本检测中的对抗攻击与鲁棒防御》——2026.07.19

- 预期收获
- 题目内涵解析与研究目标
- 研究背景与意义
- 研究历史与现状
- 知识基础
- 算法原理
 - DP-Net
 - RAMP-Net
- 特点总结与工作展望
- 参考文献

- 预期收获
 - 掌握机生文本检测的基本概念与常见方法
 - 理解跨域与对抗场景下检测失效的成因
 - 了解动态扰动与流形约束两条利用扰动训练的技术路线

- 内涵解析

- 机生文本检测：对大模型生成的文本进行识别，区分人类撰写文本与机器生成文本
 - 跨领域：检测方法在**训练域和检测域**不一致时仍能有效工作
 - 跨模型：检测方法对**未见过/未知来源**大模型生成文本仍具备识别能力
- 动态扰动：在文本表示上加入强度可调的随机变化，根据反馈调整，而非固定噪声

- 研究目标

- 面向大模型时代的机器生成文本检测任务
- 结合**深度学习、自然语言处理**等技术
- 围绕训练期扰动这一技术路线，分析算法的设计思路
- 提高检测器面对已知攻击、未知攻击及不同攻击强度时的鲁棒性与泛化能力，为学术诚信、内容审核和媒体平台治理提供更加可靠的检测能力

- 研究背景

- 机生文本带来了**学术不端、虚假信息传播**等风险，检测与标识成为合规刚需
- 现有检测方法在训练域内表现良好，但面对**模型迭代、领域变化与改写攻击**时性能显著下降。目标域标注数据昂贵，为每个新领域重训检测器不现实，需要一种不依赖目标域标签、在训练阶段注入鲁棒性的手段

- 研究意义

- 把扰动变为训练的增强手段，用源域数据换取跨域泛化与对抗鲁棒，降低跨领域检测的适配成本
- 为学术诚信、内容审核与平台治理提供更稳的检测能力，支撑起AIGC标识能力与溯源合规

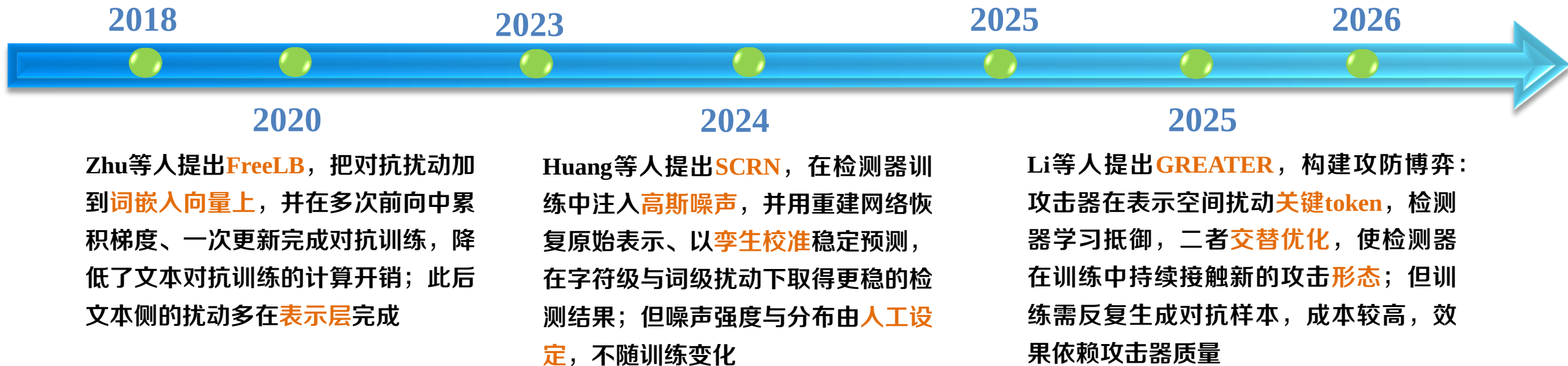


Madry等人将对抗训练形式化为“在扰动预算内搜索**最坏扰动**再训练”，**PGD**由此成为鲁棒学习的基准；但图像上的梯度扰动无法直接迁移到机生文本检测的离散token上

Hu等人提出**RADAR**，用改写器生成语义保持的改写文本作为扰动，与检测器**交替对抗训练**，使检测器在改写攻击下保持判别力；把**扰动训练引入**机生文本检测

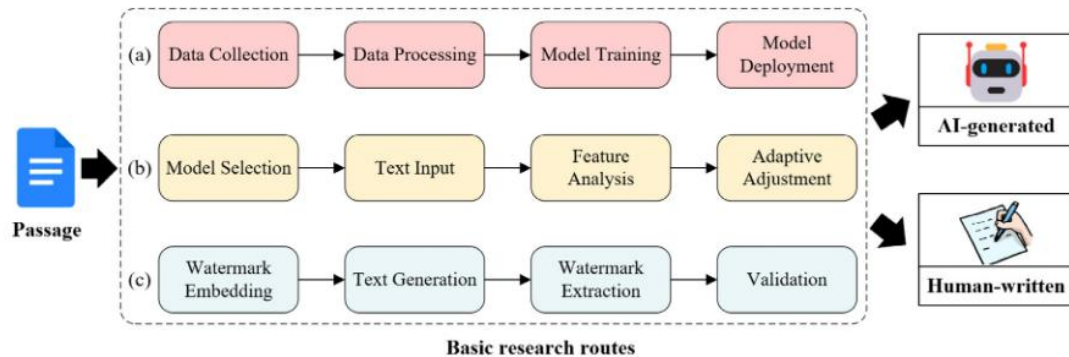
Zhou等人提出**DP-Net**，把噪声的均值与方差作为连续动作，由DDPG依据训练损失的回报调整**扰动强度**，在不访问目标域数据的条件下同时提升**泛化与鲁棒**；但扰动较强时语义仍会**漂移**

Li等人提出**RAMP-Net**，将噪声与原表示按**插值系数**混合，并把系数约束在**语义流形**附近，配合双流残差融合与语义锚定头联合训练，在提升强扰动鲁棒性的同时抑制语义漂移；但流形假设与判定阈值仍按源域设定，跨域时需重新校准



- 基于监督训练与特征学习

- 核心思想：用已标注的人类文本与机生文本训练**分类器**，学习人机判别边界
- 常见思路
 - 用BERT/RoBERTa等编码器提取语义与风格特征
 - 用代理大模型提供的token概率等作为特征
 - 训练中引入扰动或对抗样本，提升跨域与对抗鲁棒性
- 代表方法：Ghostbuster、SCRN、DP-Net、RAMP-Net
- 优势：域内准确率高，可扩展到**多分类与细粒度任务**
- 局限：学到的是生成器与领域特征，文本来源**换域、换模型**时性能下降



- 基于统计概率分布特征

- 核心思想：不训练检测器，用预训练模型计算token概率获取统计特征
- 常见特征：困惑度、Token Rank、概率曲率、双模型困惑度比
- 代表方法：DetectGPT、Fast-DetectGPT、Binoculars
- 优势：可零样本检测、部署成本低
- 局限：只抓表层统计，改写、回译容易失效，训练困难

