DetectAnyLLM: Towards Generalizable and Robust Detection of Machine-Generated Text Across Domains and Models

Jiachen Fu
VCIP, CS, Nankai University
Tianjin, China
fujiachen2005@gmail.com

Chun-Le Guo*
VCIP, CS, Nankai University
Tianjin, China
NKIARI
Shenzhen Futian, China
guochunle@nankai.edu.cn

Chongyi Li†
VCIP, CS, Nankai University
Tianjin, China
NKIARI
Shenzhen Futian, China
lichongyi@nankai.edu.cn

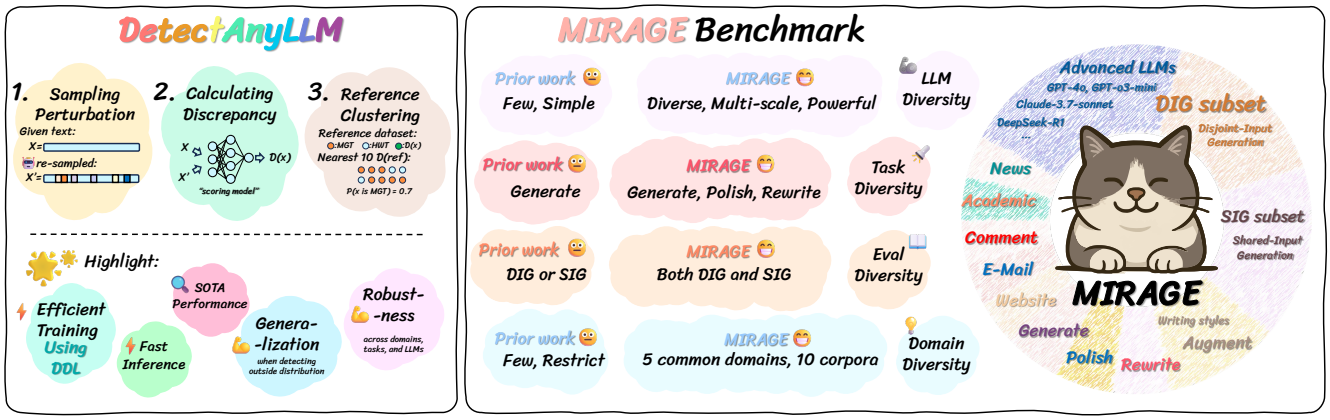

Figure 1: 左: 我们的 DetectAnyLLM 通过一个三步流程——采样扰动、计算差异与参考聚类，实现了高效率、强鲁棒性及出色的泛化能力。右：我们的 MIRAGE 基准测试强调了跨文本域、任务、评估场景和源大语言模型的多样性，实现了全面而稳健的评估。

Abstract

大型语言模型（LLM）的飞速发展使得机器生成文本检测（MGTD）任务受到了广泛的关注。然而，现有的方法在复杂的现实世界场景中举步维艰：零样本检测器严重依赖于评分模型的原始输出分布，而基于训练的检测器则常常因对训练数据的过拟合而受限，从而限制了其泛化能力。我们发现，基于训练的检测器的性能瓶颈源于训练目标与任务需求之间的错位。为了解决这个问题，我们提出了**直接差异学习（DDL）**，这是一种新颖的优化策略，它利用面向任务的知识直接对检测器进行优化。DDL 使检测器能够更好地捕捉检测任务的核心语义，从而增强其鲁棒性和泛化能力。在此基础上，我们引入了 **DetectAnyLLM**，一个统一的检测框架，它在各种不同的大型语言模型上均实现了当前最先进的 MGTD 性能。为确保评估的可靠性，我们构建了 **MIRAGE**，这是目前最多样化的多任务 MGTD 基准测试。MIRAGE 从 5 个文本领域的 10 个语料库中抽样人类编写的文本，然后使用 17 个前沿的 LLM 对这些文本进行重新生成或修订，涵盖了广泛的专有模型和文本风格。在 MIRAGE 上进行的大量实验揭示了现有方法在复杂环境中的局限性。相比之下，DetectAnyLLM 的性能始终优于这些方法，在相同的训练数据和基础评分模型下，性能提升超过 70%，这突显了 DDL 方法的有效性。项目主页：https://fjc2005.github.io/detectanyllm。

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; **Natural language processing**; • **Security and privacy**;

Keywords

Machine-Generated Text Detection, AI-Text Detection, AI Safety, Natural Language Processing, Deep Learning

ACM Reference Format:
Jiachen Fu, Chun-Le Guo, and Chongyi Li. 2025. DetectAnyLLM: Towards Generalizable and Robust Detection of Machine-Generated Text Across Domains and Models. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3746027.3754862

1 引言

先进的大型语言模型 *(LLMs)* [2, 18, 24, 26, 31, 48] 可以轻易地生成与人类写作几乎无法区分的文本 [10, 45]。如果被滥用，它可能会对社会构成严重风险 [52]。为应对这一担忧，机器生成文

*通讯作者.
†项目领导.

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754862

本检测 *(MGTD)* 任务应运而生 [1, 14, 25, 27, 29, 44]。MGTD 是一项二元分类任务，旨在区分给定文本是由人类编写的，还是由机器生成（或修订）的。

在本研究中，我们同时考虑了对机器生成文本 *(MGT)* 和机器修订文本 *(MRT)* 的检测，其中 MRT 指的是模型在人类编写文本 *(HWT)* 基础上进行润色或改写的文本。我们的研究专注于黑盒检测，这比白盒设置更能反映真实世界的应用场景。

现有的一些 MGTD 方法 [7, 14, 19, 25, 34, 44, 49, 59] 基于 MGT 和 HWT 的词元（token）概率分布存在差异这一假设。其中大多数方法利用预训练语言模型（被称为评分模型），来估计给定文本的词元概率，并计算用于区分的分类指标。

现有的 MGTD 方法可分为零样本方法 [5, 7, 14, 34, 46] 和基于训练的方法 [9, 25]。零样本方法通常依赖于评分模型的固有能力 [6]。然而，这些模型通常规模较小，知识有限且输出模式简单。因此，当检测的文本偏离其固有分布时，这类方法往往难以达到可靠的性能。基于训练的方法使用监督微调 *(SFT)* [25, 36] 或偏好学习 [9, 22, 40]，来使评分模型的输出分布与构建训练数据的模型的分布对齐。尽管这种方法提升了对特定模型的检测性能，但似乎很难将这种检测知识泛化到训练数据之外的模型上 [7, 47, 54]。

我们指出，SFT 和偏好学习都是在引导评分模型模仿生成器，而不是直接为检测任务进行优化。换言之，以往基于训练的方法其训练目标是面向模型的，而非面向任务的。这导致评分模型只能学到训练数据所用生成器的知识，而无法直接学习检测任务本身的知识。最终，这损害了检测器的泛化性和鲁棒性。

我们提出了**直接差异学习 (DDL)**，这是一种新颖的优化策略，它通过直接使用输出的分类指标来优化评分模型，从而使模型能够**学会成为一个检测器，而不仅仅是另一个语言模型**。DDL 的思想源于将评分模型从其语言模型的身份中解放出来，并设计一个面向任务的损失函数，从而使评分模型能够直接学习 MGTD 的内在知识，而不仅仅是拟合训练数据的分布。

此外，我们将先前的方法 [7, 9] 与 DDL 相融合，提出了一个统一的 MGTD 框架——**DetectAnyLLM**。DetectAnyLLM 通过包含重采样、差异计算和参考聚类三个步骤来实现高效且鲁棒的检测。该框架提炼了现有方法的核心思想 [7, 34]，同时利用 DDL 来增强模型的泛化能力并提高检测的鲁棒性。

尽管已有多项 MGTD 研究问世，但仍然缺乏全面的基准 [54]。现有的基准 [12, 21, 30, 55] 存在几个重大缺陷：

(1) 对 MRT 的关注有限：以 MGTBench [21] 为典型的大多数基准数据集仅关注 MGT，而忽略了对 MRT 的检测。
(2) 源语言模型范围狭窄：大多数基准依赖于小规模的开源模型，而现实世界的应用通常涉及先进的商业闭源大语言模型，如 GPT-4o [24] 和 Claude [3]。
(3) 领域覆盖受限：像 HC3 [17] 这样的基准仅从一个或少数几个领域中采样文本，忽略了机器生成文本的领域敏感性。

这些缺陷凸显了评估与实际应用之间的巨大鸿沟。尽管最近的一些研究 [5, 54] 已经认识到这些问题，但他们的数据集仍然不够全面。

为了促进全面的评估，我们构建了 **MIRAGE**，这是 MGTD 研究中规模最大、提供最多样化的商业闭源大语言模型和最全面文本领域的多任务基准。如 Table 1所示，MIRAGE 从 **5** 个常见领域的 **10** 个语料库中采样文本，并使用 **17** 个先进的主流大语言模型进行文本生成或修订，创建了超过 **93K** 组 HWT-MGT

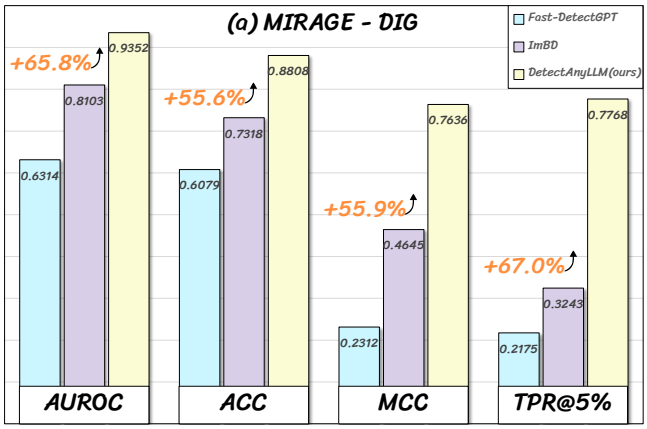

Figure 2: DetectAnyLLM 与包括 Fast-DetectGPT [7] 和 ImBD [9] 在内的当前最优方法在 MIRAGE-DIG 上的性能表现。Imp. (提升)： $(new - old)/(1.0 - old)$。

配对。MIRAGE 建立了一个更真实、更可靠的评估标准，弥合了研究与实际应用之间的差距。

尽管现有的检测方法在以往的基准上表现出看似优异的性能（AUC > 0.9）[9, 55]，但如 Figure 2所示，当在 MIRAGE 上进行评估时，它们暴露出了明显的弱点。这揭示了先前方法的泛化性和鲁棒性有待大幅提升。相比之下，DetectAnyLLM 仍然表现良好，在 DIG 子集上取得了平均 **0.9352** 的 AUROC 和 **0.7636** 的 MCC。这样的性能有力地证明了 DDL 的有效性和卓越的泛化能力。

我们的贡献可以总结为以下几点：

- 我们提出了一种新颖的、面向任务的优化方法——**直接差异学习 (DDL)**，该方法在不使用额外数据和更少资源的情况下，提升了模型的泛化性和鲁棒性。
- 我们构建了 **MIRAGE**，一个全面的 MGTD 基准。它覆盖了多样的领域和任务，并创新性地聚焦于使用商业闭源大语言模型，从而实现了更贴近现实的评估。
- 我们提出了 **DetectAnyLLM** 框架，它统一了先前的工作和 DDL，实现了高达 **70%** 的性能提升，并做到了跨领域和跨模型的可泛化、鲁棒的检测。

2 相关工作

2.1 零样本检测器

以往的 MGTD 研究关注于零样本检测，主要是出于对训练过程中过拟合问题的担忧 [4, 38]。早期的诸如 GLTR [14] 等方法利用文本熵来检测机器生成的内容，而其他一些方法则使用了基于似然或排序的策略 [25, 44]。最近，DetectGPT [34] 提供了一个新颖的视角，它利用扰动、评分和概率曲率估计来区分 MGT 和 HWT。Fast-DetectGPT [7] 改进了扰动步骤，在不牺牲性能的前提下显著加快了检测过程。尽管取得了一定进展，但零样本方法仍然受限于其对评分模型输出分布的依赖，正如 Glimpse [6] 所证明的那样，该研究通过使用更强大的评分模型展示了性能的提升。

2.2 基于训练的检测器

基于训练的检测器在特定的训练数据上对评分模型进行微调。一个早期的代表性工作是 RoBERTa-OpenAI-Detector [44]，研

Table 1: MIRAGE 与现有 MGTD 基准数据集的比较。“Size”指测试集的容量。“SIG”表示共享输入生成，“DIG”表示分离输入生成。“Commercial”指是否使用了前沿的商业闭源大语言模型（例如 GPT-4o）。在领域、任务和源语言模型方面，MIRAGE 是最多样化的基准。MIRAGE 利用强大的商业闭源大语言模型来生成和修订文本，这增加了检测的难度和评估的真实性，从而能够更忠实地评估检测器的鲁棒性。此外，MIRAGE 引入了一种新颖的双场景评估策略——DIG 和 SIG——从而可以更全面地评估检测器的准确性和泛化能力。

	数据统计			语言模型	MGT 任务			其他		
基准	规模	领域覆盖	语料库	商业模型	生成	润色	改写	增强	SIG	DIG
TuringBench [50]	40K	新闻	3	✗	✔	✗	✗	✗	✗	✔
HC3 [17]	85K	问答/评论/学术	5	1	✔	✗	✗	✗	-	-
M4 [53]	24.5K	问答/评论/学术/新闻	11	2	✔	✗	✗	✔	✔	✗
MAGE [30]	29K	问答/评论/新闻/学术/故事	10	3	✔	✗	✗	✔	✔	✗
RAID [11]	628.7K	新闻/学术/评论/文学	11	3	✔	✗	✔	✔	✔	✗
DetectRL [55]	134.4K	学术/评论	4	2	✔	✗	✔	✔	✔	✗
HART [5]	16K	新闻/文学/学术	4	4	✔	✔	✔	✔	✗	✔
MIRAGE (本文)	93.8K	学术/评论/邮件/新闻/网站	10	13	✔	✔	✔	✔	✔	✔

究人员使用 GPT-2 [39] 生成的数据对 RoBERTa [32] 模型进行微调，该模型在检测 GPT 生成的文本方面表现出色。对抗性训练 [15] 被 RADAR [23] 引入以增强 MGTD 的鲁棒性，并使用 PPO [42] 来优化生成器。最近，基于 Fast-DetectGPT [7] 框架的 ImBD [9]，利用 DPO [40] 来优化评分模型，帮助评分模型更好地捕捉训练数据的风格特征从而实现了检测性能的提升。

尽管取得了这些进展，但大多数方法仅仅专注于训练评分模型来拟合源模型的分布，而不是开发一个专用的 MGTD 检测器。这在训练过程中给评分模型带来了约束，而这些约束对 MGTD 任务本身是有害的。

2.3 MGTD 基准

早期的基准如 Turingbench [50] 主要关注由神经模型生成的新闻文章，而 ChatGPT [37] 的出现则将研究重点转移到了大语言模型（LLM）生成的文本上，MGTBench[21] 和 HC3 [17] 便是其中的代表。随后的如 MAGE [30]、MULTITuDE [33] 和 M4 [53] 为代表的工作，则探索了开放领域和多语言检测。RAID [11] 创新性地引入了对解码策略的考量以增强评估的鲁棒性，而 DetectRL [55] 则从写作攻击的角度审视了相关漏洞。然而，这些基准中的大多数都依赖于开源模型（这表明其多样性有限），并且主要关注 MGT，忽视了涉及 MRT 的更常见的真实世界应用，从而限制了它们在现实场景中的适用性。

HART [5] 通过使用六个先进的 LLM（其中只有四个是闭源 LLM）同时涵盖了 MGT 和 MRT，标志了一项进展，但它在生成器的多样性和领域范围上仍然有限。在本研究中，我们将生成器的数量扩大到 17 个，其中包括 13 个闭源 LLM 和 4 个先进的开源 LLM，几乎涵盖了现实世界应用中所有主流的 LLM。此外，我们从五个不同的领域中采样 HWT [20, 43]，并同时生成 MGT 和 MRT，从而确保评估的全面性和代表性。为了推动 MGTD 研究并实现更公平的比较，我们倡导采用统一的基准来确保评估标准的一致性。我们希望 MIRAGE 能为实现这一目标迈出有价值的一步。

3 DetectAnyLLM 框架

DetectAnyLLM 构建于 Fast-DetectGPT [7] 之上，该方法通过衡量原始文本及其扰动变体之间的对数概率差异来判断文本是否为机器生成文本 (MGT) [34]。该方法包含三个关键步骤：*1)* 对给定文本进行重采样，*2)* 计算原始文本与重采样文本之间的差异，以及 *3)* 利用该差异做出判断。我们利用 DDL 训练评分模型以增强步骤 1) 和 2)，从而使检测器能够更容易地区分 MGT 和人类书写文本 (HWT)。在 Section 3.1 中，我们描述了如何计算对数概率差异。接着，在 Section 3.2 中，我们解释了我们对该检测过程进行改进的动机，以及我们引入的具体设计。最后，在 Section 3.3 中，我们详细说明了在我们提出的框架内，差异最终如何用于机器生成文本检测 (MGTD)。

3.1 预备知识

基本假设. 机器生成的文本在每个位置上往往由高概率的词元 (token) 组成，而人类书写的文本则具有更大的可变性。尽管像 top-k 和 top-p 这样的采样策略引入了一定的随机性，但大语言模型 (LLM) 通常仍然会选择概率相对较高的词元。因此，词元概率分布中的特征可以作为区分机器生成文本和人类书写文本的有效线索。

概率差异. 给定一段文本 x 和一个评分模型 f_θ，当使用一个语言模型 q_ϕ 来产生扰动时，其概率差异（即概率曲率）[34] 可以表示为：

$$d(x, f_\theta, q_\phi) = \log f_\theta(x) - \mathbb{E}_{\tilde{x}\sim q_\phi(\cdot|x)}[\log f_\theta(\tilde{x})], \tag{1}$$

其中 $\tilde{x}$ 是 x 经过 q_ϕ 扰动后的版本。

根据该假设，机器生成的文本 x_m 倾向于具有较高的对数概率，而其扰动版本 $\tilde{x_m}$ 则显示出较低的对数概率。相比之下，人类书写的文本 x_h 通常具有较低的对数概率。当被扰动时，$\tilde{x_h}$ 的对数概率往往会增加，因为扰动过程会根据模型将 x_h 中的词语替换为可能性更高的替代词。因此，我们期望实现：

$$d(x_m, f_\theta, q_\phi) > d(x_h, f_\theta, q_\phi). \tag{2}$$

这个不等式构成了机器生成文本检测 (MGTD) 的基础 [7, 9, 34]。

在计算此差异时，获得 f_θ 是直接的，这使得 $\log f_\theta(x)$ 可以被高效地计算。然而，由于对数概率是使用马尔可夫链计算的，即使是微小的扰动也需要重新计算整个链。因此，估计 $\tilde{x}$ 的对数概率期望值是复杂的。

条件概率. [7] 是一种有偏但计算高效的原始概率估计方法：

$$f_\theta(\tilde{x}) = \prod_i f_\theta(\tilde{x}_i|\tilde{x}_{<i}) \sim \prod_i f_\theta(\tilde{x}_i|x_{<i}) = f_\theta(\tilde{x}|x). \tag{3}$$

通过引入方程 (3)，方程 (1) 中的概率差异可以进一步重构为条件概率差异：

$$d_c(x, f_\theta, q_\phi) = \frac{\log f_\theta(x|x) - \tilde{\mu}}{\tilde{\sigma}}, \tag{4}$$

其中

$$\begin{aligned} \tilde{\mu} &= \mathbb{E}_{\tilde{x}\sim q_\phi(\tilde{x}|x)}[\log f_\theta(\tilde{x}|x)], \\ \tilde{\sigma}^2 &= \mathbb{E}_{\tilde{x}\sim q_\phi(\tilde{x}|x)}[\log f_\theta(\tilde{x}|x) - \tilde{\mu}^2]. \end{aligned} \tag{5}$$