- 基于信息嵌入/水印

- 核心思想：生成时按规则偏置采样，检测端做统计检验
- 常见做法：免训练水印、需训练水印、多比特水印
- 代表方法：Kirchenbauer等
- 优势：误报率可界定，检测只需统计量
- 局限：依赖模型配合，改写、翻译、水印窃取会削弱信号

- 字符级扰动（改动小，易被拼写检查发现）

- 做法：错字、同形字、增删字符
- 代表：DeepWordBug、TextBugger

- 词级扰动（保持语义，最常用）

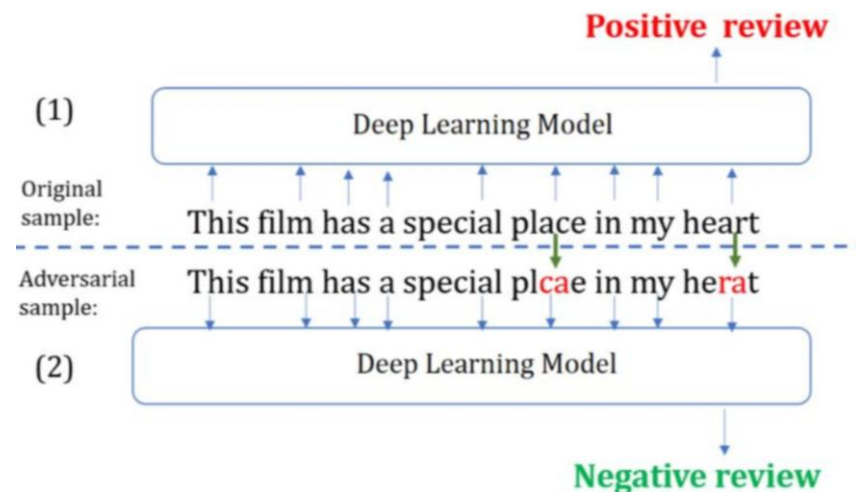
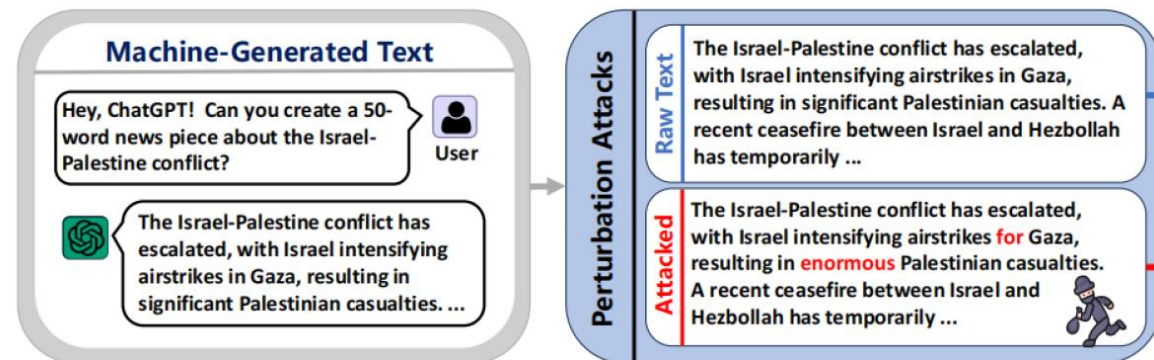
- 做法：同义替换、**近义词替换**
- 代表：Alzantot等

- 句级扰动（语义保持好，依赖改写模型）

- 做法：**改写**、**回译**、语序调整
- 代表：SCPN、RADAR改写器

- 表示级扰动（本报告两篇使用方式）

- 做法：embedding加噪或插值
- 代表：FreeLB、DP-Net、RAMP-Net



• 强化学习框架

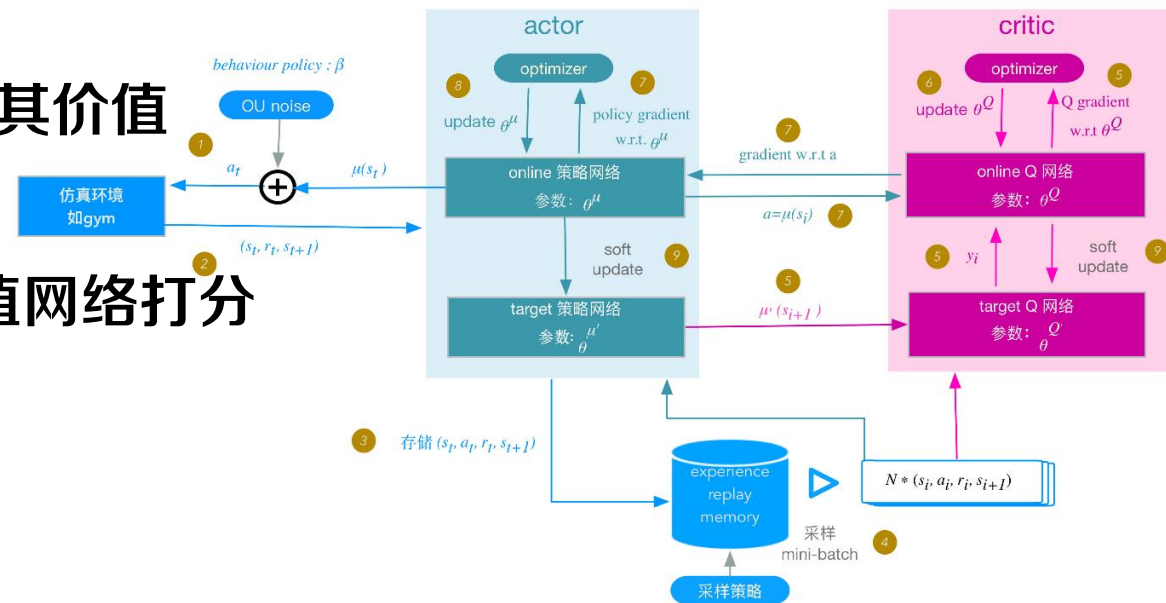
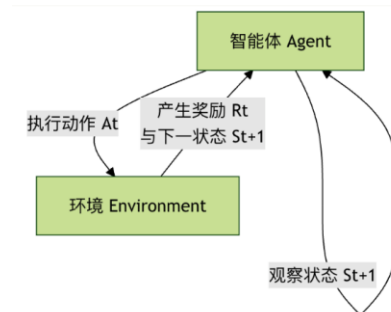
- 智能体与环境交互，按奖励调整策略，最大化**累计回报**
- 五要素：状态、动作、奖励、策略、价值

• 从Q-learning到DDPG（深度确定性策略梯度算法）

- Q-learning: **表格**存状态-动作价值，高维度失效
- DQN: 用网络拟合Q值，但动作仍须**离散**
- DDPG: **演员**输出连续动作，**评论家**估计其价值

• DDPG的三个关键设计

- 演员-评论家结构：策略网络选动作，价值网络打分
- 经验回放：打散样本、降低相关性
- 目标网络 + 软更新：防止价值估计震荡
- 适用条件：动作连续、奖励可标量化