注意到差异函数中增加了一个归一化项 $\tilde{\sigma}$，我们在 Section 5.2中进一步探讨了 $\tilde{\sigma}$ 对性能的影响。

重采样文本. 给定一个由 s 个词元组成的句子，我们使用模型 q_ϕ 来计算 $q_\phi(t|x_{<i})$，其中 i 从 1 到 s，t 代表词元。这将生成一个形状为 (s, v) 的张量 $logprobs$，其中 v 表示 q_ϕ 的词汇表大小。利用这个张量，我们仅需一行 PyTorch 代码就能高效地生成 n 个重采样样本。

对于原始版本的概率差异 [34]，扰动生成和差异估计都需要计算 n 次完整的马尔可夫链。这导致时间复杂度为 $O(n \times s)$。

通过引入条件概率 [7]，重采样方法可以替代扰动步骤。在这种表述下，生成 n 个样本和计算差异都只需要运行马尔可夫链**一次**。因此，时间复杂度降低到了 $O(s)$。

3.2 通过直接差异学习进行优化

如方程 (4) 和方程 (2) 所示，提升检测器性能的关键在于，增大由评分模型估计的 MGT 和 HWT 之间条件概率差异的分布差异。

虽然 ImBD [9] 通过引入直接偏好优化 *(DPO)* [40] 来优化评分模型，取得了显著的性能提升，但我们认为 DPO 并非 MGTD 任务的最佳优化方法。

DPO. [40] 源自近端策略优化 *(PPO)* [42] 的优化目标，即：

$$\max_\theta \mathbb{E}_{x\sim f_\theta(x)}[r(x)] - \beta \mathbb{D}_{KL}[f_\theta(x) \,\|\, f_{ref}(x)], \tag{6}$$

其中 x 是从评分模型 f_θ 的分布中采样的文本，r 是一个可以判断样本好坏的奖励函数。通过分析和重参数化该优化目标，我们可以得到 DPO 的优化目标：

$$\max_\theta \mathbb{E}_{x_m,x_h\sim D}[\log \sigma(\beta \log \frac{f_\theta(x_m)}{f_{ref}(x_m)} - \beta \log \frac{f_\theta(x_h)}{f_{ref}(x_h)})], \tag{7}$$

其中 x_m 表示 MGT，x_h 代表 HWT。f_{ref} 是一个参考模型，通常是原始的 f_θ。详细的推导过程将在补充材料中呈现。

受冗余 KL 正则化的启发. 在 PPO [42] 中，f_θ 和 f_{ref} 之间的 KL 散度项被显式地添加到优化目标中，并通过 β 调整其权重，如方程 (6) 所示。而在 DPO [40] 中，如方程 (7) 所示，这种正则化被隐式地嵌入到优化目标中，其强度同样可以通过 β 进行调整。ImBD [9] 直接采用方程(7) 作为其损失函数，并利用成对的 MGT-HWT 数据来优化评分模型 f_θ。KL 正则化迫使评分模型在学习偏好的同时保留其内部知识。

这让我们不禁思考：对于 *MGTD* 任务，在训练过程中保留评分模型的原始知识有何意义？

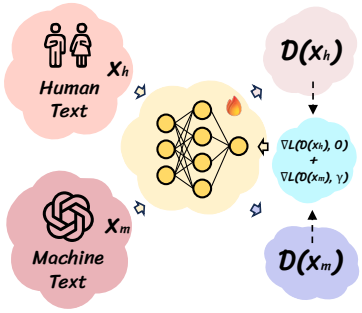

Figure 3: 直接差异学习 (Direct Discrepancy Learning) 概览。评分模型接收成对的 HWT-MGT 数据并为每个样本计算差异。我们对其进行优化，以最小化 HWT 的差异，同时最大化 MGT 的差异。

既然我们引入了训练，我们的直接目标应该是让评分模型能更好地捕捉 MGTD 任务的知识。从根本上说，我们希望训练过程能教会评分模型如何成为一个检测器。然而，KL 正则化彻底改变了这一目标：从学习 MGTD 任务的内在知识，转向使评分模型与训练数据的分布对齐。这将训练过程从学习成为一个检测器转变为模仿一个语言模型，从而误导了评分模型。

直接差异学习. 基于上述推理，我们移除了优化目标中的 KL 正则化。因此，优化目标可以重写为：

$$\max_\theta \mathbb{E}_{x\sim D}[r(x)]. \tag{8}$$

我们进一步设计了一个简单但面向任务的奖励目标 $r(x)$，定义为：

$$r(x) = \begin{cases} -\|\gamma - d_c(x, f_\theta, q_\phi)\|_1, & \text{当 } x \text{ 是 } x_m \text{ 时,} \\ -\|d_c(x, f_\theta, q_\phi)\|_1, & \text{当 } x \text{ 是 } x_h \text{ 时。} \end{cases} \tag{9}$$

其中 γ 是一个超参数。该奖励函数是基于 Section 3.1 中讨论的结论设计的，即人类书写文本 x_h 的差异倾向于较低（接近 0），而机器生成文本 x_m 的差异倾向于为正。引入参数 γ 是为了控制 x_m 的差异应该为多大的正值。在我们的实验中，γ 是任意选择的。如 Table 4 所示，一项关于 γ 值影响的实验表明，模型的性能对这一选择并不特别敏感，这表明模型对 γ 的变化具有一定的鲁棒性。在实践中，我们的输入由成对的 HWT-MGT 数据组成。我们遵循 ImBD [9] 的设置，令 $q_\phi = f_\theta$，这使我们能够使用评分模型的输出来进行优化：

$$\min_\theta \mathbb{E}_{x_m,x_h\sim D}(\|d_c(x_h, f_\theta, f_\theta)\|_1 + \|\gamma - d_c(x_m, f_\theta, f_\theta)\|_1). \tag{10}$$

我们将这种优化方法称为直接差异学习 *(DDL)*，因为它帮助评分模型直接学习 MGT 和 HWT 的期望条件概率差异。

通过移除 KL 正则化，评分模型可以基本上忘记其作为语言模型的身份。此外，基于差异 d_c 并融合了任务导向先验的奖励函数，可以帮助评分模型直接学习 MGTD 的内在知识。具体来说，HWT 的 d_c 接近 0，而 MGT 的 d_c 为正值。

3.3 通过参考聚类进行检测

我们使用参考聚类 *(Reference Clustering)* 来实现从 $d_c(x)$ 到 $p_m(x)$ 的转换。具体而言，该算法旨在估计一个给定值属于特定分布的概率，它包括：数据聚合和概率估计。

数据聚合. 我们首先收集一定数量的 MGT 文本作为 MGT 参考数据集 M，以及数量大致相等 HWT 文本作为 HWT 参考数

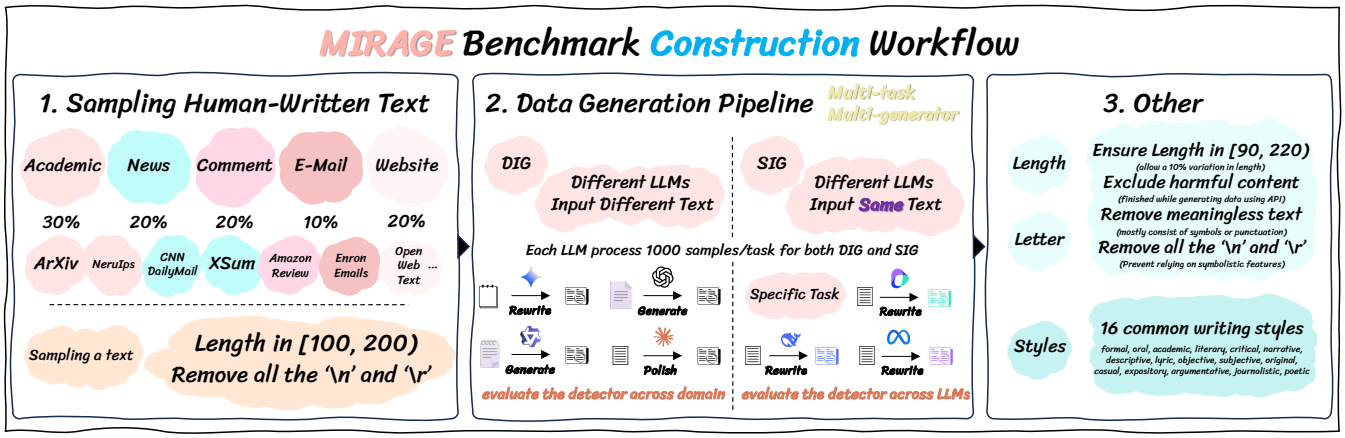

Figure 4: MIRAGE 基准构建工作流程。MIRAGE 包含 9.3 万个 HWT-MGT 对，显著展示了文本领域、源大语言模型和生成任务的多样性，同时使用写作风格控制作为增强手段。

据集 H。然后，我们采用将用于检测的评分模型 f_θ，分别为 M 和 H 中的每段文本计算条件概率差异 d_c。由此，我们可以得到在 f_θ 评分下，M 和 H 中文本的条件概率差异分布 D_m 和 D_h。

概率估计. 我们选择 $M \cup H$ 中与目标值 $d_c(x)$ 第 k 近的值作为搜索窗口 δ：

$$\delta = sorted(\{\|d_c(x_{ref}) - d_c(x)\|_1 \mid x_{ref} \in M \cup H\})[k], \tag{11}$$

其中 k 是一个超参数，应根据参考数据集的大小来确定。对于较大的参考数据集，选择较大的 k 更优，因为它可以提供更高精度的 $p_m(x)$。

然后，我们统计窗口范围内的 MGT 文本和 HWT 文本的数量：

$$\begin{aligned} cnt_m &= \sum_{d \in D_m} I(d_c(x) - \delta < d < d_c(x) + \delta), \\ cnt_h &= \sum_{d \in D_h} I(d_c(x) - \delta < d < d_c(x) + \delta). \end{aligned} \tag{12}$$

最后，我们利用局部统计比率来估计文本 x 属于 MGT 的概率：

$$p_m(x) = \frac{cnt_m}{cnt_m + cnt_h}. \tag{13}$$

由于窗口 δ 是根据数据分布自适应确定的，该方法可以在不同数据密度下保持稳定性，从而提高真实世界 MGTD 的鲁棒性。

4 MIRAGE 基准数据集

现有的基准在文本领域的多样性 [50, 55]、源大语言模型的覆盖范围 [11, 53] 以及评估任务 [17, 30] 方面表现出显著的局限性。

为了促进能更好反映真实世界应用的通用评估，我们提出了 ***M**ulti-domain **I**nclusive **R**ealistic **A**ssessment for machine **G**enerated text d**E**tection (**MIRAGE**)* 基准。MIRAGE 是迄今为止最全面的多任务机器生成文本检测（MGTD）评估框架，它涵盖了不同领域的生成式和修订式文本，并采用了最先进的大语言模型，包括 13 个专有模型和 4 个开源模型。

4.1 基准数据集构建

多领域采样. 考虑到大语言模型在不同文本领域的表现各异，MIRAGE 从 10 个语料库中采样了 5 个领域的人类编写文本（HWT）。详细信息见补充材料。

预清洗. 我们移除了所有 '\n' 字符，以防止检测器基于 '\n' 符号的存在来识别机器生成的文本。随后，我们从这些数据集中筛选出包含 100-200 个单词的文本，以控制基于长度的检测偏差。

包含的 MGT 任务. 我们遵循 [7] 和 [9] 中已有的方法，设计了三种不同的机器生成文本任务：生成（*Generate*）、润色（*Polish*）和重写（*Rewrite*）。生成任务涉及根据一个人类编写文本的前 30 个词元（token）来创建新文本。润色任务在保留原文细节和意义的同时，对现有的人类编写文本进行改进。重写任务在不改变其意义或基本细节的情况下，对给定的人类编写文本进行释义。每个任务的详细提示词将在补充材料中呈现。

真实场景的 LLM 使用. 在真实世界的应用中，人们通常依赖强大的专有大语言模型来生成或修订文本。然而，大多数现有基准 [11, 17, 30, 50, 53, 55] 依赖开源大语言模型来构建数据，这导致当前评估与真实世界应用之间存在差距。为了解决这个问题，MIRAGE 整合了 13 个主流的专有大语言模型，详见补充材料。

同时，认识到高性能开源模型在本地化应用中的部署日益增多，我们整合了四个先进的开源大语言模型 [16, 57]，以确保对当代大语言模型生态系统的全面覆盖。

组合. 我们考虑了两种不同的评估场景，以更好地反映真实世界的应用：

独立输入生成（*Disjoint-Input Generation, DIG*）：每个大语言模型基于一个独一无二的人类编写文本生成机器生成文本（MGT）或机器修订文本（MRT）。检测器必须能够区分机器输出与其源人类编写文本。

共享输入生成（*Shared-Input Generation, SIG*）：多个大语言模型从相同的人类编写文本生成机器生成文本或机器修订文本。检测器必须能够从一个共同的输入中识别出所有的机器输出。

我们设计让每个大语言模型为每项机器生成文本任务生成 2000 个样本，这些样本在 DIG 和 SIG 场景中平均分配（各 1000

Table 2: 在两种评估设置（MIRAGE-DIG 和 MIRAGE-SIG）下，三个任务（Generate、Polish、Rewrite）的测试结果。除 RoBERTa-Base/Large 外，所有方法均采用 GPT-Neo-2.7B [8] 作为评分模型。遵循 [9] 的实验设置，NPR [46] 和 DetectGPT [34] 使用 T5-3B [41] 生成扰动，而 Fast-DetectGPT [7] 则利用 GPT-J-6B [51] 生成样本。♠ 表示基于训练的方法，而 ◇ 表示需要多次模型调用的方法。"Imp." 代表相较于先前最先进技术（SOTA）的提升，计算公式为 $(new - old)/(1.0 - old)$。评估指标包括：AUROC、平衡准确率（Balanced Accuracy）、MCC 和 TPR@5%。DetectAnyLLM 在所有任务和设置中均显著优于所有基线方法。

MIRAGE-DIG (Disjoint-Input Generation)												
Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.4936	0.5091	0.0183	0.0147	0.4653	0.5000	0.0000	0.0214	0.4337	0.5000	0.0000	0.0148
LogRank [25]	0.4992	0.5128	0.0260	0.0220	0.4512	0.5000	0.0000	0.0195	0.4225	0.5000	0.0000	0.0132
Entropy [14]	0.6522	0.6150	0.2543	0.1099	0.5543	0.5417	0.1247	0.0954	0.5805	0.5566	0.1650	0.1189
RoBERTa-Base [32] ♠	0.5523	0.5397	0.1434	0.1250	0.4859	0.5010	0.0088	0.0460	0.5020	0.5049	0.0293	0.0569
RoBERTa-Large [32] ♠	0.4716	0.5217	0.0842	0.0871	0.5171	0.5151	0.0340	0.0633	0.5570	0.5385	0.0864	0.0895
LRR [46]	0.5215	0.5341	0.0777	0.0701	0.4081	0.5000	0.0000	0.0200	0.3930	0.5000	0.0000	0.0188
DNA-GPT [58] ◇	0.5733	0.5595	0.1196	0.0776	0.4771	0.5004	0.0110	0.0309	0.4453	0.5001	0.0080	0.0251
NPR [46] ◇	0.6120	0.6140	0.2604	0.0191	0.5071	0.5370	0.1071	0.0318	0.4710	0.5201	0.0663	0.0226
DetectGPT [34] ◇	0.6402	0.6258	0.2758	0.0275	0.5469	0.5531	0.1328	0.0355	0.5061	0.5266	0.0826	0.0283
Fast-DetectGPT [7]	0.7768	0.7234	0.4628	0.4310	0.5720	0.5570	0.1293	0.1189	0.5455	0.5432	0.1015	0.1025
ImBD [9] ♠	0.8597	0.7738	0.5497	0.4065	0.7888	0.7148	0.4300	0.2730	0.7825	0.7068	0.4139	0.2933
DetectAnyLLM (ours) ♠	**0.9525**	**0.8988**	**0.7975**	**0.7770**	**0.9297**	**0.8732**	**0.7487**	**0.7756**	**0.9234**	**0.8705**	**0.7447**	**0.7778**
Imp.	+66.14%	+55.26%	+55.03%	+62.43%	+66.71%	+55.54%	+55.91%	+69.13%	+64.78%	+55.83%	+56.44%	+68.56%
MIRAGE-SIG (Shared-Input Generation)												
Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.4968	0.5207	0.0196	0.0145	0.4599	0.5002	0.0030	0.0233	0.4319	0.5000	0.0000	0.0111
LogRank [25]	0.5008	0.5183	0.0182	0.0186	0.4468	0.5000	0.0000	0.0211	0.4221	0.5000	0.0000	0.0118
Entropy [14]	0.6442	0.6123	0.1592	0.1074	0.5640	0.5439	0.0516	0.0946	0.5858	0.5645	0.0918	0.1198
RoBERTa-Base [32] ♠	0.5368	0.5392	0.0529	0.1101	0.4741	0.5011	0.0048	0.0395	0.5099	0.5122	0.0221	0.0668
RoBERTa-Large [32] ♠	0.4703	0.5236	0.0417	0.0910	0.5150	0.5157	0.0283	0.0702	0.5576	0.5426	0.0405	0.0762
LRR [46]	0.5214	0.5311	0.0314	0.0657	0.4076	0.5000	0.0000	0.0238	0.3978	0.5000	0.0000	0.0174
DNA-GPT [58] ◇	0.5759	0.5647	0.0603	0.0813	0.4788	0.5001	0.0036	0.0340	0.4457	0.5002	0.0048	0.0258
NPR [46] ◇	0.6088	0.6170	0.1571	0.0185	0.5074	0.5277	0.0612	0.0293	0.4738	0.5204	0.0340	0.0177
DetectGPT [34] ◇	0.6353	0.6241	0.1719	0.0193	0.5434	0.5515	0.0668	0.0309	0.5079	0.5260	0.0431	0.0239
Fast-DetectGPT [7]	0.7706	0.7193	0.2078	0.4200	0.5727	0.5619	0.0607	0.1238	0.5480	0.5495	0.0525	0.1097
ImBD [9] ♠	0.8612	0.7791	0.5599	0.4183	0.7951	0.7199	0.4451	0.3036	0.7694	0.6920	0.3936	0.2868
DetectAnyLLM (ours) ♠	**0.9526**	**0.9059**	**0.8119**	**0.7722**	**0.9316**	**0.8740**	**0.7483**	**0.7779**	**0.9158**	**0.8643**	**0.7320**	**0.7574**
Imp.	+65.85%	+57.40%	+57.25%	+60.84%	+66.62%	+55.02%	+54.64%	+68.11%	+63.49%	+55.94%	+55.80%	+65.98%

个样本）。为了保持一致性，DIG 和 SIG 遵循相同的领域分布，具体细节见补充材料。

采样首先通过按比例合并每个领域内的源数据集来构建领域级的人类编写文本数据集。这种数据集混合策略通过防止从单一数据集中过度采样来减轻数据集偏差。

在实施过程中，SIG 被视为一个独立的"模型"，并在采样过程中与 17 个独立的大语言模型一同被整合。对于每个模型（包括 SIG），我们依次从每个领域数据集中采样数据。一旦采样完成，这些项目将从相应的领域数据集中移除，以保持 DIG 和 SIG 数据之间的区别。在每个文本领域内，数据被持续采样，直到该领域的样本数量满足补充材料中指定的要求。当一个文本领域的数据采样完成后，该过程将转移到下一个领域，重复进行，直到所有文本领域都被采样完毕。