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



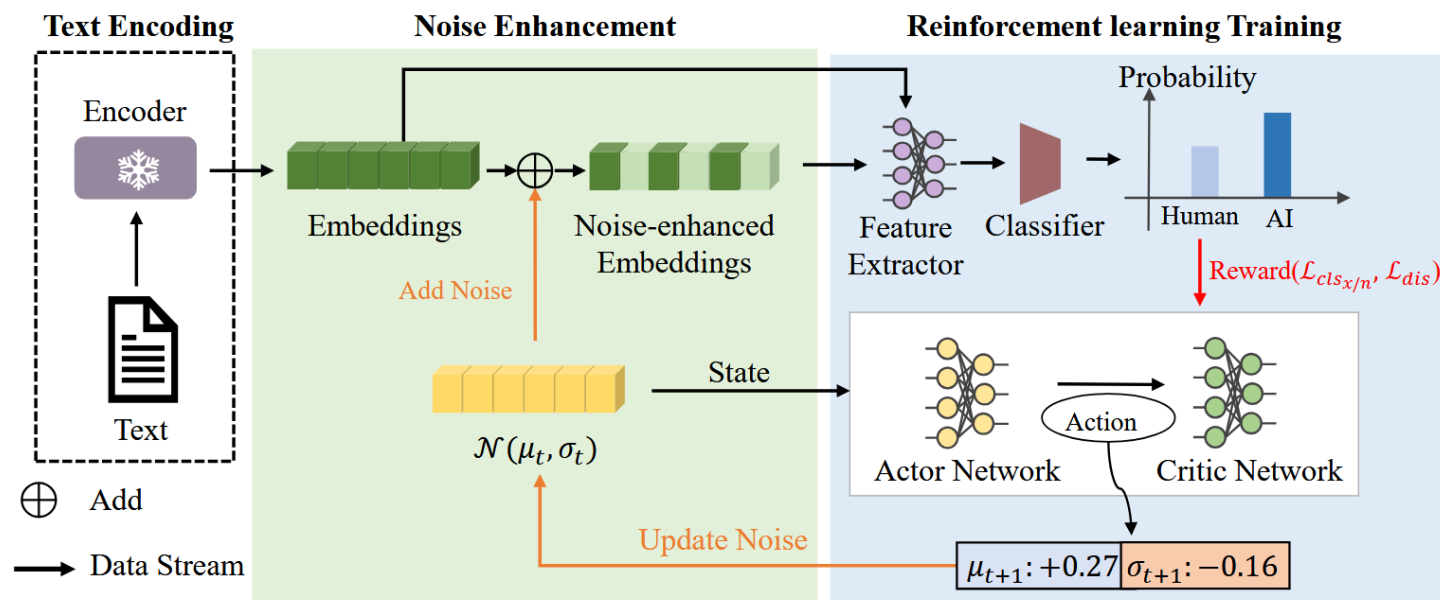
T	目标	提升检测器识别机生文本的泛化性和鲁棒性，区分人类文本与机生文本
I	输入	待检测文本 数据集：M4（arXiv 12000、Reddit 12000、WikiHow 9000、PeerRead 10486、Wikipedia 8995）
P	处理	1.用冻结RoBERTa编码文本得到文本表示，叠加高斯噪声得到加噪表示 2.输入同一特征提取器分别提取原始、加噪两路特征 3.计算两路交叉熵损失与距离损失，加权后训练检测器 4.采用DDPG依据训练反馈调整噪声的均值与方差
O	输出	待测文本是否属于机器生成文本

P	问题	现有检测方法或只关注泛化、或只关注鲁棒，缺少统一机制；单一噪声类型难以适应不同目标域
C	条件	具备源域标注数据与冻结的预训练语言模型；训练阶段可反复采样噪声
D	难点	噪声强度需人工设定、无目标域数据时难以泛化；扰动过强破坏语义、过弱导致过拟合
L	水平	NAACL 2025 CCF B

- 科学问题
 - 如何在无目标域标注的条件下，让检测器学到与生成器、领域无关的判别特征，
兼顾跨域**泛化性与鲁棒性**
- 应用问题
 - 现有检测器依赖特定语料与生成模型，难以适应领域与模型迭代，换域即性能下降，导致部署失效
 - 目标域标注成本高，难以针对每个新领域重新训练

• 创新框架

- 文本编码与加噪：冻结RoBERTa得到文本表示，采样高斯噪声并叠加
- 双分支判别：原始与加噪两路各自分类，再用距离约束把两者拉近
- 强化学习调参：损失换算得到奖励，DDPG回传更新噪声的均值与方差



• 编码

- 文本切成 n 个token, 冻结RoBERTa取表示 E_x
- 编码器不更新, 参数更新只发生在训练过程中的特征提取器与分类头

• 噪声增强

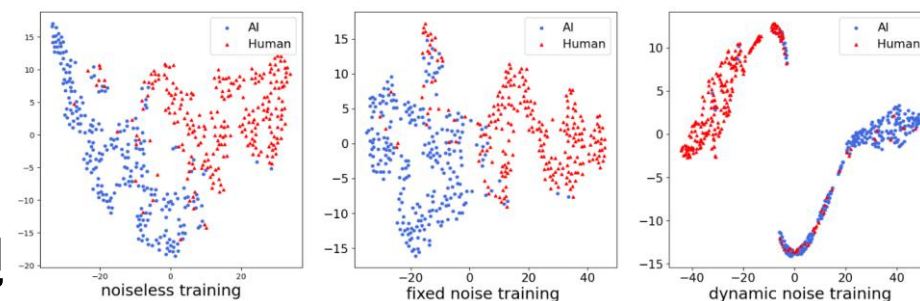
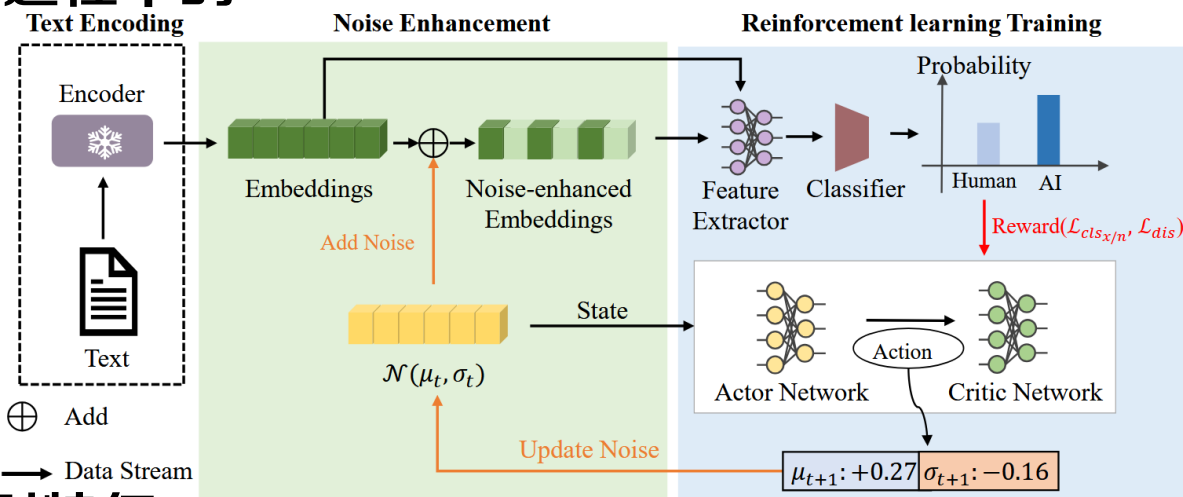
- 从 $N(\mu, \sigma^2)$ 采样噪声叠加:

$$E_n = E_x + N(\mu, \sigma^2)$$

- 经编码器得特征 z_x 、 z_n , 两路共享参数 $\oplus$ Add
- 距离损失 $\|z_x - z_n\|^2$ 迫使扰动不改变判别特征

• 训练目标

- 两路各算交叉熵, 按 λ_1 、 λ_2 、 λ_3 加权成总损失
- 每步用两路样本同时更新, 扰动只在训练期出现



- 损失与奖励

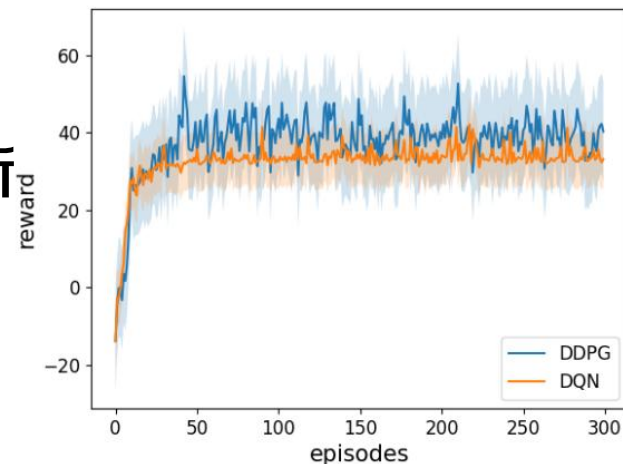
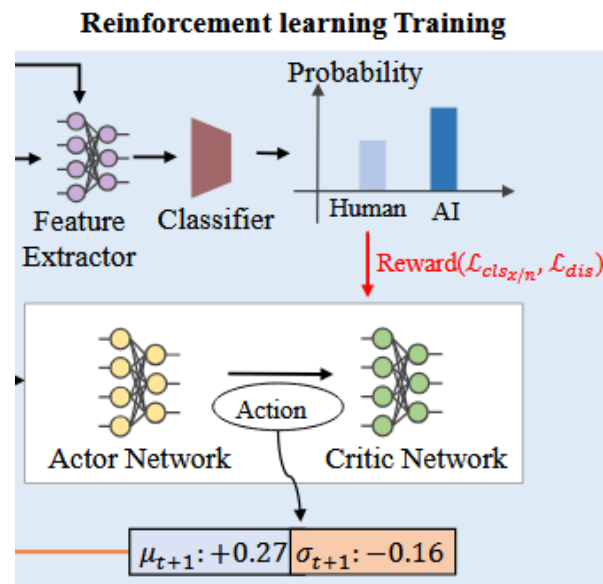
- 总损失按阈值 ε 转成奖励:

$$r(L) = \begin{cases} -\log(L - \varepsilon), & L - \varepsilon > 0 \\ e^{\varepsilon - L}, & L - \varepsilon \leq 0 \end{cases}$$

- 阈值的作用是策略先把损失压下来, 再约束扰动多样性

- DDPG的训练

- 状态 $s = (\mu, \sigma)$, 动作 a 是噪声参数的增量, 属连续动作空间
 - 交互得到 (s, a, r, s') , 存入经验回放, 随机采样更新
 - 评论家回归目标Q值 $y = r + \gamma Q'(s', \mu'(s'))$, 演员沿梯度更新
 - 目标网络按 $\theta' = \tau\theta + (1 - \tau)\theta'$ 软更新, 保证训练稳定
 - 推理: 关闭扰动与DDPG, 只走干净分支, 速度接近单流





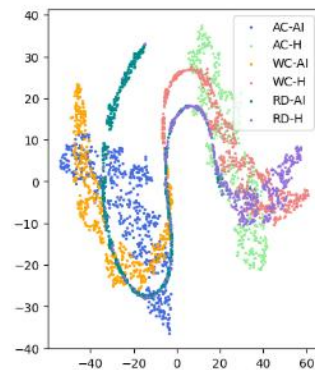
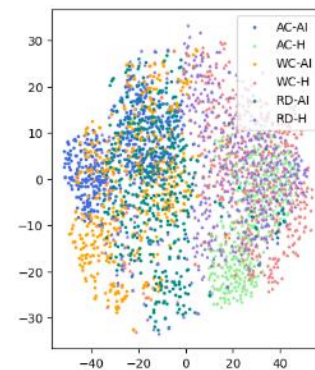
- 数据集与目标域
 - 源域Arxiv-ChatGPT
 - 目标7域： WikiHow-C、 Wikipedia-C、 Reddit-C、 Arxiv-D、 Arxiv-F、 PeerRead-D、 Reddit-D
- 三类跨域场景
 - 同生成器跨语料： Arxiv-C $\rightarrow$ WikiHow-C / Wikipedia-C / Reddit-C
 - 同语料跨生成器： Arxiv-C $\rightarrow$ Arxiv-D、 Arxiv-F
 - 语料与生成器同变： Arxiv-C $\rightarrow$ PeerRead-D、 Reddit-D
- 鲁棒与对比
 - 鲁棒： 同义替换攻击下的跨域准确率； RAID另测LLaMA-70B
 - 对比： Naive Classifier、 RoBERTa、 SCRn、 GLTR、 Fast-DetectGPT
 - 指标： Acc、 F1， AUROC

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

- 七个未见目标域上**平均准确率**最高，在Wikipedia-C、Arxiv-D、Reddit-D等域上取得领先水平

Method	Naive Classifier		Roberta		SCRN		GLTR		Fast detectGPT		DP-Net			
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	+U(ours)		+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

训练期扰动可在一定程度上替代目标域数据，但无法消除领域差异



- 同义替换下两项指标均最高；释义改写下准确率最高

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

扰动训练大幅度提升了检测方法抗改写能力，鲁棒性增强

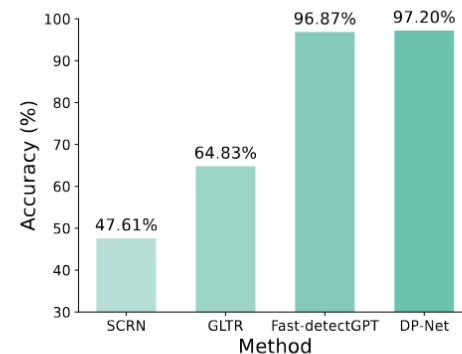
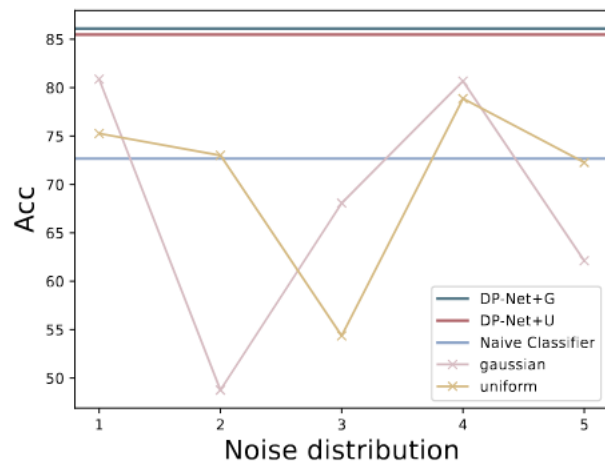


Figure 5: Comparison of Models on text generated by LLaMa-70B-chat.