该方法为每个大语言模型生成了 DIG 数据集和一个全面的 SIG 数据集，这些数据集随后被组合起来，构成了每个大语言模型在所有任务上的完整样本集。

数据增强. 语言风格是区分人类编写文本和机器生成文本的一个关键特征，与人类语言风格越接近，对机器生成文本检测器来说任务就越具挑战性。考虑到这一点，我们引入了基于大语言模型语言风格的数据增强。具体来说，我们在输入提示词中加入了"in a <style> style"这一短语。我们手动选择了 16 种不同的语言风格，并在每次大语言模型推理时随机选择一种，以实现风格的多样性。这种方法有助于评估检测器抵抗语言风格攻击的鲁棒性。

后清洗. 在从上述人类编写文本数据生成机器生成文本或机器修订文本后，我们对生成的数据进行清洗。首先，移除所有的'\n' 和 '\r'，以防止检测器利用符号特征进行检测。接着，我们移除少于 90 个单词或多于 220 个单词的文本，以防止文本长度变化对检测产生影响，最终得到 MIRAGE 基准数据集。统计结果在补充材料中呈现。

4.2 评估指标

与先前的工作 [7, 9, 34] 一致，我们采用受试者工作特征曲线下面积（*Area Under the Receiver Operating Characteristic Curve, AUROC*）作为主要评估指标。为了评估在特定阈值下的性能，我们引入了 *5%* 假阳性率下的真阳性率（*TPR@5%*）作为补充指标。此外，考虑到 MIRAGE-SIG 是一个类别不平衡的数据集，我们还报告了马修斯相关系数（*Matthews Correlation Coefficient, MCC*）和平衡准确率（*Balanced Accuracy*），以提供更全面的评估。总的来说，这套多样化的指标对检测器的性能进行了全面评估，确保评估既能反映理论的完整性，又能体现真实世界的适用性。

5 实验

5.1 主要结果

训练设置. DetectAnyLLM 中使用的评分模型和训练数据与 [9] 完全相同，以确保公平比较。详细的训练设置请参见补充材料。DDL 中的 γ 设置为 100，我们将在 Section 5.2 中讨论 γ 如何影响性能。

基准. 为了进行全面比较，我们将我们的方法与基线方法、先进的零样本方法以及当前最先进的基于训练的方法进行了性能比较。基线方法包括 *Likelihood* [44]、*Log-Rank* [25]、*LRR* [46] 和 *Entropy* [14]。先进的零样本方法包括 *DetectGPT* [34]、*NPR* [46] 和 *Fast-DetectGPT* [7]。基于训练的方法包括 *RoBERTa series* [32, 44] 和 *ImBD* [9]。

MIRAGE-DIG 上的结果. 如 Table 2 上半部分所示，我们的方法在所有指标和任务上均取得了超越所有基线方法的显著性能提升。具体而言，其 AUROC 相对增益高达 +64.78%~ +66.71%，MCC 提升高达 +56.44%。DetectAnyLLM 在所有任务中也保持了稳健的 TPR@5%，并以巨大优势（+60.84%~ +69.13%）超越了之前基于训练的 SOTA 方法 ImBD [9]。

MIRAGE-SIG 上的结果. 如 Table 2 下半部分所示，我们的方法在 MIRAGE-SIG 子集上继续保持领先，其 AUROC 达到 0.9526，平衡准确率达到 0.9059，TPR@5% 高达 0.7779，再次大幅超越所有其他方法。

在 MIRAGE 上的结果凸显了 DetectAnyLLM 在不同来源的大语言模型和文本领域中的强大泛化能力和鲁棒性，证明了 DDL 的巨大有效性。

Table 3: 在先前的测试集上的检测结果。♠ 表示基于训练的方法，而 ♢ 表示需要多次模型调用的方法。"Imp." 代表提升，计算公式为 $(new - old)/(1.0 - old)$。

ImBD [9] Test Dataset (GPT-4o polished)			
Methods	XSum [35]	Writing [13]	PubMed [28]
Likelihood [44]	0.4396	0.8077	0.4596
LogRank [25]	0.4002	0.7694	0.4472
Entropy [14]	0.6122	0.2802	0.5899
RoBERTa-Base [32] ♠	0.4921	0.4774	0.2496
RoBERTa-Large [32] ♠	0.4782	0.4708	0.3089
LRR [46]	0.3095	0.6214	0.4710
DNA-GPT [58] ♢	0.4974	0.7478	0.3151
NPR [46] ♢	0.5065	0.8444	0.3740
DetectGPT [34] ♢	0.6217	0.8771	0.5612
Fast-DetectGPT [7]	0.6293	0.8324	0.6175
ImBD [9] ♠	0.9486	0.9468	0.7743
DetectAnyLLM (ours) ♠	**0.9880**	**0.9671**	**0.8817**
Imp.	+80.16%	+38.16%	+47.59%

在先前的测试集上的检测. 我们在 ImBD [9] 使用的三个测试集上评估了 DetectAnyLLM 的性能。如 Table 3 所示，DetectAnyLLM 持续优于所有现有的 MGTD 方法。

通过比较 Table 3 和 Table 2，我们观察到基线方法在 MIRAGE 上的性能出现了严重下降。这一观察揭示了现有测试基准在全面评估检测器能力方面的局限性，从而凸显了 MIRAGE 作为一个更具挑战性基准的重要性。

效率. 由于 DDL 在执行优化时无需依赖参考模型，与 *Style Preference Optimization (SPO)* [9] 相比，其训练效率得到了显著提升。相对于 SPO [9]，DDL 的训练时间减少了 +30.12%，内存消耗降低了 +35.90%。详细信息请参见补充材料。

5.2 消融研究

参数 γ 的消融. 如 Table 4 所示，DDL 对 γ 的取值表现出很强的鲁棒性。与 Table 2 中的结果相比，对于所有选定的 γ 值，经过 DDL 训练的检测器在 AUROC 指标上始终优于所有先前的最先进方法。详细的结果、全面的分析和讨论请参见补充材料。

Table 4: DDL 中不同 γ 值的实验结果。下标为 $_t$ 的指标对应训练集，下标为 $_v$ 的指标表示在 MIRAGE-DIG 的 polish 任务上的评估结果。

	$\gamma = 10$	$\gamma = 20$	$\gamma = 50$	$\gamma = 100$	$\gamma = 500$	$\gamma = 10000$
AUROC_t	**0.9964**	0.9934	0.9883	0.9861	0.9861	0.9861
AUPR_t	**0.9965**	0.9938	0.9888	0.9833	0.9833	0.9833
AUROC_v	0.8692	0.9257	**0.9347**	0.9259	0.9259	0.9259
AUPR_v	0.8735	0.9294	**0.9458**	0.9373	0.9373	0.9373

SPO [9] 中 KL 强度 β 的消融. 我们提供了全面的实验，并证实了我们在 Section 3.2 中提出的观点。更多信息请参见补充材料。

模型大小的消融. 我们使用 SPO [9] 和 DDL 重新训练了 Qwen2-0.5B [56]、GPT-J-6B [51] 和 GPT-Neo-2.7B [8]。然后，这些模型在 MIRAGE-SIG 的 *Rewrite* 任务上进行评估。

如 Table 5 所示，在所有模型尺寸下，经 DDL 优化的检测器始终优于经 SPO [9] 优化的检测器，这证实了 DDL 的鲁棒性和适应性。值得注意的是，使用更小但更先进的大语言模型（如 Qwen2-0.5B 模型）训练的检测器取得了更好的性能。这表明评分模型的能力在很大程度上影响了检测器能力的上限。

Table 5: 关于评分模型影响的消融研究。所有模型均在相同数据上进行训练，并在 MIRAGE-SIG 的 *Rewrite* 任务上进行评估。提升部分用 红色 标记。

Method	Base Model	AUROC	Accuracy	MCC	TPR@5%
SPO [9]	Qwen2-0.5B [56]	0.8570	0.7816	0.5632	0.4508
	GPT-Neo-2.7B [8]	0.7694	0.6920	0.3936	0.2868
	GPT-J-6B [51]	0.8367	0.7557	0.5155	0.4722
DDL (ours)	Qwen2-0.5B [56]	0.9370	0.9071	0.8169	0.8575
		+55.94%	+57.46%	+58.08%	+74.05%
	GPT-Neo-2.7B [8]	0.9158	0.8643	0.7320	0.7574
		+63.49%	+55.94%	+55.80%	+65.98%
	GPT-J-6B [51]	0.8909	0.8424	0.6878	0.6047
		+33.19%	+35.49%	+35.56%	+25.10%

归一化 σ 的消融. 如 Table 6 所示，移除 σ 会导致所有指标的性能大幅下降，这凸显了 σ 对于稳定有效优化的重要性。尽管如此，在 Table 2 报告的大多数指标上，没有归一化的 DDL 仍然超越了先前的最先进方法，这强调了 DDL 的鲁棒性。我们认为，归一化项 σ 有助于标准化来自不同源大语言模型和领域的输出，从而促进更一致和更具泛化性的学习。

Table 6: σ 的消融实验结果。"norm." 表示 "normalization"。"w/" 表示 "with"，"w/o" 表示 "without"。评分模型：GPT-Neo-2.7B [8]。基准测试：MIRAGE-DIG-polish。

	AUROC	Accuracy	MCC	TPR@5%
$\text{DDL}_{\text{w/o } norm.}$	0.8499	0.7759	0.5563	0.5232
$\text{DDL}_{\text{w/ } norm.}$	**0.9297**	**0.8732**	**0.7487**	**0.7756**

6 结论

在这项研究中，我们引入了一种名为直接差异学习（*Direct Discrepancy Learning, DDL*）的新型优化策略，并开发了一个名为 *DetectAnyLLM* 的统一检测框架。我们的方法通过直接利用差异信号使评分模型能够获取面向任务的知识，并通过一种我们称之为参考聚类（*reference clustering*）的技术来实现高精度检测。我们还提出了 *MIRAGE*，这是一个全面的基准数据集，其涵盖了广泛的文本领域、最先进的大语言模型以及多种生成任务。为了全面评估检测器的性能，我们在无交集输入生成（*Disjoint-Input Generation*）和共享输入生成（*Shared-Input Generation*）这两种设置下，对 DetectAnyLLM 及现有的其他 MGTD 方法进行了评估。在 MIRAGE 和先前已有的测试集上的实验结果表明，DetectAnyLLM 的性能显著优于现有的 MGTD 方法，从而在该领域确立了新的技术前沿（state-of-the-art）。

Acknowledgments

本工作部分得到国家自然科学基金（62306153, 62225604）、天津市自然科学基金（24JCJQJC00020）、中国科协青年人才托举工程（YESS20240686）、中央高校基本科研业务费（南开大学, 070-63243143）以及深圳市科技计划项目（JCYJ20240813114237048）的资助。

本研究所使用的计算设备部分由南开大学超级计算中心（NKSC）提供支持。

References

[1] Mervat Abassy, Kareem Elozeiri, Alexander Aziz, Minh Ta, Raj Tomar, Bimarsha Adhikari, Saad Ahmed, Yuxia Wang, Osama Mohammed Afzal, Zhuohan Xie, et al. 2024. LLM-DetectAIve: a Tool for Fine-Grained Machine-Generated Text Detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 336–343.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[3] Anthropic. 2024. *Model Card Addendum: Claude 3.5 Haiku and Upgraded Claude 3.5 Sonnet.* https://assets.anthropic.com/m/1cd9d098ac3e6467/original/Claude-3-Model-Card-October-Addendum.pdf

[4] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351* (2019).

[5] Guangsheng Bao, Lihua Rong, Yanbin Zhao, Qiji Zhou, and Yue Zhang. 2025. Decoupling Content and Expression: Two-Dimensional Detection of AI-Generated Text. arXiv:cs.CL/2503.00258 https://arxiv.org/abs/2503.00258

[6] Guangsheng Bao, Yanbin Zhao, Juncai He, and Yue Zhang. 2025. Glimpse: Enabling White-Box Methods to Use Proprietary Models for Zero-Shot LLM-Generated Text Detection. In *The Thirteenth International Conference on Learning Representations, ICLR*.

[7] 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 *ICLR*.

[8] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.* https://doi.org/10.5281/zenodo.5297715

[9] Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Chen Xinhui, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Long Tang, Lei Zhang, et al. 2025. Imitate Before Detect: Aligning Machine Stylistic Preference for Machine-Revised Text Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 23559–23567.

[10] Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A Smith, and Yejin Choi. 2021. Is GPT-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. *arXiv preprint arXiv:2107.01294* (2021).

[11] Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 12463–12492.

[12] Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 12763–12771.

[13] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, Melbourne, Australia, 889–898. https://doi.org/10.18653/v1/P18-1082

[14] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. GLTR: Statistical Detection and Visualization of Generated Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 111–116.

[15] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[16] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[17] 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. *arXiv preprint arXiv:2301.07597* (2023).

[18] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[19] Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs with binoculars: zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*. 17519–17537.

[20] Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 4693–4703.

[21] Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 2251–2265.

[22] Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic Preference Optimization without Reference Model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 11170–11189.

[23] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems* 36 (2023), 15077–15095.

[24] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276* (2024).

[25] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650* (2019).

[26] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720* (2024).

[27] Ganesh Jawahar, Muhammad Abdul-Mageed, and VS Laks Lakshmanan. 2020. Automatic Detection of Machine Generated Text: A Critical Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2296–2309.

[28] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 2567–2577. https://doi.org/10.18653/v1/D19-1259

[29] Kristian Kuznetsov, Eduard Tulchinskii, Laida Kushnareva, German Magai, Serguei Barannikov, Sergey Nikolenko, and Irina Piontkovskaya. 2024. Robust AI-Generated Text Detection by Restricted Embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 17036–17055.

[30] Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated Text Detection in the Wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 36–53.

[31] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[33] Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, et al. 2023. MULTITuDE: Large-Scale Multilingual Machine-Generated Text Detection Benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 9960–9987.

[34] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*. PMLR, 24950–24962.

[35] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 1797–1807. https://doi.org/10.18653/v1/D18-1206

[36] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[37] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[38] Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023. Deepfake text detection: Limitations and opportunities. In *2023 IEEE symposium on security and privacy (SP)*. IEEE, 1613–1630.

[39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. *Language models are unsupervised multitask learners.*

[40] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2023), 53728–53741.

[41] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[43] Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2204–2213.

[44] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203* (2019).

[45] Mayank Soni and Vincent Wade. 2023. Comparing abstractive summaries generated by ChatGPT to real summaries through blinded reviewers and text classification algorithms. *arXiv preprint arXiv:2303.17650* (2023).

[46] 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*. 12395–12412.

[47] Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2024. The science of detecting LLM-generated text. *Commun. ACM* 67, 4 (2024), 50–59.

[48] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).

[49] Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. Intrinsic dimension estimation for robust detection of AI-generated texts. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 1706, 20 pages.

[50] Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A Benchmark Environment for Turing Test in the Age of Neural Text Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2001–2016.

[51] Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

[52] William Yang Wang. 2017. " Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics.

[53] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, et al. 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)*. 1369–1407.

[54] Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics* (2025), 1–66.

[55] Junchao Wu, Runzhe Zhan, Derek Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia Chao. 2024. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. *Advances in Neural Information Processing Systems* 37 (2024), 100369–100401.

[56] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 Technical Report. *CoRR* (2024).

[57] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115* (2024).

[58] 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 *ICLR*.

[59] Xiao Yu, Kejiang Chen, Qi Yang, Weiming Zhang, and Nenghai Yu. 2024. Text fluoroscopy: Detecting LLM-generated text through intrinsic features. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 15838–15846.

Table 7: 每个领域的源数据集。

Domains	Datasets
Academic	BigPatent [43], NeurIPS, ArXiv, PubMed-Abstracts [28]
eMail	Enron-Emails
Website	OpenWebText
News	CNNDailyMails, XSum [35], XLSum [20]
Comment	Amazon-Review

Table 8: 在分离输入生成（DIG）和共享输入生成（SIG）中，每个大语言模型需要执行生成任务的文本领域构成。

Academic	Mail	Website	News	Comment	**Total**
300	100	200	200	200	1000

A 更多关于直接差异学习（DDL）的细节

A.1 DPO 的推导

Direct Preference Learning (DPO) [40] 源自于 *Proximal Policy Optimization (PPO)* [42] 的优化目标，其公式表示为：

$$\max_{\theta} \mathbb{E}_{x\sim f_\theta(x)}[r(x)] - \beta \mathbb{D}_{KL}[(f_\theta(x) \,\|\, f_{ref}(x)], \tag{14}$$

其中，x 是从评分模型 f_θ 的分布中采样的文本，r 是一个可以判断样本好坏的奖励函数。通过分析并重参数化该优化目标，我们可以得到 DPO 的优化目标。f_{ref} 代表参考模型，通常是原始的 f_θ。DPO [40] 从等式 (14) 的显式解 $f_\theta = p_r$ 出发：

$$p_r(x) = \frac{1}{Z(x)} f_{ref}(x) \exp(\frac{1}{\beta} r(x)), \tag{15}$$

其中：

$$Z(x) = \sum_x f_{ref}(x) \exp(\frac{1}{\beta} r(x)). \tag{16}$$

此外，我们可以将 r 重参数化为：

$$r(x) = \beta \log \frac{p_r(x)}{f_{ref}(x)} + \beta \log Z(x), \tag{17}$$

其中，p_r 是我们希望 f_θ 成为的等式 (14) 的最优解。现在，如果我们引入 Bradley-Terry 模型来表示模型在人类编写的文本（HWT）x_h 和机器生成的文本（MGT）x_m 之间的偏好 L，我们可以得到：

$$\begin{aligned} L(x_m \succ x_h) &= \sigma(r(x_m) - r(x_h)) \\ &= \sigma(\beta \log \frac{p_r(x_m)}{f_{ref}(x_m)} - \beta \log \frac{p_r(x_h)}{f_{ref}(x_h)}), \end{aligned} \tag{18}$$

在这里我们出乎意料地消去了配分函数 $Z(x)$。通过将 p_r 替换为 f_θ 并对等式 (18) 使用最大似然估计，我们最终可以得到 DPO [40] 的优化目标：

$$\max_{\theta} \mathbb{E}_{x_m, x_h \sim D}[\log \sigma(\beta \log \frac{f_\theta(x_m)}{f_{ref}(x_m)} - \beta \log \frac{f_\theta(x_h)}{f_{ref}(x_h)})], \tag{19}$$

其中 x_m 表示机器生成的文本（MGT），x_h 代表人类编写的文本（HWT）。f_{ref} 是一个参考模型，通常是原始的 f_θ。

B 更多关于 MIRAGE 的细节

B.1 更多关于数据来源的细节

时间限制. 为确保所有采样文本均为人类编写，且未受大语言模型生成内容的污染，所使用的大部分源数据集均构建于 2021 年之前。对于包含 2021 年之后收集的数据集，我们对其中标记为 2021 年后收集的数据进行了清洗，以保证人类编写源材料的纯粹性和真实性。

来源领域和数据集. MIRAGE 涵盖了多样的文本领域，包括学术、电子邮件、网站、新闻和评论。这些领域与对应源数据集之间的映射关系总结在表 Table 7 中。

此外，我们实施了针对特定领域的预处理：仅从学术出版物（NeurIPS 和 ArXiv）中提取摘要，并从电子邮件通信（Enron-Emails 数据集）中分离出邮件正文内容。

领域构成. 由于不同领域的数据量各不相同，因此在数量上并未对所有文本领域进行同等处理。然而，对于分离输入生成（DIG）和共享输入生成（SIG），每个大语言模型被要求在各个领域生成或修订的文本比例是固定的。详细的领域分布如表 Table 8 所示。

统计结果. 表 Table 9 展示了 MIRAGE 数据集在两种任务设置下的总体统计数据：分离输入生成（Disjoint-Input Generation）和共享输入生成（Shared-Input Generation）。在每种设置下，数据集均包含三种任务类型——生成 (Gen.)、润色 (Pol.) 和重写 (Rew.)。两种设置下的实例数量是均衡的，每种任务类型大约包含 14,000 到 16,000 个样本。这种均衡的分布确保了数据集能够支持在不同生成和修订场景下对大语言模型进行全面评估。

Table 9: MIRAGE 的统计结果。

	Disjoint-Input Generation			Shared-Input Generation		
Tasks	Gen.	Pol.	Rew.	Gen.	Pol.	Rew.
Count	16412	14776	15735	16388	14776	15751

B.2 源 LLMs

用于在 MIRAGE 中生成数据的源大语言模型罗列在表 Table 10 中。总计，MIRAGE 使用了 13 个强大的商业大语言模型和 4 个先进的开源大语言模型来采样机器生成的文本 (MGT)。这一选择反映了我们重点评估在真实世界应用场景中的检测性能，同时仍然保持对开源大语言模型生态的关注。

Table 10: 商业大语言模型以粗体突出显示。

Series	Models
GPT	**GPT-4o** [24], **GPT-o3-mini** [26], **GPT-4o-mini** [24]
Claude	**Claude-3.5-Haiku**, **Claude-3.7-sonnet** [3]
DeepSeek	**DeepSeek-V3** [31], **DeepSeek-R1** [18]
Gemini	**Gemini-2.0-Flash**, **Gemini-2.0-Flash-Lite** [48]
Qwen	Qwen-2.5-7B [57], Qwen-2.5-7B-R1-Distill [18], **QwQ-Plus**
LlaMa	LlaMA-3.1-8B [16], LlaMa-3.1-8B-R1-Distill [18]
Grok	**Grok2**
Moonshot	**Moonshot-v1**
Doubao	**Doubao-1.5-pro-32k**

Table 11: SPO [9] 中不同 β 值的详细结果。带有下标 $_t$ 的指标对应训练集，下标 $_v$ 表示在 MIRAGE-DIG 的润色任务上的评估结果。Avg.D(*) 表示 * 的平均差异度，其中 $\mathbf{x}_h$ 代表人类编写的文本，$\mathbf{x}_m$ 代表机器生成的文本。ΔD 表示 Avg.D($\mathbf{x}_h$) 和 Avg.D($\mathbf{x}_m$) 之间的距离，更高的 ΔD 通常更有利于区分 $\mathbf{x}_h$ 和 $\mathbf{x}_m$。

β	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.60	0.70	0.80	0.90	0.95
AUROC_t	**0.9490**	0.9192	0.9088	0.9009	0.8945	0.8920	0.8888	0.8871	0.8821	0.8786	0.8742	0.8700	0.8622	0.8554	0.8542
AUPR_t	**0.9566**	0.9277	0.9155	0.9058	0.8982	0.8966	0.8935	0.8934	0.8897	0.8869	0.8846	0.8805	0.8741	0.8695	0.8689
$\text{Avg.D}(\text{x}_h)_t$	-29.14	-13.50	-8.68	-5.72	-4.47	-3.57	-2.92	-2.36	-2.06	-2.05	-1.54	-1.51	-1.37	-0.98	-0.93
$\text{Avg.D}(\text{x}_m)_t$	-11.55	-6.56	-3.88	-2.11	-1.37	-0.79	-0.39	-0.01	0.17	0.14	0.50	0.50	0.53	0.83	0.86
ΔD	17.59	6.94	4.8	3.61	3.10	2.78	2.53	2.35	2.23	2.19	2.04	2.01	1.90	1.81	1.79
AUROC_v	**0.7888**	0.7273	0.7045	0.6669	0.6801	0.6786	0.6745	0.6625	0.6688	0.6672	0.6520	0.6644	0.6499	0.6315	0.6477
AUPR_v	**0.7756**	0.7122	0.6910	0.6602	0.6783	0.6757	0.6764	0.6663	0.6778	0.6719	0.6597	0.6723	0.6578	0.6365	0.6566
$\text{Avg.D}(\text{x}_h)_v$	-26.28	-17.65	-11.50	-7.30	-5.30	-3.78	-2.67	-1.82	-1.33	-1.05	-0.47	-0.45	-0.26	0.32	0.34
$\text{Avg.D}(\text{x}_m)_v$	-18.91	-14.53	-9.94	-6.27	-4.51	-3.11	-2.06	-1.20	-0.74	-0.51	-0.00	0.00	0.10	0.78	0.82
ΔD	7.37	3.12	1.56	1.03	0.79	0.67	0.61	0.62	0.59	0.54	0.47	0.45	0.36	0.46	0.48

B.3 用于构造数据的 Prompt

用于所有三个生成任务的系统提示是相同的，具体来说是："你是一位专业的写作助手，能够撰写高质量、连贯且引人入胜的文章。" 我们在用户提示中添加了风格控制信号，以进行数据增强，从而提升我们基准测试的鲁棒性。

风格控制. 风格控制信号以 "in a <style> style" 的形式直接添加到用户提示中。<style> 是从一个预先准备好的风格列表中随机选择的，详细信息如 Table 12所示。

Table 12: 详细的风格列表。

Style List
formal, oral, academic, literary, critical, narrative, descriptive, lyric, objective, subjective, original, casual, expository, argumentative, journalistic, poetic

用于生成的 Prompt. "请以 <style> 风格写一篇约 150 词的文章，并严格以以下内容开头：<original>"。<original> 是一段人类编写文本（HWT）的前 30 个词符。

用于润色的 Prompt. "请以 style 风格润色以下文本，不要遗漏任何原始细节。确保润色后文本的长度与原文相似。直接输出你润色后的文本。以下是原文：original" <original> 是一段完整的人类编写文本（HWT）。

用于重写的 Prompt. "请以 style 风格转述以下文本，不要遗漏任何原始细节。确保转述后文本的长度与原文相似。直接输出你转述后的文本。以下是原文：original" <original> 是一段完整的人类编写文本（HWT）。

B.4 更多关于评估指标的细节

与先前的工作 [7, 9] 一致，我们采用受试者工作特征曲线下面积 *(AUROC)* 作为评估机器生成文本（MGT）检测器性能的主要指标。虽然 AUROC 提供了一种与阈值无关的分类能力衡量标准，但它不一定能反映检测器在特定工作点上的有效性，而这些工作点在实际部署中通常至关重要。

为了解决这一局限性，我们引入 *5%* 假阳性率下的真阳性率 *(TPR@5%)* 作为一项重要的补充指标。TPR@5% 反映了检测器在严格的假阳性约束下运行时的灵敏度，这对于要求高精度的应用尤其重要。

此外，考虑到 MIRAGE-SIG 是一个类别不平衡的数据集，我们还额外报告了马修斯相关系数 *(MCC)* 和平衡准确率，以提供更全面的评估。MCC 通过考虑混淆矩阵的所有四个要素来捕捉二元分类的整体质量，这使得它在类别不平衡的情况下尤其具有信息价值。平衡准确率取代了标准准确率，其计算方式为真阳性率和真阴性率的算术平均值，因此更适合用于评估在不平衡数据集上的性能。

总而言之，这套多样化的指标为检测器的性能提供了全面的评估，确保了评估结果既能反映理论上的完备性，也兼顾了现实世界中的适用性。

C 更多关于实验的细节

C.1 实验设置

设备. 我们所有的实验均在 Linux 4.18.0 (CentOS 7) 环境下进行，使用单张配备 48GB 显存的 NVIDIA A40 GPU。实验所用的 Python 版本为 3.10.16，PyTorch 版本为 2.5.1，Transformers 版本为 4.47.1，Datasets 版本为 3.2.0。

训练数据集. 我们在 ImBD [9] 所使用的数据集上训练 DetectAnyLLM，具体来说，该数据集包含 500 对 HWT-MGT 数据，其中 MGT 是由 GPT-3.5-Turbo 生成的机器润色文本。

LoRA 配置. 我们遵循 ImBD [9] 的设置，采用专为因果语言建模设计的 LoRA 配置，其秩（rank）为 8，LoRA alpha 值为 32，丢弃率（dropout rate）为 0.1。

复现 ImBD 的设置. 为了进行比较评估，我们遵循原论文中描述的训练配置复现了 ImBD [9]。具体而言，我们将学习率设置为 0.0001，beta 系数设为 0.05。唯一的改动是将训练周期数从原论文报告的 2 轮增加到 5 轮，以确保模型完全收敛。在整个训练过程中，我们监控模型在验证集上的表现以防止过拟合，并验证复现的 ImBD 模型与原始模型性能相当。

训练 DetectAnyLLM 的设置. 我们使用与复现 ImBD [9] 时完全相同的超参数来训练 DetectAnyLLM，包括 0.0001 的学习率和 5 个训练周期。对于直接差异学习 *(Direct Discrepancy Learning, DDL)* 中的优化目标，我们将超参数 γ 设置为 100。这是因为当 γ 超过此值后，模型性能并未进一步提升，表明模型已经完全收敛。此外，由于模型性能在更大的 γ 值下保持稳定，该设置也确保了对不同训练环境的兼容性。因为最优的 γ 值是未知的，我们仅选择一个足够大的值，以提供一个安全且具有泛化性的配置。

C.2 KL-正则化的经验验证

为了评估 DPO 风格训练中 KL-正则化项的影响，我们对一系列 β 值进行了消融研究，这些 β 值直接控制了隐式 KL 约束的

Table 13: DDL 中不同 γ 值的详细结果。下标为 $_t$ 的指标对应训练集，下标为 $_v$ 的指标表示在 MIRAGE-DIG 的润色任务上的评估结果。Avg.D(*) 表示 * 的平均差异，其中 $\mathbf{x}_h$ 代表人类编写的文本，$\mathbf{x}_m$ 代表机器生成的文本。

	$\gamma=1$	$\gamma=2$	$\gamma=5$	$\gamma=10$	$\gamma=20$	$\gamma=30$	$\gamma=40$	$\gamma=50$	$\gamma=60$	$\gamma=70$	$\gamma=80$	$\gamma=90$	$\gamma=100$	$\gamma=500$	$\gamma=10000$
AUROC_t	0.9501	0.9910	**0.9983**	0.9964	0.9934	0.9900	0.9886	0.9883	0.9880	0.9879	0.9861	0.9861	0.9861	0.9861	0.9861
AUPR_t	0.9379	0.9910	**0.9983**	0.9965	0.9938	0.9911	0.9865	0.9888	0.9852	0.9852	0.9833	0.9833	0.9833	0.9833	0.9833
$\text{Avg.D}(\text{x}_h)_t$	0.07	0.14	0.20	0.27	0.54	0.80	1.08	1.45	1.55	1.73	1.91	1.91	1.91	1.91	1.91
$\text{Avg.D}(\text{x}_m)_t$	0.95	1.88	4.86	8.92	17.24	24.64	31.82	37.93	40.81	41.92	42.12	42.12	42.12	42.12	42.12
AUROC_v	0.5481	0.6000	0.7833	0.8692	0.9257	0.9360	**0.9377**	0.9347	0.9251	0.9270	0.9259	0.9259	0.9259	0.9259	0.9259
AUPR_v	0.5206	0.5562	0.7452	0.8735	0.9294	**0.9472**	0.9461	0.9458	0.9401	0.9382	0.9373	0.9373	0.9373	0.9373	0.9373
$\text{Avg.D}(\text{x}_h)_v$	1.1	1.24	1.76	0.84	3.11	3.37	4.66	4.62	4.72	5.36	5.23	5.23	5.23	5.23	5.23
$\text{Avg.D}(\text{x}_m)_v$	1.36	1.85	4.86	7.11	16.09	24.22	32.96	34.86	36.75	39.86	39.43	39.43	39.43	39.43	39.43

Table 14: DDL 与 SPO [9] 的训练时间成本比较。结果在 ImBD [9] 的训练数据集上测试。设备：单张 NVIDIA A40。模型：GPT-J-6B [51]。"Imp." 代表提升（Improvement），计算公式为 -(新值 - 旧值) / (旧值)。

Optim.	Batch Size	Time Cost/Epoch	Memory Usage
SPO [9]	1	166s	31.45GB
DDL(ours)	1	**116s**	**20.16GB**
Imp.	–	+30.12%	+35.90%

强度。根据第 3.2 节中的公式，较大的 β 会强制评分模型 f_θ 与参考模型 f_{ref} 之间进行更强的对齐，从而有效地将学习目标约束在分布一致性上，而非任务特定的可区分性上。

Table 11 提供了有力的经验证据，支持了我们的假设，即 KL-正则化对于 MGTD 任务是多余的，甚至是有害的。随着 β 的增加，我们观察到所有评估指标下的检测性能都出现了持续且显著的下降。例如，在训练集上，当 β 从 0.05 增加到 0.95 时，AUROC 和 AUPR 从 **0.9490**/**0.9566** 下降到 0.8542/0.8689。在验证集上也观察到类似的趋势，其中 AUROC 从 0.7888 下降到 0.6477，AUPR 从 0.7756 下降到 0.6566。

在较低的 β 值下，$\text{D}(\text{x}_h)$ 和 $\text{D}(\text{x}_m)$ 之间的差异是显著的（例如，在 $\beta=0.05$ 时，训练集上的差异为 17.59），这使得评分模型能够有效地区分自然序列和扰动序列。随着 β 的增加，这个差距迅速缩小——在 $\beta=0.30$ 时降至 3.0 以下，并在更高的值时接近于零。验证集也表现出类似的模式：差异从 $\beta=0.05$ 时的 7.37 缩小到 $\beta=0.95$ 时的仅 0.48。

这些结果表明，强 KL-正则化损害了模型从训练数据中学习面向任务的判别性信号的能力。评分模型非但没有成为一个更好的检测器，反而被约束得像一个通用语言模型，从而限制了其区分机器生成文本和人类编写文本的有效性。这验证了我们的理论直觉：虽然 KL 项在通用偏好建模任务中可能有助于保留内部知识，但在 MGTD 任务中却适得其反。

C.3 更多关于主要结果的细节

效率提升. 如第 3.2 节所讨论，当使用评分模型 f_θ 作为采样模型 q_ϕ 时，与 SPO [9] 不同，DDL 在训练过程中无需加载一个独立的参考模型。这种设计使得 DDL 能够用单个模型进行训练，从而显著提升了训练效率。Table 14 详细比较了 SPO [9] 和 DDL 之间的训练时间和内存使用情况。

SPO [9] 在训练时需要同时加载两个大模型，导致了高昂的内存需求——具体来说，使用 GPT-J-6B [51] 进行训练需要 31.45GB 内存。这超过了许多常用 GPU 的容量。相比之下，DDL 仅需 20.16GB 内存，这使得在广泛使用的 GPU 上进行训练成为可能。

C.4 关于 DDL 中 gamma 的讨论

γ 影响的详细分析. 如 Table 13 所示，尽管在特定值上存在明显的性能峰值（例如，训练集上为 $\gamma=5$，验证集上为 $\gamma=30$–40），但在广泛的 γ 值范围内，各项指标都保持在较高水平。例如，即使当 γ 从 10 增加到 10000 时，AUROC_t 和 AUPR_t 也仅出现轻微下降（从约 0.9964 降至 0.9861），而 AUROC_v 和 AUPR_v 在达到各自的最优值后保持稳定。

与此同时，AUROC_v 和 AUPR_v 在 $\gamma=30$–$\gamma=40$ 的区间内持续提升，分别在 $\gamma=30$（AUPR_v = 0.9472）和 $\gamma=40$（AUROC_v = 0.9377）时达到峰值，之后进入平稳期。

DDL 对 γ 的鲁棒性. 尽管平均差异和训练指标随 γ 的增加而变化，但评估性能（AUROC_v 和 AUPR_v）在从 $\gamma=30$ 到 $\gamma=10000$ 的宽泛范围内保持相对稳定。例如，AUROC_v 在 0.93 附近的一个狭窄区间内波动，而 AUPR_v 即使在 γ 发生数量级的变化时也保持在 0.93 以上。这表明，尽管 γ 影响着 x_m 的差异值应为多大的正数，但模型的下游泛化能力对其具体数值并不过分敏感。

这种平稳效应表明，一旦 γ 超过一个适中的阈值，该方法就能保持强大的性能，而不会对值的进一步增加过分敏感。此外，$\text{Avg.D}(\text{x}_h)$ 和 $\text{Avg.D}(\text{x}_m)$ 在某一点后达到饱和，表明模型的行为变得一致，避免了不稳定的变化。在实际应用中，这是一个理想的特性，因为在这些场景下调整像 γ 这样的超参数可能具有挑战性或资源密集。

γ 有效性的解释. 我们的方法对超参数 γ 的鲁棒性源于直接差异学习 *(Direct Discrepancy Learning, DDL)* 的设计。在 DDL 中，γ 作为一个引导优化的边界（margin）：它促使模型将机器生成文本（MGT）的差异分数 $\text{D}(\text{x}_m)$ 保持在接近 γ 的水平，同时将人类创作文本（HWT）的差异分数 $\text{D}(\text{x}_h)$ 最小化至零。这种设置在 HWT 和 MGT 之间的差异空间中构建了一个明确的分离目标。

通过引入这样一个明确的分离目标，我们实现了**学习成为一个检测器，而非另一个语言模型**的目标。

一个较小的 γ（例如 1–5）可能仍能促进分离，但可能无法提供足够的边界，导致 HWT 和 MGT 的差异值之间出现重叠。随着 γ 的增加，模型被激励将 $\text{D}(\text{x}_m)$ 推向离零更远的位置，从而提高可区分性并增强性能——这一点在 AUROC/AUPR 指标从 $\gamma=1$ 到 $\gamma=5$ 及更高值的上升过程中尤为明显。

性能平稳期的解释. 一旦 γ 超过某个阈值（例如 $\gamma\geq30$），我们观察到性能指标（AUROC_t 和 AUROC_v）都达到饱和。这表明 HWT 和 MGT 之间的差异已达到足够大的边界：$\text{D}(\text{x}_h)$ 持续接近 0，而 $\text{D}(\text{x}_m)$ 已经足够大。进一步增加 γ 只会提高 x_m 的目标差异值，而不会改变分类边界。因此，模型趋于稳定，因为

它无法再从更大的 γ 中提取出额外有用的分离信息。这解释了我们观察到的鲁棒性——DDL 在广泛的 γ 值范围内实现了有效分离并保持了高性能。

将 γ 设为 100 的原因. 我们在主要实验中选择 $\gamma = 100$ 主要基于两个原因。首先，如消融实验结果所示，当 $\gamma = 100$ 时性能已经进入平稳期，这意味着该设置在提供高性能的同时，避免了对进一步超参数调优的敏感性——使其成为实际应用中一个实用且可靠的选择。其次，一个更高的 γ 值确保了清晰且一致的差异边界，从而提高了在不同数据集或模型上的可解释性和稳定性。在部署场景中，进行广泛的超参数搜索可能并不可行，因此这种鲁棒性至关重要。

总结. 如 Table 13 所示，存在一个阈值 t_h，使得对于所有 $\gamma \geq t_h$，性能都保持稳定。相反，将 γ 设置得过小会导致性能显著下降，而将其增加到超过 t_h 并不会引起性能急剧下跌。因此，我们建议将 γ 设置为一个相对较大的值，因为在实际应用中，最优值通常是未知的。

C.5 特定大语言模型上的检测结果

我们在特定大语言模型（LLM）的层面上扩展了第 5.1 节中的主要结果，以探究不同方法对特定 LLM 生成文本的检测能力。

结果. 如下表所示，DetectAnyLLM 在所有指标、领域、任务和源 LLM 上均表现出持续强劲的性能。在润色（Polish）和重写（Rewrite）任务上，它以平均近 70% 的优势超越了先前的 SOTA 方法。在涉及特定 LLM 生成文本的某些设置中，FastDetectGPT[7] 和 ImBD[9] 的性能略微优于 DetectAnyLLM。我们认为这是由于生成（Generate）任务相对简单，使得早期的方法也能够表现出竞争力。即便在这些情况下，DetectAnyLLM 的 AUROC 值差距也不超过 10^{-2}，这表明在该任务上性能可能已达到饱和。

如 Table 15 和 Table 18 所示，经 Claude-3.5-Haiku 润色的 DIG 文本的 AUROC 达到了 **0.9903**，而 Claude-3.7-Sonnet 的则为 **0.9096**。类似地，经 GPT-4o-mini 重写的 SIG 文本的 AUROC 达到了 **0.9176**，而 GPT-4o 的则为 **0.8697**。这些结果表明，检测来自较小 LLM 的文本通常比检测其对应更强大模型的文本更容易。

Table 15: 生成器: GPT-4o, GPT-4o-mini. "Imp.": 相对于之前 SOTA 的提升, 计算方式为 $(new - old)/(1.0 - old)$.