- 去掉DDPG后平均AUROC与F1的跌幅最大，冻结编码器次之
- 噪声类型同样影响结果：高斯噪声准确率更高，均匀噪声F1略高

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

扰动强度的可学习性决定最终收益，固定加噪难以覆盖不同场景



• 算法贡献

- 统一视角：把鲁棒性定义为**域偏移**，训练期扰动机制兼顾泛化与鲁棒
- 动态扰动：DDPG把噪声参数当连续动作，按损失回报调整，无需为目标域手工设参
- 双分支对齐：原始与加噪两路共享判别边界，用**距离损失**约束特征一致

• 算法不足

- 噪声自由：表示空间无约束加噪，强扰动下**语义会漂移**
- 单流融合：两路特征混在一起学，未显式区分干净与扰动
- 依赖源域：仍需源域标注与冻结PLM，跨域效果受源域质量制约

XYUL-491



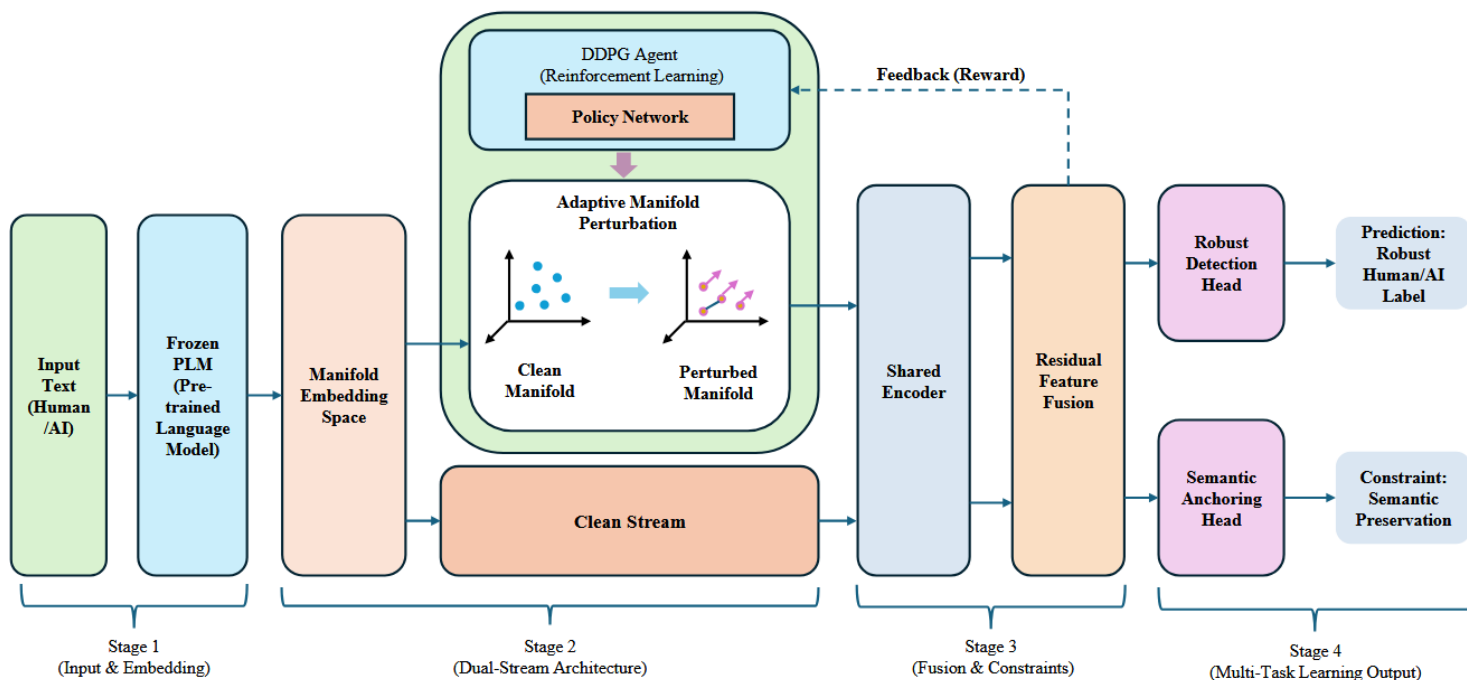
Reinforcement-Guided Adaptive Manifold Perturbation Framework for Robust AI-Generated Text Detection

T	目标	提升检测器在域偏移与强扰动下的判别稳定，兼顾语义保持与跨域泛化
I	输入	待检测文本 数据集：M4（Wikipedia、Reddit、WikiHow、PeerRead、arXiv × ChatGPT、Davinci、Flan-T5、Cohere）
P	处理	1. 用RoBERTa编码文本得到上下文表示，并按插值系数生成扰动表示 2. 干净流与扰动流共享编码器，拼接后经残差融合 3. 计算主分类损失、辅助语义锚定损失与一致性损失，加权后训练 4. 采用DDPG依据总损失回报调整噪声的均值与方差
O	输出	待测文本是否属于机器生成文本
P	问题	现有方法把静态噪声增强，忽略语义流形；单流融合与单一分类目标在强扰动下易致语义漂移
C	条件	具备源域标注数据与冻结的预训练语言模型；训练阶段可反复采样噪声
D	难点	扰动幅度与语义保持难以两全、两路共享编码器时相互牵制；无目标域数据下仍需泛化
L	水平	IEEE TDSC 2026 CCF A

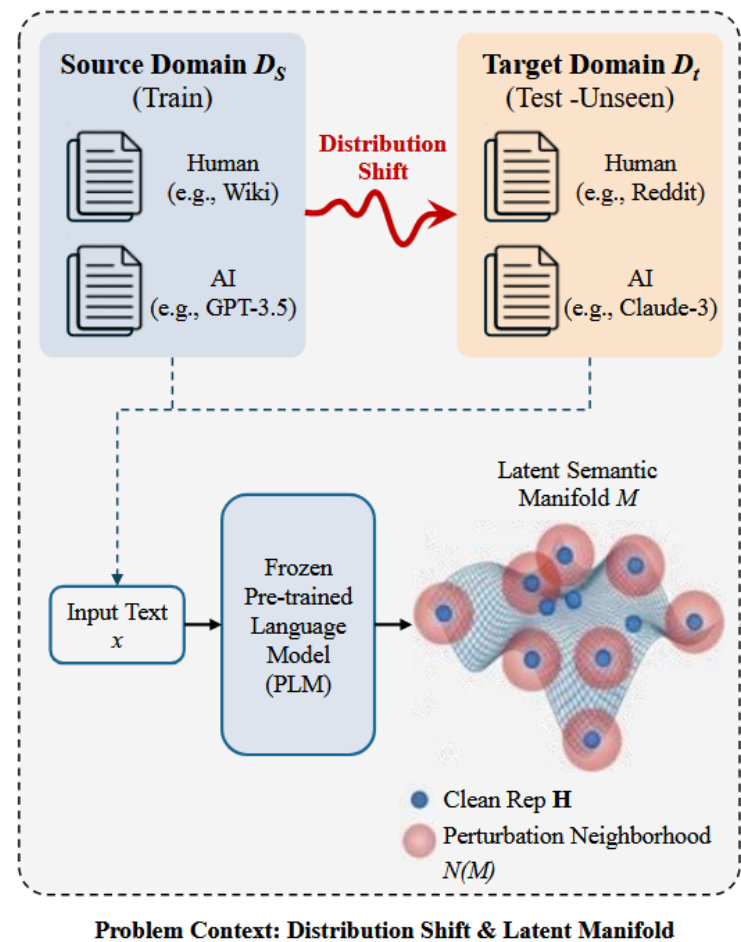
- **科学问题**
 - 如何在自适应调整扰动强度的同时，把扰动约束在语义流形附近，使检测器在强扰动下仍保持语义一致与判别力
- **应用问题**
 - 多数方法把扰动当作静态噪声增强，采用单流融合与单一分类目标，强扰动下语义漂移，跨域部署时性能下降