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
MIRAGE-DIG, GPT-4o												
Likelihood [44]	0.5021	0.5476	0.1117	0.0040	0.4732	0.5052	0.0227	0.0216	0.4281	0.5000	0.0000	0.0134
LogRank [25]	0.5053	0.5406	0.1011	0.0050	0.4638	0.5005	0.0227	0.0216	0.4191	0.5000	0.0000	0.0113
Entropy [14]	0.6414	0.6259	0.2976	0.1103	0.5358	0.5402	0.1002	0.0670	0.5733	0.5520	0.1953	0.1236
RoBERTa-Base [32] ♠	0.4950	0.5206	0.0857	0.0843	0.5013	0.5036	0.0259	0.0474	0.5244	0.5335	0.0767	0.0639
RoBERTa-Large [32] ♠	0.4478	0.5005	0.0224	0.0361	0.5715	0.5603	0.1207	0.0732	0.6244	0.5896	0.1849	0.0865
LRR [46]	0.5172	0.5281	0.0562	0.0130	0.4334	0.5000	0.0000	0.0206	0.3957	0.5000	0.0000	0.0206
DNA-GPT [58] ◇	0.5886	0.5913	0.1984	0.0171	0.4660	0.5021	0.0321	0.0330	0.4124	0.5010	0.0321	0.0227
NPR [46] ◇	0.5792	0.6068	0.2674	0.0050	0.4648	0.5155	0.0746	0.0361	0.4197	0.5051	0.0529	0.0216
DetectGPT [34] ◇	0.6314	0.6364	0.3132	0.0090	0.4871	0.5201	0.0631	0.0351	0.4371	0.5108	0.0601	0.0227
Fast-DetectGPT [7]	0.8419	0.7683	0.5402	0.3952	0.5251	0.5340	0.0746	0.0515	0.4958	0.5082	0.0168	0.0319
ImBD [9] ♠	0.8971	0.8260	0.6672	0.4253	0.5992	0.5758	0.1990	0.1732	0.6134	0.5824	0.1932	0.1586
DetectAnyLLM(ours) ♠	**0.9503**	**0.9032**	**0.8079**	**0.7743**	**0.9161**	**0.8619**	**0.7256**	**0.7577**	**0.9007**	**0.8435**	**0.6980**	**0.7168**
Imp.	+51.70%	+44.38%	+42.28%	+60.73%	+79.08%	+67.44%	+65.74%	+70.70%	+73.57%	+61.86%	+62.47%	+66.34%
MIRAGE-SIG, GPT-4o												
Likelihood [44]	0.4930	0.5427	0.0975	0.0040	0.4766	0.5046	0.0095	0.0278	0.4275	0.5000	0.0000	0.0103
LogRank [25]	0.4919	0.5407	0.0919	0.0030	0.4647	0.5000	0.0000	0.0247	0.4215	0.5000	0.0000	0.0103
Entropy [14]	0.6475	0.6246	0.2919	0.1176	0.5370	0.5391	0.0894	0.0525	0.5775	0.5622	0.1756	0.1080
RoBERTa-Base [32] ♠	0.4772	0.5106	0.0695	0.0583	0.5040	0.5113	0.0296	0.0443	0.5395	0.5273	0.0690	0.0720
RoBERTa-Large [32] ♠	0.4438	0.5010	0.0317	0.0221	0.5706	0.5608	0.1218	0.0731	0.6172	0.5844	0.1701	0.0874
LRR [46]	0.5006	0.5216	0.0486	0.0161	0.4272	0.5000	0.0000	0.0278	0.4087	0.5000	0.0000	0.0113
DNA-GPT [58] ◇	0.5610	0.5704	0.1482	0.0161	0.4732	0.5005	0.0227	0.0319	0.4222	0.5000	0.0000	0.0226
NPR [46] ◇	0.5619	0.5950	0.2502	0.0080	0.4716	0.5108	0.0647	0.0340	0.4297	0.5031	0.0276	0.0154
DetectGPT [34] ◇	0.5993	0.6171	0.2794	0.0040	0.4890	0.5263	0.0751	0.0247	0.4449	0.5087	0.0442	0.0113
Fast-DetectGPT [7]	0.8266	0.7487	0.4976	0.3588	0.5279	0.5371	0.0742	0.0566	0.4872	0.5062	0.0256	0.0267
ImBD [9] ♠	0.9048	0.8302	0.6690	0.4844	0.6118	0.5860	0.1947	0.1565	0.6120	0.5802	0.1726	0.1235
DetectAnyLLM(ours) ♠	**0.9656**	**0.9382**	**0.8783**	**0.8523**	**0.9080**	**0.8615**	**0.7268**	**0.7353**	**0.8697**	**0.8138**	**0.6410**	**0.6235**
Imp.	+63.87%	+63.61%	+63.22%	+71.35%	+76.29%	+66.54%	+66.07%	+68.62%	+65.95%	+55.20%	+56.45%	+57.04%
MIRAGE-DIG, GPT-4o-mini												
Likelihood [44]	0.5124	0.5523	0.1196	0.0060	0.4839	0.5202	0.0439	0.0174	0.4618	0.5131	0.0285	0.0179
LogRank [25]	0.5194	0.5513	0.1037	0.0090	0.4720	0.5093	0.0194	0.0153	0.4518	0.5053	0.0111	0.0137
Entropy [14]	0.6334	0.6226	0.2698	0.1186	0.5364	0.5447	0.1944	0.1242	0.5398	0.5426	0.1592	0.0988
RoBERTa-Base [32] ♠	0.5024	0.5256	0.1066	0.0995	0.4728	0.5027	0.0194	0.0490	0.5224	0.5168	0.0418	0.0557
RoBERTa-Large [32] ♠	0.5235	0.5176	0.0352	0.0372	0.5811	0.5550	0.1104	0.0926	0.6440	0.6041	0.2106	0.1094
LRR [46]	0.5377	0.5503	0.1005	0.0302	0.4252	0.5000	0.0000	0.0218	0.4190	0.5005	0.0132	0.0158
DNA-GPT [58] ◇	0.5846	0.5889	0.1964	0.0452	0.4770	0.5005	0.0135	0.0218	0.4469	0.5005	0.0229	0.0221
NPR [46] ◇	0.6006	0.6231	0.3056	0.0121	0.4904	0.5441	0.1212	0.0251	0.4261	0.5053	0.0434	0.0252
DetectGPT [34] ◇	0.6327	0.6402	0.3115	0.0161	0.5308	0.5572	0.1533	0.0370	0.4461	0.5142	0.0549	0.0252
Fast-DetectGPT [7]	0.8545	0.7779	0.5599	0.4362	0.5857	0.5752	0.1512	0.0643	0.5111	0.5189	0.0556	0.0336
ImBD [9] ♠	0.9101	0.8322	0.6677	0.5196	0.6767	0.6340	0.3098	0.2440	0.6267	0.6004	0.2265	0.1556
DetectAnyLLM(ours) ♠	**0.9611**	**0.9166**	**0.8336**	**0.8221**	**0.9496**	**0.9009**	**0.8022**	**0.8290**	**0.9209**	**0.8617**	**0.7248**	**0.7350**
Imp.	+56.74%	+50.30%	+49.92%	+62.97%	+84.40%	+72.92%	+71.35%	+77.38%	+77.78%	+65.07%	+64.43%	+68.62%
MIRAGE-SIG, GPT-4o-mini												
Likelihood [44]	0.5099	0.5592	0.1241	0.0110	0.5119	0.5342	0.0725	0.0271	0.4555	0.5000	0.0000	0.0115
LogRank [25]	0.5103	0.5527	0.1134	0.0100	0.4975	0.5212	0.0428	0.0261	0.4432	0.5000	0.0000	0.0105
Entropy [14]	0.6334	0.6108	0.2675	0.1304	0.5117	0.5364	0.1659	0.1021	0.5447	0.5387	0.1808	0.1067
RoBERTa-Base [32] ♠	0.5031	0.5221	0.0693	0.0672	0.4734	0.5000	0.0000	0.0304	0.5338	0.5214	0.0614	0.0680
RoBERTa-Large [32] ♠	0.5418	0.5286	0.0614	0.0692	0.5808	0.5679	0.1361	0.0803	0.6331	0.5973	0.1967	0.0690
LRR [46]	0.5189	0.5391	0.0784	0.0271	0.4432	0.5000	0.0000	0.0185	0.4048	0.5000	0.0000	0.0146
DNA-GPT [58] ◇	0.5687	0.5752	0.1552	0.0321	0.4971	0.5195	0.0425	0.0337	0.4401	0.5000	0.0000	0.0199
NPR [46] ◇	0.5882	0.6128	0.2591	0.0070	0.4868	0.5206	0.0724	0.0358	0.4274	0.5078	0.0542	0.0230
DetectGPT [34] ◇	0.6199	0.6249	0.2804	0.0070	0.5303	0.5451	0.1058	0.0271	0.4446	0.5126	0.0606	0.0251
Fast-DetectGPT [7]	0.8398	0.7658	0.5323	0.3731	0.5860	0.5717	0.1435	0.0749	0.4960	0.5136	0.0422	0.0356
ImBD [9] ♠	0.9158	0.8340	0.6695	0.5637	0.6629	0.6260	0.2812	0.2356	0.6251	0.5988	0.2279	0.1799
DetectAnyLLM(ours) ♠	**0.9656**	**0.9418**	**0.8847**	**0.8365**	**0.9425**	**0.8925**	**0.7861**	**0.8089**	**0.9176**	**0.8708**	**0.7467**	**0.7416**
Imp.	+59.18%	+64.95%	+65.11%	+62.53%	+82.93%	+71.26%	+70.24%	+75.00%	+77.54%	+67.80%	+67.19%	+68.49%

Table 16: 生成器: GPT-o3-mini, Moonshot-v1. "Imp.": 相对于之前 SOTA 的提升, 计算方式为 $(new - old)/(1.0 - old)$.

MIRAGE-DIG, GPT-o3-mini

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.2513	0.5015	0.0317	0.0030	0.4473	0.5022	0.0331	0.0164	0.4064	0.5005	0.0232	0.0086
LogRank [25]	0.2465	0.5010	0.0317	0.0030	0.4319	0.5005	0.0135	0.0120	0.3951	0.5000	0.0000	0.0065
Entropy [14]	0.7324	0.6819	0.3674	0.1573	0.5222	0.5377	0.1187	0.0350	0.5584	0.5500	0.1343	0.1000
RoBERTa-Base [32] ♠	0.4050	0.5000	0.0000	0.0080	0.4395	0.5000	0.0000	0.0175	0.4779	0.5005	0.0056	0.0376
RoBERTa-Large [32] ♠	0.4319	0.5005	0.0224	0.0271	0.5169	0.5055	0.0111	0.0438	0.5680	0.5457	0.0945	0.0742
LRR [46]	0.2687	0.5010	0.0317	0.0040	0.3927	0.5000	0.0000	0.0120	0.3628	0.5000	0.0000	0.0226
DNA-GPT [58] ◇	0.2675	0.5005	0.0068	0.0020	0.4441	0.5038	0.0424	0.0164	0.4124	0.5005	0.0232	0.0129
NPR [46] ◇	0.3819	0.5205	0.1046	0.0040	0.4497	0.5153	0.0604	0.0241	0.3986	0.5048	0.0541	0.0118
DetectGPT [34] ◇	0.3953	0.5210	0.1004	0.0040	0.4720	0.5252	0.0932	0.0263	0.4167	0.5065	0.0514	0.0172
Fast-DetectGPT [7]	0.4107	0.5005	0.0068	0.0291	0.4327	0.5000	0.0000	0.0252	0.3981	0.5000	0.0000	0.0140
ImBD [9] ♠	0.8792	0.8151	0.6468	0.2956	0.6438	0.6138	0.2550	0.1805	0.6563	0.6199	0.2652	0.2140
DetectAnyLLM(ours) ♠	**0.9368**	**0.8993**	**0.7995**	**0.6162**	**0.9008**	**0.8414**	**0.6858**	**0.6751**	**0.8987**	**0.8484**	**0.6986**	**0.7118**
Imp.	+47.68%	+45.53%	+43.24%	+45.52%	+72.15%	+58.92%	+57.82%	+60.35%	+70.53%	+60.11%	+58.99%	+63.34%

MIRAGE-SIG, GPT-o3-mini

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.2405	0.5010	0.0142	0.0000	0.4470	0.5011	0.0232	0.0238	0.4148	0.5000	0.0000	0.0054
LogRank [25]	0.2361	0.5000	0.0000	0.0000	0.4304	0.5005	0.0232	0.0173	0.4039	0.5000	0.0000	0.0064
Entropy [14]	0.7380	0.6795	0.3937	0.1645	0.5294	0.5394	0.0925	0.0454	0.5581	0.5494	0.1298	0.0806
RoBERTa-Base [32] ♠	0.3946	0.5000	0.0000	0.0090	0.4393	0.5000	0.0000	0.0173	0.4689	0.5000	0.0000	0.0333
RoBERTa-Large [32] ♠	0.4466	0.5000	0.0000	0.0221	0.4968	0.5022	0.0093	0.0454	0.5516	0.5387	0.0774	0.0494
LRR [46]	0.2620	0.5000	0.0000	0.0020	0.3888	0.5000	0.0000	0.0194	0.3763	0.5000	0.0000	0.0172
DNA-GPT [58] ◇	0.2672	0.5000	0.0000	0.0000	0.4508	0.5032	0.0232	0.0216	0.4145	0.5005	0.0232	0.0215
NPR [46] ◇	0.3966	0.5110	0.0486	0.0030	0.4536	0.5124	0.0795	0.0270	0.4089	0.5027	0.0439	0.0161
DetectGPT [34] ◇	0.4016	0.5120	0.0341	0.0030	0.4784	0.5205	0.0782	0.0259	0.4321	0.5016	0.0190	0.0183
Fast-DetectGPT [7]	0.4093	0.5000	0.0000	0.0271	0.4400	0.5005	0.0065	0.0205	0.4139	0.5000	0.0000	0.0204
ImBD [9] ♠	0.8963	0.8340	0.6762	0.3942	0.6432	0.6220	0.2589	0.1901	0.6429	0.6139	0.2372	0.1923
DetectAnyLLM(ours) ♠	**0.9528**	**0.9293**	**0.8598**	**0.7773**	**0.8980**	**0.8499**	**0.7008**	**0.6976**	**0.8948**	**0.8528**	**0.7110**	**0.7143**
Imp.	+54.52%	+57.40%	+56.70%	+63.25%	+71.40%	+60.29%	+59.63%	+62.67%	+70.54%	+61.89%	+62.12%	+64.63%

MIRAGE-DIG, Moonshot-v1

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6179	0.6155	0.2547	0.0307	0.5356	0.5565	0.1146	0.0337	0.4927	0.5189	0.0482	0.0263
LogRank [25]	0.6386	0.6207	0.2632	0.0419	0.5273	0.5457	0.0958	0.0348	0.4838	0.5084	0.0173	0.0242
Entropy [14]	0.6193	0.6017	0.2364	0.0532	0.5239	0.5348	0.0839	0.0652	0.5430	0.5425	0.1129	0.0683
RoBERTa-Base [32] ♠	0.6624	0.6334	0.2722	0.2117	0.5236	0.5261	0.0849	0.0891	0.5427	0.5357	0.0924	0.0945
RoBERTa-Large [32] ♠	0.5987	0.5721	0.1670	0.1380	0.5838	0.5679	0.1462	0.1087	0.6118	0.5788	0.1685	0.1355
LRR [46]	0.6878	0.6380	0.2815	0.1237	0.4909	0.5098	0.0196	0.0293	0.4549	0.5000	0.0000	0.0252
DNA-GPT [58] ◇	0.7157	0.6621	0.3426	0.1820	0.5525	0.5549	0.1098	0.0370	0.4995	0.5105	0.0586	0.0462
NPR [46] ◇	0.6852	0.6682	0.3636	0.0215	0.5469	0.5668	0.1551	0.0467	0.4807	0.5189	0.0708	0.0315
DetectGPT [34] ◇	0.6783	0.6595	0.3531	0.0112	0.5860	0.5859	0.1946	0.0511	0.5108	0.5331	0.1038	0.0294
Fast-DetectGPT [7]	**0.9069**	0.8298	0.6603	**0.6319**	0.6829	0.6315	0.2815	0.1837	0.6073	0.5814	0.1842	0.1429
ImBD [9] ♠	0.7824	0.7014	0.4051	0.3405	0.6475	0.6141	0.2693	0.2185	0.6206	0.5930	0.2280	0.1912
DetectAnyLLM(ours) ♠	0.9057	**0.8497**	**0.6996**	0.4468	**0.9243**	**0.8652**	**0.7308**	**0.7163**	**0.9047**	**0.8503**	**0.7036**	**0.7258**
Imp.	–	+11.71%	+11.55%	–	+76.14%	+63.42%	+62.53%	+63.70%	+74.88%	+63.23%	+61.60%	+66.10%

MIRAGE-SIG, Moonshot-v1

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6159	0.6264	0.2707	0.0091	0.5484	0.5562	0.1211	0.0375	0.4993	0.5253	0.0512	0.0169
LogRank [25]	0.6344	0.6244	0.2787	0.0152	0.5372	0.5469	0.1007	0.0331	0.4921	0.5205	0.0411	0.0190
Entropy [14]	0.6177	0.6081	0.2559	0.0548	0.5194	0.5281	0.1102	0.0628	0.5482	0.5511	0.1297	0.0801
RoBERTa-Base [32] ♠	0.6504	0.6223	0.2690	0.1939	0.5128	0.5204	0.0582	0.0706	0.5534	0.5474	0.1178	0.0927
RoBERTa-Large [32] ♠	0.6190	0.5898	0.1871	0.1492	0.5865	0.5656	0.1316	0.0970	0.6171	0.5864	0.1730	0.1106
LRR [46]	0.6810	0.6411	0.2853	0.0975	0.4878	0.5061	0.0235	0.0320	0.4638	0.5000	0.0000	0.0253
DNA-GPT [58] ◇	0.7070	0.6640	0.3327	0.1076	0.5564	0.5507	0.1029	0.0562	0.5271	0.5295	0.0774	0.0390
NPR [46] ◇	0.6649	0.6629	0.3389	0.0254	0.5484	0.5579	0.1314	0.0386	0.4985	0.5353	0.0954	0.0221
DetectGPT [34] ◇	0.6589	0.6492	0.3354	0.0173	0.5872	0.5877	0.1778	0.0430	0.5317	0.5421	0.1267	0.0358
Fast-DetectGPT [7]	0.9113	0.8406	0.6813	0.6254	0.6958	0.6527	0.3131	0.1996	0.6369	0.6001	0.2072	0.1444
ImBD [9] ♠	0.7529	0.6716	0.3626	0.3178	0.6675	0.6356	0.2861	0.2348	0.6221	0.5948	0.2489	0.2192
DetectAnyLLM(ours) ♠	**0.9421**	**0.9056**	**0.8113**	**0.7015**	**0.9248**	**0.8682**	**0.7406**	**0.7497**	**0.8988**	**0.8361**	**0.6795**	**0.6786**
Imp.	+34.72%	+40.76%	+40.81%	+20.33%	+75.27%	+62.06%	+62.23%	+67.29%	+72.12%	+59.03%	+57.33%	+58.84%

Table 17: 生成器: DeepSeek-R1, DeepSeek-V3. "Imp.": 相对于之前 SOTA 的提升, 计算方式为 $(new - old)/(1.0 - old)$.

MIRAGE-DIG, DeepSeek-R1

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.3248	0.5005	0.0224	0.0010	0.3533	0.5000	0.0000	0.0100	0.3285	0.5000	0.0000	0.0093
LogRank [25]	0.2896	0.5005	0.0224	0.0010	0.3235	0.5000	0.0000	0.0089	0.3037	0.5000	0.0000	0.0062
Entropy [14]	0.6454	0.6242	0.2939	0.0902	0.5593	0.5456	0.1194	0.1000	0.5772	0.5542	0.1638	0.1373
RoBERTa-Base [32] ♠	0.5335	0.5000	0.0000	0.0000	0.4483	0.5000	0.0000	0.0000	0.4207	0.5000	0.0000	0.0021
RoBERTa-Large [32] ♠	0.2018	0.5000	0.0000	0.0000	0.2631	0.5000	0.0000	0.0056	0.3501	0.5000	0.0000	0.0196
LRR [46]	0.2159	0.5005	0.0224	0.0000	0.2493	0.5000	0.0000	0.0089	0.2508	0.5000	0.0000	0.0072
DNA-GPT [58] ◇	0.2842	0.5025	0.0448	0.0000	0.3112	0.5000	0.0000	0.0100	0.2962	0.5005	0.0227	0.0114
NPR [46] ◇	0.5706	0.6172	0.2931	0.0130	0.5089	0.5556	0.1648	0.0267	0.4591	0.5253	0.0775	0.0175
DetectGPT [34] ◇	0.6776	0.6623	0.3693	0.0451	0.5792	0.5811	0.2026	0.0400	0.5111	0.5320	0.0912	0.0248
Fast-DetectGPT [7]	0.3838	0.5000	0.0000	0.0230	0.2661	0.5000	0.0000	0.0111	0.2360	0.5000	0.0000	0.0041
ImBD [9] ♠	0.8972	0.8181	0.6363	0.5020	0.7989	0.7333	0.4855	0.4044	0.7307	0.6847	0.3779	0.3189
DetectAnyLLM(ours) ♠	**0.9566**	**0.9088**	**0.8194**	**0.7505**	**0.9720**	**0.9278**	**0.8556**	**0.8978**	**0.9612**	**0.9262**	**0.8532**	**0.8958**
Imp.	+57.78%	+49.86%	+50.36%	+49.90%	+86.09%	+72.92%	+71.93%	+82.84%	+85.58%	+76.60%	+76.40%	+84.70%

MIRAGE-SIG, DeepSeek-R1

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.3301	0.5015	0.0224	0.0040	0.3358	0.5006	0.0235	0.0122	0.3347	0.5000	0.0000	0.0083
LogRank [25]	0.2983	0.5000	0.0000	0.0040	0.3063	0.5000	0.0000	0.0110	0.3098	0.5000	0.0000	0.0072
Entropy [14]	0.6389	0.6048	0.2534	0.1284	0.5739	0.5475	0.1313	0.1094	0.5759	0.5537	0.1895	0.1259
RoBERTa-Base [32] ♠	0.5245	0.5000	0.0000	0.0000	0.4373	0.5000	0.0000	0.0000	0.4201	0.5000	0.0000	0.0010
RoBERTa-Large [32] ♠	0.2007	0.5000	0.0000	0.0000	0.2768	0.5000	0.0000	0.0066	0.3504	0.5005	0.0227	0.0124
LRR [46]	0.2322	0.5000	0.0000	0.0020	0.2396	0.5000	0.0000	0.0055	0.2483	0.5000	0.0000	0.0052
DNA-GPT [58] ◇	0.2830	0.5010	0.0100	0.0000	0.2896	0.5000	0.0000	0.0033	0.2982	0.5000	0.0000	0.0062
NPR [46] ◇	0.5762	0.6138	0.2804	0.0181	0.5115	0.5403	0.1516	0.0221	0.4631	0.5263	0.0867	0.0175
DetectGPT [34] ◇	0.6823	0.6605	0.3485	0.0451	0.5821	0.5751	0.1658	0.0309	0.5135	0.5335	0.1215	0.0351
Fast-DetectGPT [7]	0.3874	0.5000	0.0000	0.0261	0.2580	0.5000	0.0000	0.0088	0.2437	0.5000	0.0000	0.0031
ImBD [9] ♠	0.9087	0.8400	0.6803	0.4835	0.7885	0.7249	0.4693	0.4320	0.7383	0.6827	0.3754	0.3148
DetectAnyLLM(ours) ♠	**0.9617**	**0.9303**	**0.8617**	**0.8646**	**0.9696**	**0.9215**	**0.8437**	**0.8796**	**0.9534**	**0.9174**	**0.8352**	**0.8772**
Imp.	+58.04%	+56.43%	+56.74%	+73.79%	+85.63%	+71.49%	+70.55%	+78.79%	+82.19%	+73.98%	+73.61%	+82.08%

MIRAGE-DIG, DeepSeek-V3

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6430	0.6484	0.3038	0.0201	0.6044	0.5885	0.1849	0.0421	0.5370	0.5610	0.1292	0.0242
LogRank [25]	0.6495	0.6439	0.2991	0.0272	0.5893	0.5785	0.1683	0.0432	0.5242	0.5547	0.1136	0.0231
Entropy [14]	0.5548	0.5805	0.2026	0.0523	0.4305	0.5079	0.0401	0.0285	0.4848	0.5258	0.1164	0.0852
RoBERTa-Base [32] ♠	0.5416	0.5302	0.1092	0.1026	0.4806	0.5000	0.0000	0.0316	0.4764	0.5005	0.0050	0.0410
RoBERTa-Large [32] ♠	0.4194	0.5015	0.0389	0.0322	0.5217	0.5216	0.0439	0.0421	0.5530	0.5373	0.0769	0.0641
LRR [46]	0.6567	0.6363	0.2800	0.0744	0.5189	0.5269	0.0545	0.0443	0.4758	0.5095	0.0206	0.0263
DNA-GPT [58] ◇	0.6980	0.6635	0.3314	0.0885	0.5620	0.5558	0.1308	0.0527	0.5146	0.5310	0.0667	0.0347
NPR [46] ◇	0.7026	0.6977	0.4364	0.0292	0.5712	0.5732	0.1820	0.0558	0.5230	0.5568	0.1394	0.0358
DetectGPT [34] ◇	0.7581	0.7294	0.4737	0.0412	0.6084	0.5959	0.2169	0.0601	0.5656	0.5773	0.1914	0.0379
Fast-DetectGPT [7]	0.9415	0.8732	0.7481	**0.7435**	0.6353	0.6064	0.2148	0.1191	0.6088	0.5910	0.1871	0.0715
ImBD [9] ♠	0.9119	0.8270	0.6550	0.5765	0.6800	0.6444	0.3130	0.2561	0.6930	0.6598	0.3373	0.3028
DetectAnyLLM(ours) ♠	**0.9473**	**0.8999**	**0.8016**	0.7404	**0.9152**	**0.8541**	**0.7122**	**0.7208**	**0.9125**	**0.8649**	**0.7353**	**0.7497**
Imp.	+9.97%	+21.03%	+21.24%	–	+73.49%	+58.96%	+58.10%	+62.46%	+71.51%	+60.28%	+60.06%	+64.10%

MIRAGE-SIG, DeepSeek-V3

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6522	0.6505	0.3055	0.0291	0.6144	0.6056	0.2211	0.0442	0.5545	0.5665	0.1371	0.0211
LogRank [25]	0.6556	0.6454	0.2944	0.0321	0.5980	0.5911	0.1826	0.0453	0.5407	0.5533	0.1066	0.0211
Entropy [14]	0.5455	0.5682	0.1772	0.0532	0.4244	0.5048	0.0388	0.0248	0.4753	0.5322	0.1078	0.0749
RoBERTa-Base [32] ♠	0.5305	0.5246	0.0891	0.0782	0.4630	0.5000	0.0000	0.0409	0.4912	0.5053	0.0386	0.0454
RoBERTa-Large [32] ♠	0.4179	0.5005	0.0224	0.0341	0.5121	0.5086	0.0430	0.0603	0.5509	0.5438	0.0882	0.0538
LRR [46]	0.6556	0.6249	0.2576	0.0612	0.5252	0.5318	0.0641	0.0442	0.4839	0.5084	0.0230	0.0232
DNA-GPT [58] ◇	0.7149	0.6760	0.3687	0.0802	0.5779	0.5722	0.1516	0.0420	0.5239	0.5311	0.0786	0.0359
NPR [46] ◇	0.6938	0.6871	0.4142	0.0281	0.5728	0.5781	0.1673	0.0442	0.5397	0.5570	0.1385	0.0253
DetectGPT [34] ◇	0.7579	0.7197	0.4701	0.0381	0.6136	0.6067	0.2236	0.0506	0.5783	0.5770	0.1724	0.0348
Fast-DetectGPT [7]	0.9359	0.8666	0.7332	0.6971	0.6385	0.6110	0.2237	0.1196	0.6162	0.5833	0.1772	0.1086
ImBD [9] ♠	0.9177	0.8360	0.6728	0.6018	0.6868	0.6439	0.3127	0.2737	0.6880	0.6572	0.3267	0.2985
DetectAnyLLM(ours) ♠	**0.9656**	**0.9343**	**0.8687**	**0.8887**	**0.9286**	**0.8723**	**0.7448**	**0.7392**	**0.9112**	**0.8660**	**0.7329**	**0.7373**
Imp.	+46.32%	+50.75%	+50.80%	+63.25%	+77.21%	+64.15%	+62.87%	+64.09%	+71.54%	+60.92%	+60.33%	+62.56%

Table 18: 生成器: Claude-3.5-Haiku, 3.7-sonnet. "Imp.": 相对于之前 SOTA 的提升, 计算方式为 $(new - old)/(1.0 - old)$.

MIRAGE-DIG, Claude-3.5-haiku

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.3121	0.5000	0.0000	0.0000	0.2962	0.5000	0.0000	0.0000	0.2763	0.5000	0.0000	0.0011
LogRank [25]	0.3153	0.5000	0.0000	0.0000	0.2761	0.5000	0.0000	0.0000	0.2592	0.5000	0.0000	0.0011
Entropy [14]	0.7963	0.7320	0.4736	0.2660	0.7422	0.6918	0.3840	0.1875	0.7518	0.6762	0.3784	0.2338
RoBERTa-Base [32] ♠	0.5036	0.5090	0.0545	0.0510	0.4503	0.5000	0.0000	0.0098	0.4400	0.5000	0.0000	0.0103
RoBERTa-Large [32] ♠	0.4095	0.5005	0.0049	0.0370	0.3540	0.5000	0.0000	0.0110	0.3947	0.5000	0.0000	0.0194
LRR [46]	0.3547	0.5000	0.0000	0.0060	0.2417	0.5000	0.0000	0.0000	0.2319	0.5000	0.0000	0.0023
DNA-GPT [58] ♢	0.4682	0.5020	0.0259	0.0120	0.4072	0.5012	0.0248	0.0086	0.3809	0.5000	0.0000	0.0080
NPR [46] ♢	0.5282	0.5800	0.2244	0.0000	0.4723	0.5484	0.1407	0.0061	0.4520	0.5496	0.1446	0.0034
DetectGPT [34] ♢	0.5598	0.5870	0.2290	0.0050	0.5418	0.5705	0.1724	0.0196	0.5318	0.5661	0.1612	0.0171
Fast-DetectGPT [7]	0.7517	0.6875	0.3768	0.2540	0.6380	0.6134	0.2292	0.1005	0.6291	0.5992	0.2061	0.1163
ImBD [9] ♠	0.9153	0.8460	0.7042	0.5010	0.8748	0.7843	0.5709	0.5061	0.8554	0.7754	0.5510	0.5245
DetectAnyLLM(ours) ♠	**0.9441**	**0.9090**	**0.8195**	**0.6650**	**0.9903**	**0.9602**	**0.9203**	**0.9681**	**0.9796**	**0.9458**	**0.8917**	**0.9396**
Imp.	+33.97%	+40.91%	+38.96%	+32.87%	+92.24%	+81.53%	+81.44%	+93.55%	+85.87%	+75.89%	+75.88%	+87.29%

MIRAGE-SIG, Claude-3.5-haiku

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.3260	0.5000	0.0000	0.0000	0.2745	0.5000	0.0000	0.0012	0.2880	0.5000	0.0000	0.0023
LogRank [25]	0.3267	0.5000	0.0000	0.0000	0.2574	0.5000	0.0000	0.0012	0.2738	0.5000	0.0000	0.0023
Entropy [14]	0.7826	0.7210	0.4428	0.2216	0.7617	0.6951	0.3924	0.2122	0.7445	0.6680	0.3708	0.2526
RoBERTa-Base [32] ♠	0.4608	0.5035	0.0316	0.0282	0.4257	0.5000	0.0000	0.0024	0.4356	0.5000	0.0000	0.0137
RoBERTa-Large [32] ♠	0.4108	0.5000	0.0000	0.0222	0.3683	0.5000	0.0000	0.0085	0.4150	0.5006	0.0239	0.0126
LRR [46]	0.3568	0.5000	0.0000	0.0010	0.2302	0.5000	0.0000	0.0000	0.2540	0.5000	0.0000	0.0023
DNA-GPT [58] ♢	0.4659	0.5146	0.0597	0.0081	0.3960	0.5006	0.0247	0.0085	0.3767	0.5000	0.0000	0.0080
NPR [46] ♢	0.5427	0.5886	0.2348	0.0010	0.4712	0.5433	0.1601	0.0061	0.4507	0.5440	0.1299	0.0046
DetectGPT [34] ♢	0.5757	0.5972	0.2481	0.0000	0.5255	0.5567	0.1426	0.0110	0.5317	0.5577	0.1278	0.0103
Fast-DetectGPT [7]	0.7419	0.6757	0.3529	0.2346	0.6195	0.5921	0.1842	0.1256	0.6152	0.5954	0.1983	0.1326
ImBD [9] ♠	0.9223	0.8550	0.7167	0.5257	0.8808	0.7909	0.5835	0.5451	0.8489	0.7691	0.5430	0.4651
DetectAnyLLM(ours) ♠	**0.9621**	**0.9421**	**0.8858**	**0.8540**	**0.9844**	**0.9433**	**0.8866**	**0.9354**	**0.9759**	**0.9429**	**0.8859**	**0.9326**
Imp.	+51.26%	+60.07%	+59.67%	+69.21%	+86.93%	+72.89%	+72.78%	+85.79%	+84.01%	+75.25%	+75.04%	+87.39%

MIRAGE-DIG, Claude-3.7-sonnet

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.2908	0.5015	0.0317	0.0020	0.3740	0.5000	0.0000	0.0157	0.3061	0.5000	0.0000	0.0083
LogRank [25]	0.2868	0.5010	0.0224	0.0020	0.3619	0.5000	0.0000	0.0168	0.2939	0.5000	0.0000	0.0062
Entropy [14]	0.7081	0.6633	0.3289	0.1528	0.5848	0.5613	0.1928	0.1342	0.6360	0.5937	0.2416	0.1905
RoBERTa-Base [32] ♠	0.4216	0.5000	0.0000	0.0040	0.4425	0.5000	0.0000	0.0126	0.4188	0.5000	0.0000	0.0072
RoBERTa-Large [32] ♠	0.2314	0.5000	0.0000	0.0010	0.3875	0.5000	0.0000	0.0115	0.4109	0.5005	0.0228	0.0280
LRR [46]	0.2995	0.5020	0.0449	0.0060	0.3310	0.5000	0.0000	0.0168	0.2739	0.5000	0.0000	0.0114
DNA-GPT [58] ♢	0.3649	0.5000	0.0000	0.0020	0.4039	0.5000	0.0000	0.0220	0.3407	0.5005	0.0228	0.0197
NPR [46] ♢	0.5223	0.5698	0.1927	0.0070	0.4563	0.5168	0.0630	0.0377	0.4098	0.5036	0.0321	0.0155
DetectGPT [34] ♢	0.5435	0.5754	0.2201	0.0151	0.4833	0.5168	0.0708	0.0283	0.4364	0.5062	0.0443	0.0186
Fast-DetectGPT [7]	0.4048	0.5000	0.0000	0.0211	0.3466	0.5005	0.0229	0.0168	0.2992	0.5000	0.0000	0.0104
ImBD [9] ♠	**0.8576**	0.7920	0.5887	**0.3307**	0.6024	0.5755	0.1640	0.1111	0.6319	0.6020	0.2070	0.1356
DetectAnyLLM(ours) ♠	0.8526	**0.8015**	**0.6057**	0.3136	**0.9096**	**0.8538**	**0.7114**	**0.7201**	**0.9167**	**0.8732**	**0.7483**	**0.7816**
Imp.	–	+4.59%	+4.13%	–	+77.27%	+65.56%	+64.24%	+67.68%	+77.12%	+68.14%	+66.81%	+73.02%

MIRAGE-SIG, Claude-3.7-sonnet

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.3055	0.5000	0.0000	0.0070	0.3663	0.5000	0.0000	0.0178	0.3158	0.5000	0.0000	0.0072
LogRank [25]	0.2986	0.5000	0.0000	0.0050	0.3541	0.5000	0.0000	0.0167	0.3068	0.5000	0.0000	0.0072
Entropy [14]	0.6916	0.6406	0.2926	0.1491	0.5834	0.5575	0.2005	0.1233	0.6278	0.5916	0.2277	0.1543
RoBERTa-Base [32] ♠	0.4125	0.5000	0.0000	0.0020	0.4220	0.5000	0.0000	0.0094	0.4332	0.5000	0.0000	0.0144
RoBERTa-Large [32] ♠	0.2389	0.5000	0.0000	0.0010	0.3955	0.5000	0.0000	0.0261	0.4182	0.5000	0.0000	0.0154
LRR [46]	0.3036	0.5000	0.0000	0.0050	0.3277	0.5000	0.0000	0.0104	0.2974	0.5000	0.0000	0.0093
DNA-GPT [58] ♢	0.3769	0.5000	0.0000	0.0110	0.3917	0.5000	0.0000	0.0230	0.3536	0.5005	0.0227	0.0134
NPR [46] ♢	0.5168	0.5616	0.1623	0.0100	0.4607	0.5094	0.0606	0.0282	0.4104	0.5093	0.0394	0.0103
DetectGPT [34] ♢	0.5366	0.5591	0.1794	0.0180	0.4880	0.5094	0.0433	0.0261	0.4330	0.5041	0.0227	0.0144
Fast-DetectGPT [7]	0.4086	0.5000	0.0000	0.0270	0.3396	0.5000	0.0000	0.0167	0.2944	0.5000	0.0000	0.0113
ImBD [9] ♠	0.8568	0.7858	0.5739	0.3103	0.6237	0.6082	0.2188	0.1160	0.6113	0.5921	0.1842	0.1152
DetectAnyLLM(ours) ♠	**0.8836**	**0.8478**	**0.6966**	**0.4565**	**0.9151**	**0.8600**	**0.7233**	**0.7429**	**0.9102**	**0.8735**	**0.7487**	**0.7623**
Imp.	+18.71%	+28.97%	+28.79%	+21.19%	+77.43%	+64.27%	+64.58%	+70.68%	+75.89%	+68.98%	+67.46%	+71.90%

Table 19: 生成器: Gemini-2.0-flash, flash-lite. "Imp.": 相对于之前 SOTA 的提升, 计算方式为 $(new - old)/(1.0 - old)$.