• 创新框架

- 输入与嵌入：文本经冻结PLM得到嵌入表示
- 双流与扰动：干净流与扰动流并行，DDPG按奖励调扰动，扰动受流形约束
- 融合与约束：共享编码器提取特征，残差层融合两路特征
- 多任务输出：检测头判断人机，语义锚定头保证语义



- 分布偏移
 - 训练域（Wiki + GPT-3.5）与测试域（Reddit + Claude-3）在语料与生成器两方面同时发生变化，存在严重的分布偏移
- 语义流形
 - 文本经冻结PLM映射到同一语义流形 M ，语义相近的样本在流形上彼此邻近
 - 每个干净表示 H 的周围存在扰动邻域 $N(M)$ ，语义保持的改写应落在该邻域内
- 鲁棒性定义
 - 当 $H_{aug} \in N(M)$ 时要求 $g(H_{aug}) \approx g(H)$ ，扰动也因此需约束在流形附近



扰动生成

- 冻结PLM得到上下文表示 H ，噪声状态取 $S = (\mu, \sigma)$
- 演员输出动作 a ，经状态更新与截断得到插值系数 $\alpha \in (0.01, 0.5)$
- 采样噪声 N 后按 α 插值，得到 $H_{aug} = (1 - \alpha)H + \alpha N$

约束方式

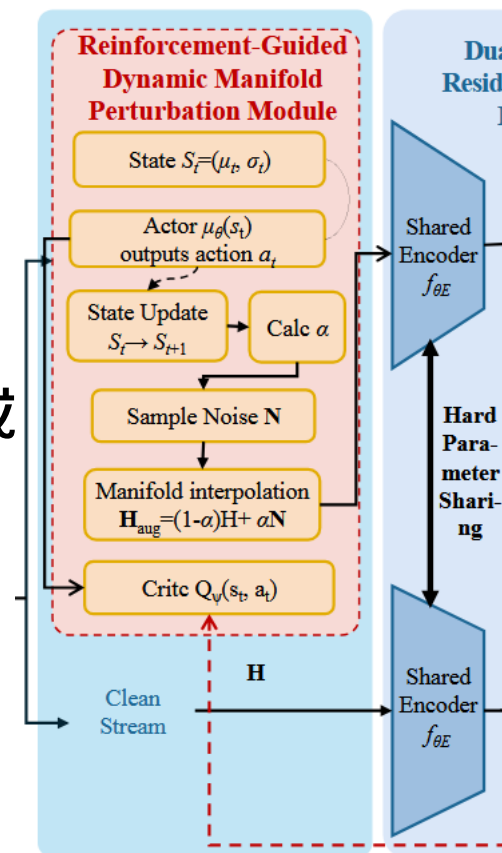
- 插值系数将扰动幅度限制在语义流形邻域内，避免直接相加造成的语义漂移

奖励设计

- 三类损失加权为总损失，经阈值 τ 换算为奖励：

$$r_t = \begin{cases} -\log(L_t - \tau + 1), & L_t > \tau \\ e^{\tau - L_t}, & L_t \leq \tau \end{cases}$$

- 评论家估计Q值，演员沿奖励提升方向更新，进入下一轮迭代



- 共享编码

- 干净表示 H 与扰动表示 H_{aug} 送入同一**共享编码器**，分别得到句子级特征 Z_{clean} 与 Z_{noise}

- 两路共享编码器参数，训练过程中**同步更新**

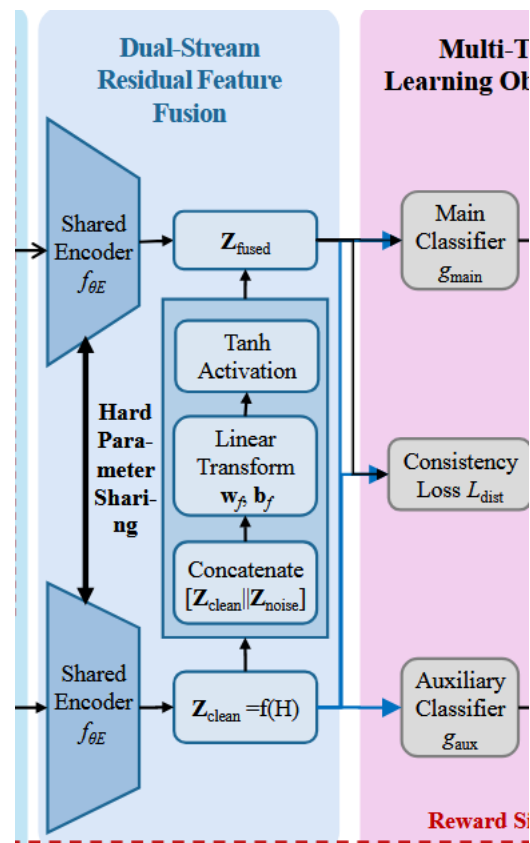
- 残差融合

- 两路特征拼接后经线性变换与tanh激活得到融合特征 Z_{fused}

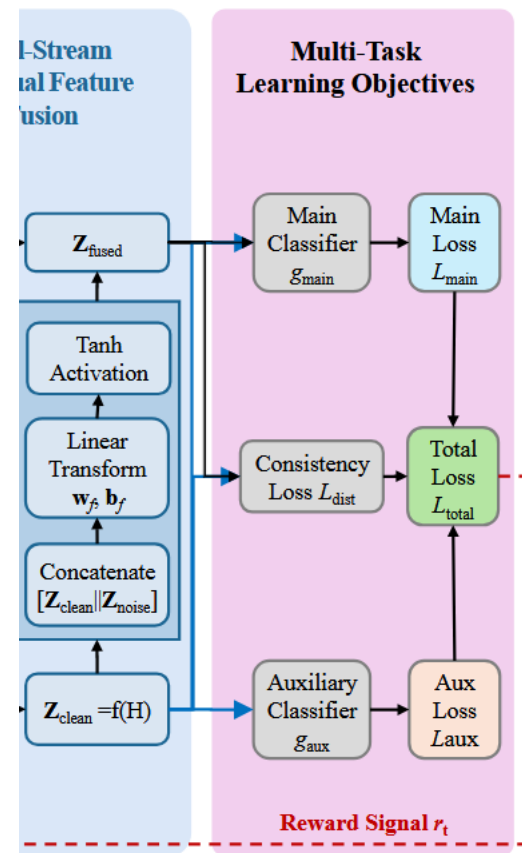
- Z_{clean} 同时作为**语义锚定头**的输入

- 与单流结构的区别

- 单流结构将两类特征**混合编码**，模型只能被动适应噪声
- 双流结构显式建模干净与扰动特征的关系，在强扰动下更稳定



- 任务一：人机检测
 - 以 Z_{fused} 为输入判断文本来源，损失为 L_{main}
- 任务二：语义保持
 - 以 Z_{clean} 为输入预测同一标签，损失为 L_{aux}
 - 目的：强扰动训练下不遗忘干净语义，避免语义漂移
- 一致性约束
 - $L_{dis} = \|Z_{clean} - Z_{noise}\|^2$ ，约束两路特征的差异
- 联合优化
 - $L_{total} = \lambda_1 L_{main} + \lambda_2 L_{aux} + \lambda_3 L_{dis}$ ($\lambda_1=\lambda_2=1.0$ 、 $\lambda_3=0.1$)



• 数据集

- M4英文子集：五语料（Wikipedia、Reddit、WikiHow、PeerRead、arXiv）与五生成器（ChatGPT、Davinci、Flan-T5、Cohere、Dolly）

• 三类目标域

- 跨语料：PeerRead、Reddit、WikiHow、Wikipedia-ChatGPT
- 跨模型：Arxiv-Cohere、Arxiv-Davinci、Arxiv-Flan-T5、Arxiv-Dolly
- 跨语料与跨模型：PeerRead-Cohere、PeerRead-Davinci等

• 实验设置

- 源域Arxiv-ChatGPT单源训练

• 对比与指标

- 对比方法：BERT、SCRN、GLTR、PPL、DP-Net
- 指标：Acc、F1



- 单域数据集上各项指标均高于基线；跨语料时Wikipedia保持较高水平，WikiHow与PeerRead下降明显

Dataset	BERT		SCRN		GLTR		PPL		DP-Net		Ours	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Arxiv-ChatGPT	0.9456	0.9451	0.9852	0.9855	0.9566	0.9562	0.9461	0.9461	0.9977	0.9981	0.9998	0.9998
PeerRead-ChatGPT	0.9495	0.9551	0.9745	0.9772	0.9480	0.9496	0.9417	0.9417	0.9939	0.9939	0.9967	0.9967
Reddit-ChatGPT	0.9461	0.9462	0.9881	0.9880	0.9862	0.9866	0.9406	0.9406	0.9967	0.9967	0.9983	0.9983
WikiHow-ChatGPT	0.9333	0.9333	0.9779	0.9779	0.9796	0.9796	0.9346	0.9346	0.9977	0.9977	0.9979	0.9979
Wikipedia-ChatGPT	0.9512	0.9553	0.9801	0.9809	0.9688	0.9720	0.9561	0.9555	0.9757	0.9800	0.9986	0.9987

Source Domain (Train)	Target Domain (Test)	Acc	F1
Arxiv-ChatGPT	PeerRead-ChatGPT	0.8242	0.7944
	Reddit-ChatGPT	0.8672	0.8787
	WikiHow-ChatGPT	0.7383	0.7102
	Wikipedia-ChatGPT	0.9613	0.9625

单域检测已接近饱和，跨语料迁移构成主要性能瓶颈

- 换生成器后整体稳定；语料与生成器同时变化时PeerRead系列退化最明显

Source Domain (Train)	Target Domain (Test)	Acc	F1
Arxiv-ChatGPT	PeerRead-Cohere	0.8140	0.7798
	PeerRead-Davinci	0.8677	0.8528
	PeerRead-Dolly	0.7216	0.7215
	Reddit-Cohere	0.7557	0.7290
	Reddit-Davinci	0.8549	0.8656
	Reddit-Dolly	0.8560	0.8672
	WikiHow-Cohere	0.8391	0.8398
	WikiHow-Davinci	0.7887	0.7783
	WikiHow-Dolly	0.8060	0.8001
	Wikipedia-Cohere	0.9315	0.9321
	Wikipedia-Davinci	0.9503	0.9514
	Wikipedia-Dolly	0.8616	0.8509

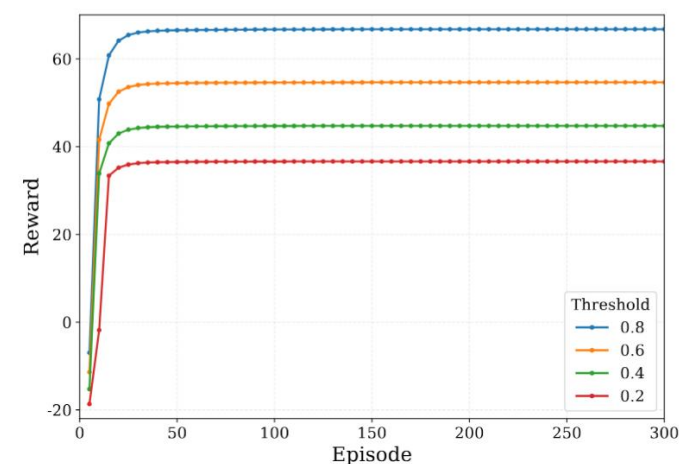
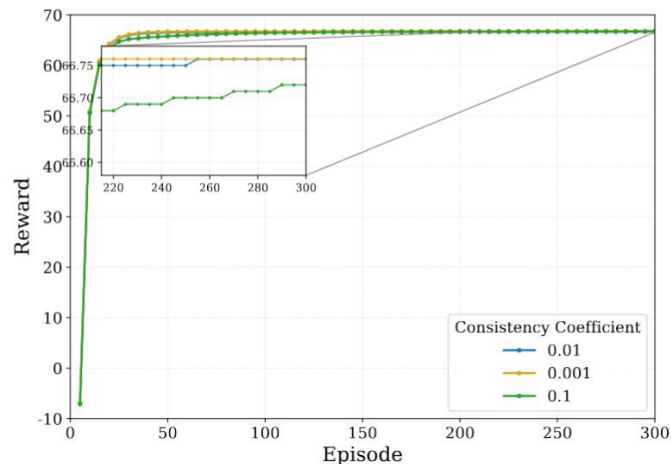
Source Domain (Train)	Target Domain (Test)	Acc	F1
Arxiv-ChatGPT	Arxiv-Cohere	0.9825	0.9822
	Arxiv-Davinci	0.9345	0.9299
	Arxiv-FlanT5	0.9237	0.9174
	Arxiv-Dolly	0.9755	0.9749