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
MIRAGE-DIG, Gemini-2.0-flash												
Likelihood [44]	0.4471	0.5020	0.0052	0.0193	0.4069	0.5000	0.0000	0.0197	0.3967	0.5000	0.0000	0.0054
LogRank [25]	0.4445	0.5000	0.0000	0.0234	0.3876	0.5000	0.0000	0.0219	0.3837	0.5000	0.0000	0.0075
Entropy [14]	0.6913	0.6514	0.3071	0.1626	0.6132	0.5864	0.2002	0.1488	0.6299	0.5927	0.2471	0.1726
RoBERTa-Base [32] ♠	0.4250	0.5010	0.0101	0.0274	0.4100	0.5000	0.0000	0.0175	0.4409	0.5000	0.0000	0.0214
RoBERTa-Large [32] ♠	0.2978	0.5005	0.0225	0.0091	0.4322	0.5000	0.0000	0.0208	0.4876	0.5032	0.0232	0.0482
LRR [46]	0.4521	0.5005	0.0225	0.0386	0.3392	0.5000	0.0000	0.0066	0.3523	0.5000	0.0000	0.0086
DNA-GPT [58] ◇	0.6123	0.5971	0.2079	0.0447	0.4558	0.5005	0.0234	0.0295	0.4328	0.5005	0.0232	0.0193
NPR [46] ◇	0.6319	0.6418	0.3309	0.0224	0.5142	0.5454	0.1278	0.0252	0.4957	0.5461	0.1435	0.0182
DetectGPT [34] ◇	0.6715	0.6550	0.3516	0.0366	0.5671	0.5700	0.1670	0.0383	0.5530	0.5648	0.1620	0.0300
Fast-DetectGPT [7]	0.8157	0.7464	0.4929	0.3953	0.5862	0.5706	0.1426	0.0886	0.6026	0.5841	0.1703	0.1125
ImBD [9] ♠	0.8402	0.7597	0.5261	0.3313	0.6612	0.6329	0.2858	0.2243	0.6723	0.6372	0.3046	0.2669
DetectAnyLLM(ours) ♠	**0.9265**	**0.8775**	**0.7558**	**0.6514**	**0.9541**	**0.9114**	**0.8229**	**0.8490**	**0.9559**	**0.9148**	**0.8303**	**0.8767**
Imp.	+54.01%	+49.05%	+48.47%	+42.35%	+86.45%	+75.86%	+75.20%	+80.54%	+86.55%	+76.51%	+75.60%	+83.19%
MIRAGE-SIG, Gemini-2.0-flash												
Likelihood [44]	0.4480	0.5046	0.0107	0.0102	0.4047	0.5000	0.0000	0.0198	0.3812	0.5000	0.0000	0.0097
LogRank [25]	0.4457	0.5025	0.0058	0.0102	0.3892	0.5000	0.0000	0.0165	0.3689	0.5000	0.0000	0.0108
Entropy [14]	0.6923	0.6463	0.3137	0.1376	0.6229	0.5862	0.2435	0.1603	0.6453	0.6016	0.2574	0.1914
RoBERTa-Base [32] ♠	0.4079	0.5005	0.0045	0.0214	0.4098	0.5005	0.0234	0.0198	0.4455	0.5000	0.0000	0.0292
RoBERTa-Large [32] ♠	0.3014	0.5000	0.0000	0.0020	0.4495	0.5005	0.0234	0.0439	0.4829	0.5005	0.0233	0.0432
LRR [46]	0.4568	0.5000	0.0000	0.0204	0.3524	0.5000	0.0000	0.0132	0.3407	0.5000	0.0000	0.0119
DNA-GPT [58] ◇	0.6038	0.5897	0.1968	0.0449	0.4597	0.5000	0.0000	0.0263	0.4227	0.5000	0.0000	0.0205
NPR [46] ◇	0.6156	0.6346	0.2973	0.0224	0.5297	0.5543	0.1354	0.0263	0.4995	0.5481	0.1396	0.0086
DetectGPT [34] ◇	0.6589	0.6488	0.3231	0.0255	0.5820	0.5851	0.1887	0.0285	0.5567	0.5638	0.1656	0.0205
Fast-DetectGPT [7]	0.8090	0.7334	0.4756	0.3802	0.6004	0.5801	0.1604	0.1098	0.5872	0.5692	0.1493	0.1005
ImBD [9] ♠	0.8395	0.7604	0.5217	0.3578	0.6838	0.6454	0.3064	0.2437	0.6738	0.6308	0.2770	0.2097
DetectAnyLLM(ours) ♠	**0.9477**	**0.9139**	**0.8282**	**0.7676**	**0.9536**	**0.9034**	**0.8076**	**0.8419**	**0.9524**	**0.9103**	**0.8212**	**0.8595**
Imp.	+67.44%	+64.04%	+64.08%	+62.50%	+85.31%	+72.76%	+72.26%	+79.10%	+85.40%	+75.70%	+75.27%	+82.22%
MIRAGE-DIG, Gemini-2.0-flash-lite												
Likelihood [44]	0.4082	0.5005	0.0226	0.0041	0.3996	0.5000	0.0000	0.0185	0.3921	0.5000	0.0000	0.0128
LogRank [25]	0.4178	0.5000	0.0000	0.0082	0.3841	0.5000	0.0000	0.0131	0.3824	0.5000	0.0000	0.0117
Entropy [14]	0.7528	0.7028	0.4084	0.1869	0.6432	0.6009	0.2350	0.1778	0.6571	0.6102	0.2473	0.1864
RoBERTa-Base [32] ♠	0.4395	0.5005	0.0035	0.0429	0.4494	0.5000	0.0000	0.0262	0.4534	0.5000	0.0000	0.0330
RoBERTa-Large [32] ♠	0.3808	0.5000	0.0000	0.0184	0.4941	0.5049	0.0420	0.0393	0.5182	0.5106	0.0462	0.0437
LRR [46]	0.4634	0.5005	0.0226	0.0378	0.3462	0.5000	0.0000	0.0164	0.3636	0.5000	0.0000	0.0128
DNA-GPT [58] ◇	0.5756	0.5746	0.1520	0.0429	0.4564	0.5005	0.0234	0.0294	0.4320	0.5000	0.0000	0.0266
NPR [46] ◇	0.6043	0.6185	0.2616	0.0143	0.5133	0.5458	0.1394	0.0229	0.4865	0.5346	0.1249	0.0202
DetectGPT [34] ◇	0.6280	0.6318	0.2966	0.0174	0.5623	0.5671	0.1658	0.0164	0.5272	0.5506	0.1576	0.0202
Fast-DetectGPT [7]	0.8464	0.7712	0.5426	0.4331	0.6605	0.6309	0.2736	0.1679	0.6599	0.6235	0.2482	0.1502
ImBD [9] ♠	0.8604	0.7891	0.5792	0.3626	0.6965	0.6483	0.3153	0.2628	0.6706	0.6400	0.3057	0.2662
DetectAnyLLM(ours) ♠	**0.9122**	**0.8601**	**0.7204**	**0.5485**	**0.9690**	**0.9226**	**0.8453**	**0.8811**	**0.9623**	**0.9164**	**0.8331**	**0.8616**
Imp.	+37.07%	+33.66%	+33.55%	+20.36%	+89.77%	+77.98%	+77.41%	+83.88%	+88.56%	+76.78%	+75.97%	+81.13%
MIRAGE-SIG, Gemini-2.0-flash-lite												
Likelihood [44]	0.4149	0.5000	0.0000	0.0081	0.3951	0.5005	0.0232	0.0173	0.3941	0.5000	0.0000	0.0096
LogRank [25]	0.4167	0.5000	0.0000	0.0091	0.3817	0.5000	0.0000	0.0162	0.3848	0.5000	0.0000	0.0096
Entropy [14]	0.7415	0.6819	0.3639	0.2175	0.6534	0.6138	0.2348	0.1694	0.6658	0.6157	0.2535	0.1815
RoBERTa-Base [32] ♠	0.4224	0.5000	0.0000	0.0183	0.4264	0.5000	0.0000	0.0129	0.4604	0.5000	0.0000	0.0372
RoBERTa-Large [32] ♠	0.3609	0.5000	0.0000	0.0142	0.4813	0.5027	0.0162	0.0464	0.5338	0.5234	0.0475	0.0403
LRR [46]	0.4415	0.5000	0.0000	0.0274	0.3475	0.5000	0.0000	0.0205	0.3658	0.5000	0.0000	0.0127
DNA-GPT [58] ◇	0.5780	0.5742	0.1532	0.0417	0.4567	0.5000	0.0000	0.0194	0.4447	0.5000	0.0000	0.0191
NPR [46] ◇	0.6103	0.6184	0.2723	0.0132	0.5199	0.5431	0.1365	0.0205	0.5008	0.5356	0.1117	0.0138
DetectGPT [34] ◇	0.6321	0.6250	0.2832	0.0163	0.5712	0.5766	0.1749	0.0388	0.5473	0.5536	0.1398	0.0234
Fast-DetectGPT [7]	0.8418	0.7581	0.5178	0.4421	0.6724	0.6311	0.2623	0.1780	0.6705	0.6322	0.2753	0.1900
ImBD [9] ♠	0.8564	0.7769	0.5562	0.3872	0.6971	0.6451	0.3156	0.2621	0.6816	0.6391	0.3121	0.2813
DetectAnyLLM(ours) ♠	**0.9358**	**0.8984**	**0.7968**	**0.7104**	**0.9606**	**0.9186**	**0.8381**	**0.8781**	**0.9513**	**0.9002**	**0.8011**	**0.8291**
Imp.	+55.26%	+54.44%	+54.21%	+48.09%	+87.01%	+77.05%	+76.34%	+83.48%	+84.72%	+72.35%	+71.09%	+76.22%

Table 20: 生成器: Doubao1.5pro, Grok2. "Imp.": 相对于之前 SOTA 的提升, 计算方式为 $(new - old)/(1.0 - old)$.

MIRAGE-DIG, Doubao1.5pro

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.4877	0.5291	0.0707	0.0040	0.4548	0.5073	0.0258	0.0093	0.4377	0.5000	0.0000	0.0105
LogRank [25]	0.5231	0.5486	0.1033	0.0100	0.4496	0.5027	0.0258	0.0093	0.4395	0.5000	0.0000	0.0105
Entropy [14]	0.6627	0.6249	0.2944	0.1073	0.5673	0.5650	0.1887	0.1273	0.5816	0.5634	0.2068	0.1327
RoBERTa-Base [32] ♠	0.4880	0.5045	0.0391	0.0421	0.4709	0.5000	0.0000	0.0305	0.5464	0.5314	0.0757	0.0780
RoBERTa-Large [32] ♠	0.4792	0.5025	0.0448	0.0371	0.5404	0.5345	0.0690	0.0584	0.6026	0.5669	0.1467	0.1013
LRR [46]	0.6335	0.6103	0.2214	0.0712	0.4315	0.5000	0.0000	0.0146	0.4516	0.5006	0.0241	0.0175
DNA-GPT [58] ♢	0.6432	0.6204	0.2529	0.0662	0.5055	0.5345	0.0707	0.0186	0.4844	0.5087	0.0418	0.0407
NPR [46] ♢	0.5756	0.5832	0.1933	0.0150	0.4475	0.5192	0.0659	0.0265	0.4586	0.5169	0.0615	0.0233
DetectGPT [34] ♢	0.5303	0.5667	0.1746	0.0040	0.4883	0.5305	0.0772	0.0358	0.4810	0.5215	0.0761	0.0244
Fast-DetectGPT [7]	0.8007	0.7282	0.4569	**0.3270**	0.5753	0.5590	0.1312	0.0889	0.5470	0.5518	0.1039	0.0477
ImBD [9] ♠	0.8150	0.7452	0.5195	0.2508	0.6552	0.6194	0.2507	0.1976	0.6091	0.5832	0.1703	0.1048
DetectAnyLLM(ours) ♠	**0.8165**	**0.7603**	**0.5291**	0.3039	**0.8795**	**0.8322**	**0.6683**	**0.6419**	**0.7609**	**0.7183**	**0.4576**	**0.3935**
Imp.	+0.80%	+5.91%	+2.00%	–	+65.06%	+55.92%	+55.74%	+55.37%	+38.84%	+32.40%	+31.62%	+30.07%

MIRAGE-SIG, Doubao1.5pro

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.5010	0.5503	0.1150	0.0050	0.4742	0.5161	0.0346	0.0147	0.4436	0.5000	0.0000	0.0069
LogRank [25]	0.5330	0.5683	0.1491	0.0080	0.4677	0.5027	0.0259	0.0188	0.4442	0.5000	0.0000	0.0069
Entropy [14]	0.6532	0.6271	0.2804	0.0945	0.5647	0.5516	0.1573	0.0992	0.5831	0.5677	0.2145	0.1353
RoBERTa-Base [32] ♠	0.4704	0.5141	0.0462	0.0543	0.4572	0.5000	0.0000	0.0295	0.5370	0.5269	0.0802	0.0814
RoBERTa-Large [32] ♠	0.4842	0.5075	0.0705	0.0452	0.5342	0.5228	0.0850	0.0818	0.5943	0.5722	0.1449	0.0917
LRR [46]	0.6346	0.6151	0.2313	0.0724	0.4482	0.5000	0.0000	0.0228	0.4516	0.5000	0.0000	0.0218
DNA-GPT [58] ♢	0.6585	0.6317	0.2682	0.0834	0.5408	0.5436	0.0871	0.0389	0.5006	0.5126	0.0258	0.0344
NPR [46] ♢	0.5742	0.5940	0.2161	0.0141	0.5011	0.5308	0.0952	0.0241	0.4539	0.5120	0.0386	0.0183
DetectGPT [34] ♢	0.5291	0.5719	0.1949	0.0020	0.5430	0.5483	0.1182	0.0201	0.4826	0.5189	0.0479	0.0229
Fast-DetectGPT [7]	0.7906	0.7211	0.4425	0.3146	0.6173	0.5932	0.1865	0.1193	0.5775	0.5568	0.1270	0.0780
ImBD [9] ♠	0.8055	0.7307	0.4691	0.2503	0.6789	0.6280	0.2652	0.1944	0.6258	0.5981	0.2016	0.1342
DetectAnyLLM(ours) ♠	**0.8594**	**0.8095**	**0.6246**	**0.4643**	**0.8920**	**0.8271**	**0.6561**	**0.6327**	**0.7605**	**0.7144**	**0.4468**	**0.4002**
Imp.	+27.73%	+29.29%	+29.29%	+21.85%	+66.37%	+53.51%	+53.21%	+54.41%	+36.01%	+28.96%	+29.57%	+30.64%

MIRAGE-DIG, Grok2

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.5837	0.5876	0.1872	0.0100	0.4545	0.5021	0.0229	0.0231	0.4441	0.5000	0.0000	0.0190
LogRank [25]	0.5964	0.5906	0.1953	0.0140	0.4417	0.5000	0.0000	0.0200	0.4333	0.5000	0.0000	0.0179
Entropy [14]	0.6154	0.6041	0.2320	0.0701	0.5466	0.5478	0.1168	0.0704	0.5579	0.5580	0.1233	0.0770
RoBERTa-Base [32] ♠	0.5690	0.5711	0.1797	0.1311	0.4921	0.5042	0.0172	0.0494	0.4621	0.5000	0.0000	0.0411
RoBERTa-Large [32] ♠	0.4974	0.5210	0.0549	0.0541	0.5505	0.5394	0.0789	0.0557	0.5568	0.5469	0.0940	0.0570
LRR [46]	0.6278	0.6001	0.2006	0.0841	0.4041	0.5000	0.0000	0.0179	0.4042	0.5000	0.0000	0.0116
DNA-GPT [58] ♢	0.6971	0.6582	0.3215	0.1131	0.4612	0.5026	0.0324	0.0252	0.4681	0.5047	0.0325	0.0243
NPR [46] ♢	0.6492	0.6512	0.3476	0.0150	0.4634	0.5184	0.0943	0.0252	0.4350	0.5084	0.0609	0.0148
DetectGPT [34] ♢	0.6595	0.6507	0.3640	0.0140	0.4927	0.5320	0.1188	0.0252	0.4631	0.5148	0.0755	0.0253
Fast-DetectGPT [7]	0.9074	0.8363	0.6730	0.5976	0.5309	0.5373	0.0746	0.0599	0.5274	0.5295	0.0690	0.0496
ImBD [9] ♠	0.8608	0.7828	0.5777	0.3784	0.6094	**0.5893**	0.1813	0.1544	0.6287	0.5918	0.2250	0.1835
DetectAnyLLM(ours) ♠	**0.9323**	**0.8844**	**0.7690**	0.5856	**0.9152**	**0.8713**	**0.7450**	**0.7405**	**0.9257**	**0.8834**	**0.7692**	**0.7890**
Imp.	+26.87%	+29.36%	+29.36%	–	+78.30%	+68.67%	+68.85%	+69.32%	+79.99%	+71.45%	+70.22%	+74.16%

MIRAGE-SIG, Grok2

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.5995	0.6060	0.2332	0.0180	0.4524	0.5000	0.0000	0.0160	0.4274	0.5000	0.0000	0.0159
LogRank [25]	0.6087	0.6050	0.2369	0.0250	0.4366	0.5000	0.0000	0.0139	0.4181	0.5000	0.0000	0.0180
Entropy [14]	0.6037	0.5975	0.2280	0.0450	0.5521	0.5438	0.1002	0.0544	0.5793	0.5693	0.1466	0.0921
RoBERTa-Base [32] ♠	0.5509	0.5605	0.1683	0.1330	0.4425	0.5000	0.0000	0.0235	0.4993	0.5095	0.0255	0.0434
RoBERTa-Large [32] ♠	0.4871	0.5230	0.0909	0.0850	0.5136	0.5117	0.0548	0.0598	0.5573	0.5487	0.0981	0.0540
LRR [46]	0.6355	0.6045	0.2107	0.0860	0.3927	0.5005	0.0103	0.0107	0.3964	0.5000	0.0000	0.0116
DNA-GPT [58] ♢	0.7035	0.6550	0.3172	0.1310	0.4731	0.5123	0.0453	0.0267	0.4491	0.5026	0.0325	0.0233
NPR [46] ♢	0.6515	0.6555	0.3259	0.0160	0.4655	0.5149	0.0727	0.0288	0.4446	0.5143	0.0758	0.0201
DetectGPT [34] ♢	0.6694	0.6615	0.3520	0.0110	0.4990	0.5299	0.0921	0.0224	0.4736	0.5164	0.0682	0.0275
Fast-DetectGPT [7]	0.8946	0.8125	0.6281	0.5790	0.5304	0.5277	0.0718	0.0555	0.5245	0.5291	0.0703	0.0561
ImBD [9] ♠	0.8614	0.7795	0.5608	0.3910	0.6156	0.5993	0.2047	0.1483	0.6067	0.5862	0.1939	0.1397
DetectAnyLLM(ours) ♠	**0.9532**	**0.9240**	**0.8485**	**0.7830**	**0.9231**	**0.8789**	**0.7615**	**0.7769**	**0.9209**	**0.8688**	**0.7393**	**0.7291**
Imp.	+55.61%	+59.47%	+59.26%	+48.46%	+79.99%	+69.77%	+70.01%	+73.81%	+79.88%	+68.29%	+67.65%	+68.51%

Table 21: 生成器: Qwen2.5-7B/R1-Distill. "Imp.": 相对于之前 SOTA 的提升, 计算方式为 $(new - old)/(1.0 - old)$.