生成器更替的影响有限，语料层面的分布差异是鲁棒性下降的主要来源

- 在简单扰动策略中平均最优；源域换WikiHow变体后仍可用

Method	Avg. Acc	Avg. F1
Fixed Gaussian Noise	0.7984	0.7906
Scheduled Noise Annealing	0.8117	0.8035
Learnable Static Noise	0.8219	0.8142
Meta Perturbation Control	0.8296	0.8221
RAMP-Net	0.8373	0.8307

Target Domain	WikiHow-ChatGPT		WikiHow-Cohere		WikiHow-Davinci		WikiHow-Dolly	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
PeerRead-Cohere	0.7125	0.7088	0.6612	0.6655	0.6842	0.6795	0.6827	0.6883
PeerRead-ChatGPT	0.7457	0.7512	0.6485	0.6420	0.7025	0.6988	0.6854	0.6918
PeerRead-Davinci	0.6982	0.6924	0.6315	0.6495	0.7354	0.7312	0.7286	0.7242
PeerRead-Dolly	0.6545	0.6497	0.6420	0.6483	0.6486	0.6423	0.7987	0.7945
Arxiv-Cohere	0.7218	0.7183	0.7785	0.7815	0.7155	0.7112	0.6843	0.6796
Arxiv-ChatGPT	0.7383	0.7462	0.7353	0.7290	0.7428	0.7385	0.7024	0.6885
Arxiv-Davinci	0.6805	0.5899	0.6929	0.6886	0.6683	0.6624	0.7396	0.7352
Arxiv-FlanT5	0.6282	0.6775	0.7018	0.6988	0.6952	0.6915	0.7257	0.7213
Arxiv-Dolly	0.6495	0.6461	0.7241	0.7194	0.7314	0.7258	0.7428	0.7384
Reddit-Cohere	0.6492	0.6789	0.6726	0.6757	0.6655	0.6612	0.6153	0.6114
Reddit-ChatGPT	0.6715	0.6121	0.6453	0.6415	0.6824	0.6785	0.6385	0.6246
Reddit-Davinci	0.6374	0.6368	0.6260	0.6318	0.7192	0.7156	0.6677	0.6683
Reddit-Dolly	0.6241	0.6235	0.6385	0.6325	0.6358	0.6294	0.6654	0.6815
Wikipedia-Cohere	0.7523	0.7319	0.7988	0.8019	0.7425	0.7386	0.7723	0.7685
Wikipedia-ChatGPT	0.7656	0.7861	0.7520	0.7486	0.7588	0.7542	0.7856	0.7817
Wikipedia-Davinci	0.7010	0.7095	0.7248	0.7192	0.7854	0.7815	0.8284	0.8342
Wikipedia-Dolly	0.7165	0.7158	0.7213	0.7156	0.7226	0.7183	0.8325	0.8286



扰动强度的可学习性比噪声形式更关键，参数在合理区间内稳定



- 开销与DP-Net相当、高于BERT；不同随机种子下波动很小；增大训练预算后性能继续提升

Source Domain	Target Domain	Seeds	Acc	F1	Conv. Episode	Time (h)
Arxiv-ChatGPT	PeerRead-ChatGPT	5	0.8241 ± 0.0063	0.7947 ± 0.0081	258 ± 17	4.43 ± 0.06
Arxiv-ChatGPT	Reddit-ChatGPT	5	0.8668 ± 0.0057	0.8782 ± 0.0064	246 ± 15	4.42 ± 0.05
Arxiv-ChatGPT	WikiHow-ChatGPT	5	0.7386 ± 0.0079	0.7105 ± 0.0092	286 ± 21	4.45 ± 0.07
Arxiv-ChatGPT	Wikipedia-ChatGPT	5	0.9610 ± 0.0041	0.9622 ± 0.0045	229 ± 13	4.41 ± 0.05
Arxiv-ChatGPT	Arxiv-Cohere	5	0.9821 ± 0.0032	0.9819 ± 0.0035	218 ± 11	4.40 ± 0.04
Arxiv-ChatGPT	Arxiv-Davinci	5	0.9342 ± 0.0054	0.9296 ± 0.0062	251 ± 16	4.43 ± 0.06
Arxiv-ChatGPT	Arxiv-FlanT5	5	0.9240 ± 0.0061	0.9177 ± 0.0068	263 ± 18	4.44 ± 0.06
Arxiv-ChatGPT	Arxiv-Dolly	5	0.9752 ± 0.0038	0.9746 ± 0.0041	224 ± 12	4.41 ± 0.05

Model	Time (s/ep)	Mem. (GB)	Thpt. (samples/s)
BERT	30	2.1	182
SCRN	33	2.5	170
GLTR	32	2.5	167
PPL	34	2.5	167
DP-Net	42	3.0	140
RAMP-Net	40	3.0	140

Budget Factor	Setting	Acc	F1
Replay buffer size	1,000	0.9956	0.9954
Replay buffer size	5,000	0.9971	0.9970
Replay buffer size	10,000	0.9980	0.9979
Replay buffer size	50,000	0.9992	0.9991
Replay buffer size	100,000	0.9998	0.9998
Fine-tuning budget	10% / 30 of 300 epochs	0.9928	0.9925
Fine-tuning budget	25% / 75 of 300 epochs	0.9957	0.9955
Fine-tuning budget	50% / 150 of 300 epochs	0.9978	0.9977
Fine-tuning budget	75% / 225 of 300 epochs	0.9991	0.9990
Fine-tuning budget	100% / 300 of 300 epochs	0.9998	0.9998

检测性能的提升未伴随显著开销，且在不同随机种子与训练预算下保持稳定

• 算法贡献

- 扰动设计：把鲁棒性视为流形上的**局部不变性**，扰动幅度受邻域约束
- 融合结构：干净流与扰动流共享编码器，残差融合建模两类特征
- 训练目标：检测头与语义锚定头共享参数，检测与**语义保持**联合优化

• 算法不足

- 流形近似：邻域范围由插值系数经验近似，缺乏几何验证
- 语义约束：锚定头仅作用于干净流，扰动流语义依赖噪声假设
- 迁移成本：阈值与插值区间依赖源域，换域需**重新校准**



特点总结与未来展望

- 特点总结

- 扰动从检测时的试探，变为**训练时的增强**
 - 不生成新文本，只在表示层加噪或插值，让检测器在训练阶段见到数据分布的变化
 - DP-Net 让噪声强度自适应，RAMP-Net 把扰动限制在语义流形附近
- 噪声强度从人工设定，改为强化学习**动态调整**
- 语义保持从训练后补救，改为训练中一起优化

- 未来展望

- 从单一表示层扰动，扩展到句级改写、跨语言与多层级
- 从依赖源域假设，转向**少量目标域数据校准**
- 从双流结构的高开销，走向轻量化、可部署
- 从经验调参，走向有**理论依据**的扰动强度设计

- [1] Zhou Y, Hou W J, Peng W, et al. Kill two birds with one stone: generalized and robust AI-generated text detection via dynamic perturbations. Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) [C]. Albuquerque, New Mexico: Association for Computational Linguistics, 2025: 8864-8875.**
- [2] Li C, Luo T, Qiu K, et al. Reinforcement-guided adaptive manifold perturbation framework for robust AI-generated text detection [J]. IEEE Transactions on Dependable and Secure Computing, 2026: 1-16. DOI: 10.1109/TDSC.2026.3716942.**

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！