MIRAGE-DIG, Qwen2.5-7B-Instruct

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6303	0.6260	0.2634	0.0260	0.4915	0.5223	0.0460	0.0319	0.4605	0.5026	0.0063	0.0201
LogRank [25]	0.6513	0.6375	0.2755	0.0312	0.4833	0.5176	0.0370	0.0277	0.4537	0.5000	0.0000	0.0180
Entropy [14]	0.6164	0.6078	0.2305	0.0771	0.5493	0.5441	0.1845	0.1138	0.5729	0.5618	0.1744	0.0993
RoBERTa-Base [32] ♠	0.6959	0.6568	0.3370	0.2948	0.5456	0.5324	0.0771	0.0596	0.5699	0.5665	0.1437	0.0813
RoBERTa-Large [32] ♠	0.6896	0.6448	0.2898	0.1969	0.6368	0.6117	0.2237	0.1074	0.6536	0.6008	0.2046	0.1341
LRR [46]	0.7032	0.6615	0.3236	0.1000	0.4498	0.5000	0.0000	0.0245	0.4335	0.5000	0.0000	0.0264
DNA-GPT [58] ◇	0.7173	0.6823	0.3689	0.0969	0.5125	0.5234	0.0512	0.0394	0.4841	0.5079	0.0325	0.0296
NPR [46] ◇	0.7013	0.6776	0.3779	0.0323	0.5427	0.5585	0.1282	0.0426	0.4799	0.5185	0.0617	0.0317
DetectGPT [34] ◇	0.7127	0.6849	0.3883	0.0573	0.5584	0.5665	0.1419	0.0415	0.4920	0.5296	0.1020	0.0306
Fast-DetectGPT [7]	**0.9451**	**0.8786**	**0.7576**	**0.7698**	0.6506	0.6213	0.2453	0.1287	0.6125	0.5834	0.1742	0.1130
ImBD [9] ♠	0.7033	0.6594	0.3544	0.3115	0.6133	0.5904	0.1969	0.1596	0.6114	0.5908	0.2047	0.1626
DetectAnyLLM(ours) ♠	0.9252	0.8688	0.7384	0.6052	**0.8638**	**0.7995**	**0.6143**	**0.6170**	**0.8664**	**0.8173**	**0.6371**	**0.6304**
Imp.	–	–	–	–	+61.03%	+47.05%	+48.90%	+54.43%	+61.43%	+54.23%	+54.38%	+55.86%

MIRAGE-SIG, Qwen2.5-7B-Instruct

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6404	0.6375	0.2885	0.0223	0.5047	0.5251	0.0519	0.0320	0.4625	0.5037	0.0074	0.0212
LogRank [25]	0.6613	0.6492	0.3006	0.0340	0.4968	0.5208	0.0423	0.0320	0.4582	0.5000	0.0000	0.0201
Entropy [14]	0.6051	0.5977	0.2278	0.0701	0.5499	0.5421	0.1850	0.1109	0.5803	0.5573	0.1951	0.1241
RoBERTa-Base [32] ♠	0.6832	0.6502	0.3243	0.2335	0.5474	0.5341	0.0896	0.0618	0.5849	0.5668	0.1531	0.1092
RoBERTa-Large [32] ♠	0.6801	0.6343	0.2790	0.1964	0.6305	0.5975	0.1968	0.1279	0.6621	0.6262	0.2524	0.1092
LRR [46]	0.7121	0.6699	0.3401	0.1369	0.4622	0.5027	0.0231	0.0309	0.4432	0.5000	0.0000	0.0276
DNA-GPT [58] ◇	0.7349	0.6948	0.3975	0.0913	0.5193	0.5304	0.0674	0.0437	0.4767	0.5011	0.0103	0.0318
NPR [46] ◇	0.7077	0.6948	0.4114	0.0393	0.5498	0.5581	0.1297	0.0458	0.4982	0.5323	0.0908	0.0308
DetectGPT [34] ◇	0.7007	0.6725	0.3729	0.0393	0.5675	0.5661	0.1599	0.0512	0.5124	0.5403	0.1249	0.0233
Fast-DetectGPT [7]	**0.9398**	0.8758	0.7533	**0.7707**	0.6643	0.6253	0.2535	0.1578	0.6395	0.6050	0.2112	0.1379
ImBD [9] ♠	0.6995	0.6529	0.3484	0.3153	0.6135	0.5933	0.2307	0.1887	0.6125	0.5923	0.1964	0.1580
DetectAnyLLM(ours) ♠	0.9387	**0.8997**	**0.7994**	0.7325	**0.8753**	**0.8140**	**0.6309**	**0.6098**	**0.8714**	**0.8150**	**0.6361**	**0.6278**
Imp.	–	+19.23%	+18.68%	–	+62.85%	+50.36%	+50.56%	+51.91%	+61.95%	+50.50%	+51.32%	+55.79%

MIRAGE-DIG, Qwen2.5-7B-Instruct-R1-Distill

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6056	0.6112	0.2236	0.0110	0.5148	0.5403	0.0861	0.0285	0.5025	0.5158	0.0382	0.0294
LogRank [25]	0.6209	0.6161	0.2354	0.0220	0.5051	0.5285	0.0615	0.0285	0.4965	0.5124	0.0271	0.0238
Entropy [14]	0.6075	0.5972	0.2423	0.0819	0.5483	0.5452	0.1251	0.0855	0.5476	0.5402	0.1042	0.0769
RoBERTa-Base [32] ♠	0.7125	0.6693	0.3600	0.3142	0.5952	0.5713	0.1662	0.1326	0.6098	0.5871	0.1801	0.1041
RoBERTa-Large [32] ♠	0.7827	0.7194	0.4414	0.3839	0.7362	0.6735	0.3474	0.2305	0.7330	0.6708	0.3419	0.2138
LRR [46]	0.6569	0.6247	0.2525	0.0746	0.4698	0.5006	0.0111	0.0273	0.4762	0.5017	0.0238	0.0226
DNA-GPT [58] ◇	0.6482	0.6235	0.2501	0.0905	0.5148	0.5297	0.0879	0.0446	0.4872	0.5107	0.0412	0.0238
NPR [46] ◇	0.6416	0.6479	0.3286	0.0257	0.5137	0.5502	0.1423	0.0335	0.4991	0.5288	0.0870	0.0362
DetectGPT [34] ◇	0.6480	0.6449	0.3276	0.0281	0.5422	0.5582	0.1511	0.0409	0.5157	0.5390	0.0958	0.0385
Fast-DetectGPT [7]	0.9207	0.8454	0.6938	**0.6626**	0.7091	0.6543	0.3110	0.2714	0.6665	0.6295	0.2742	0.1912
ImBD [9] ♠	0.7347	0.6705	0.4022	0.3778	0.6698	0.6388	0.3202	0.2788	0.6339	0.6114	0.2673	0.2319
DetectAnyLLM(ours) ♠	**0.9332**	**0.8778**	**0.7556**	0.6577	**0.8962**	**0.8445**	**0.6912**	**0.6766**	**0.8758**	**0.8167**	**0.6427**	**0.6210**
Imp.	+15.74%	+20.95%	+20.19%	–	+60.66%	+52.37%	+52.69%	+55.15%	+53.48%	+44.33%	+45.72%	+50.66%

MIRAGE-SIG, Qwen2.5-7B-Instruct-R1-Distill

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6183	0.6190	0.2462	0.0110	0.4994	0.5212	0.0425	0.0218	0.4862	0.5137	0.0280	0.0102
LogRank [25]	0.6348	0.6209	0.2531	0.0208	0.4932	0.5139	0.0282	0.0194	0.4789	0.5097	0.0208	0.0114
Entropy [14]	0.5841	0.5842	0.2179	0.0549	0.5621	0.5484	0.1204	0.0944	0.5596	0.5535	0.1502	0.0774
RoBERTa-Base [32] ♠	0.7115	0.6697	0.3658	0.2821	0.5758	0.5617	0.1558	0.1138	0.5896	0.5700	0.1768	0.1502
RoBERTa-Large [32] ♠	0.7808	0.7131	0.4293	0.3370	0.7082	0.6465	0.3040	0.2288	0.7265	0.6706	0.3430	0.2071
LRR [46]	0.6778	0.6404	0.2852	0.0745	0.4663	0.5012	0.0246	0.0291	0.4610	0.5006	0.0239	0.0159
DNA-GPT [58] ◇	0.6659	0.6374	0.2809	0.0904	0.4948	0.5236	0.0571	0.0266	0.4776	0.5142	0.0579	0.0375
NPR [46] ◇	0.6475	0.6471	0.3331	0.0281	0.5173	0.5381	0.1107	0.0230	0.4915	0.5301	0.0889	0.0250
DetectGPT [34] ◇	0.6615	0.6416	0.3127	0.0379	0.5420	0.5539	0.1267	0.0375	0.5092	0.5296	0.1173	0.0319
Fast-DetectGPT [7]	0.9192	0.8462	0.6923	0.6874	0.7019	0.6429	0.3162	0.2627	0.6624	0.6183	0.2419	0.1593
ImBD [9] ♠	0.7005	0.6709	0.4018	0.3736	0.6638	0.6380	0.2987	0.2542	0.6658	0.6320	0.2840	0.2378
DetectAnyLLM(ours) ♠	**0.9524**	**0.9176**	**0.8353**	**0.8059**	**0.8906**	**0.8305**	**0.6698**	**0.6768**	**0.8673**	**0.8146**	**0.6440**	**0.6246**
Imp.	+41.06%	+46.43%	+46.46%	+37.89%	+62.51%	+52.05%	+51.72%	+56.16%	+51.47%	+43.70%	+45.82%	+50.75%

Table 22: 生成器: LlaMa3.1-8B/R1-Distill. "Imp.": 相对于之前 SOTA 的提升, 计算方式为 $(new - old)/(1.0 - old)$.

MIRAGE-DIG, LlaMa3.1-8B-Instruct

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.7698	0.7395	0.4823	0.1247	0.6213	0.6195	0.2509	0.0260	0.5904	0.5988	0.2174	0.0186
LogRank [25]	0.7944	0.7588	0.5212	0.1865	0.6093	0.6008	0.2217	0.0228	0.5811	0.5837	0.1864	0.0221
Entropy [14]	0.5764	0.5690	0.1623	0.0552	0.5324	0.5545	0.1345	0.0374	0.5726	0.5756	0.1799	0.0616
RoBERTa-Base [32] ♠	0.8327	0.7682	0.5403	0.5121	0.5748	0.5496	0.1175	0.1073	0.6350	0.6099	0.2532	0.2035
RoBERTa-Large [32] ♠	0.7945	0.7246	0.4697	0.4161	0.6461	0.6220	0.2495	0.0943	0.6999	0.6506	0.3040	0.1651
LRR [46]	0.8405	0.7710	0.5436	0.4029	0.5503	0.5512	0.1197	0.0293	0.5408	0.5448	0.0909	0.0384
DNA-GPT [58] ♢	0.8741	0.8151	0.6308	0.4625	0.6908	0.6528	0.3298	0.0992	0.6239	0.6035	0.2148	0.0558
NPR [46] ♢	0.7772	0.7467	0.5155	0.0695	0.5993	0.6065	0.2521	0.0211	0.6029	0.5983	0.2418	0.0244
DetectGPT [34] ♢	0.7984	0.7610	0.5471	0.0673	0.6516	0.6398	0.3069	0.0325	0.6509	0.6320	0.2990	0.0407
Fast-DetectGPT [7]	**0.9944**	**0.9741**	**0.9482**	**0.9845**	0.8546	0.7732	0.5515	0.5057	0.8682	0.7983	0.5965	0.5256
ImBD [9] ♠	0.8708	0.7787	0.5675	0.4691	0.7267	0.6772	0.3798	0.3447	0.7212	0.6715	0.3894	0.3302
DetectAnyLLM(ours) ♠	0.9235	0.8499	0.7002	0.6645	**0.9467**	**0.8919**	**0.7839**	**0.7935**	**0.9678**	**0.9198**	**0.8417**	**0.8872**
Imp.	–	–	–	–	+63.32%	+52.33%	+51.81%	+58.22%	+75.58%	+60.23%	+60.77%	+76.23%

MIRAGE-SIG, LlaMa3.1-8B-Instruct

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.7701	0.7461	0.4970	0.1057	0.6071	0.6153	0.2497	0.0265	0.5754	0.5935	0.1979	0.0150
LogRank [25]	0.7934	0.7500	0.5073	0.1641	0.5951	0.5995	0.2108	0.0216	0.5683	0.5751	0.1734	0.0162
Entropy [14]	0.5545	0.5600	0.1516	0.0595	0.5632	0.5713	0.1622	0.0348	0.5732	0.5768	0.1772	0.0612
RoBERTa-Base [32] ♠	0.8226	0.7698	0.5612	0.5055	0.5325	0.5390	0.0955	0.0879	0.6481	0.6195	0.2689	0.2182
RoBERTa-Large [32] ♠	0.7850	0.7230	0.4492	0.3678	0.6316	0.6078	0.2188	0.1161	0.7055	0.6542	0.3091	0.1709
LRR [46]	0.8363	0.7671	0.5384	0.4042	0.5321	0.5398	0.0899	0.0265	0.5391	0.5410	0.0983	0.0393
DNA-GPT [58] ♢	0.8866	0.8271	0.6547	0.4901	0.6783	0.6468	0.2936	0.0879	0.6061	0.5901	0.1969	0.0450
NPR [46] ♢	0.7856	0.7561	0.5465	0.0628	0.5999	0.6012	0.2367	0.0282	0.5888	0.5987	0.2250	0.0162
DetectGPT [34] ♢	0.7999	0.7643	0.5360	0.0308	0.6508	0.6260	0.2782	0.0332	0.6450	0.6253	0.2810	0.0300
Fast-DetectGPT [7]	**0.9931**	**0.9725**	**0.9450**	**0.9791**	0.8641	0.7828	0.5706	0.5307	0.8519	0.7760	0.5533	0.4988
ImBD [9] ♠	0.8757	0.7880	0.5798	0.4824	0.7514	0.6915	0.4019	0.3516	0.7152	0.6617	0.3725	0.3372
DetectAnyLLM(ours) ♠	0.9492	0.9091	0.8185	0.7709	**0.9493**	**0.8905**	**0.7825**	**0.8043**	**0.9589**	**0.9042**	**0.8090**	**0.8395**
Imp.	–	–	–	–	+62.71%	+49.62%	+49.34%	+58.30%	+72.26%	+57.22%	+57.24%	+67.97%

MIRAGE-DIG, LlaMa3.1-8B-Instruct-R1-Distill

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6583	0.6445	0.3108	0.0095	0.5952	0.5922	0.2220	0.0301	0.5359	0.5509	0.1170	0.0284
LogRank [25]	0.6726	0.6522	0.3143	0.0166	0.5910	0.5869	0.1945	0.0235	0.5282	0.5410	0.0953	0.0252
Entropy [14]	0.5717	0.5904	0.2487	0.0392	0.4731	0.5235	0.0910	0.0366	0.5138	0.5383	0.0984	0.0438
RoBERTa-Base [32] ♠	0.7288	0.6790	0.3734	0.3329	0.5680	0.5660	0.1803	0.1229	0.5836	0.5755	0.1562	0.1072
RoBERTa-Large [32] ♠	0.7414	0.6748	0.3690	0.2723	0.6510	0.6190	0.2380	0.1281	0.6647	0.6214	0.2431	0.1400
LRR [46]	0.7062	0.6522	0.3065	0.1617	0.5596	0.5588	0.1244	0.0222	0.4977	0.5131	0.0354	0.0339
DNA-GPT [58] ♢	0.6855	0.6576	0.3238	0.0892	0.5585	0.5588	0.1312	0.0392	0.5109	0.5246	0.0593	0.0339
NPR [46] ♢	0.6476	0.6468	0.3011	0.0131	0.5575	0.5725	0.1720	0.0484	0.5016	0.5306	0.0968	0.0263
DetectGPT [34] ♢	0.6546	0.6457	0.3247	0.0107	0.5843	0.5889	0.2114	0.0366	0.5276	0.5492	0.1126	0.0252
Fast-DetectGPT [7]	0.9223	0.8484	0.6976	0.6754	0.7160	0.6660	0.3363	0.2209	0.6426	0.6034	0.2176	0.1477
ImBD [9] ♠	0.7661	0.7081	0.4537	0.4328	0.6463	0.6131	0.2858	0.2431	0.6261	0.6034	0.2611	0.2287
DetectAnyLLM(ours) ♠	**0.9541**	**0.8995**	**0.7995**	**0.7800**	**0.8949**	**0.8359**	**0.6721**	**0.6523**	**0.8855**	**0.8233**	**0.6554**	**0.6554**
Imp.	+40.95%	+33.73%	+33.68%	+32.23%	+63.00%	+50.88%	+50.61%	+54.06%	+65.85%	+53.32%	+53.36%	+55.32%

MIRAGE-SIG, LlaMa3.1-8B-Instruct-R1-Distill

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.6540	0.6486	0.3186	0.0133	0.6019	0.5920	0.2049	0.0363	0.5411	0.5569	0.1167	0.0184
LogRank [25]	0.6707	0.6594	0.3254	0.0242	0.5962	0.5803	0.1821	0.0337	0.5350	0.5450	0.0930	0.0238
Entropy [14]	0.5621	0.5815	0.2278	0.0314	0.4736	0.5207	0.0813	0.0350	0.5050	0.5336	0.1286	0.0390
RoBERTa-Base [32] ♠	0.7119	0.6781	0.3721	0.2717	0.5722	0.5531	0.1383	0.1166	0.5891	0.5720	0.1709	0.1181
RoBERTa-Large [32] ♠	0.7276	0.6649	0.3345	0.2766	0.6661	0.6211	0.2467	0.1710	0.6808	0.6376	0.2755	0.1463
LRR [46]	0.7070	0.6673	0.3491	0.1196	0.5612	0.5544	0.1094	0.0453	0.5108	0.5200	0.0508	0.0368
DNA-GPT [58] ♢	0.6989	0.6606	0.3215	0.0809	0.5693	0.5680	0.1498	0.0544	0.5122	0.5276	0.0907	0.0238
NPR [46] ♢	0.6567	0.6479	0.3291	0.0121	0.5764	0.5771	0.1859	0.0389	0.5224	0.5379	0.1224	0.0368
DetectGPT [34] ♢	0.6618	0.6582	0.3413	0.0085	0.6004	0.5933	0.2188	0.0544	0.5405	0.5504	0.1583	0.0358
Fast-DetectGPT [7]	0.9189	0.8424	0.6850	0.6510	0.7247	0.6645	0.3315	0.2396	0.6474	0.6051	0.2194	0.1603
ImBD [9] ♠	0.7669	0.6987	0.4448	0.4263	0.6606	0.6276	0.3144	0.2617	0.6309	0.6056	0.2633	0.2297
DetectAnyLLM(ours) ♠	**0.9642**	**0.9245**	**0.8497**	**0.8563**	**0.8743**	**0.8180**	**0.6439**	**0.6127**	**0.8776**	**0.8207**	**0.6470**	**0.6186**
Imp.	+55.90%	+52.11%	+52.28%	+58.82%	+54.32%	+45.75%	+46.73%	+47.54%	+61.66%	+50.52%	+51.27%	+50.49%

Table 23: 生成器: QwQ-Plus-32B. "Imp.": 相对于之前 SOTA 的提升, 计算方式为 $(new - old)/(1.0 - old)$.

MIRAGE-DIG, QwQ-Plus-32B												
Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.3957	0.5083	0.0568	0.0010	0.4576	0.5065	0.0255	0.0156	0.3788	0.5000	0.0000	0.0068
LogRank [25]	0.3715	0.5062	0.0525	0.0041	0.4298	0.5006	0.0255	0.0130	0.3553	0.5000	0.0000	0.0068
Entropy [14]	0.6403	0.6165	0.2723	0.0714	0.5070	0.5234	0.0806	0.0519	0.5786	0.5609	0.1837	0.1229
RoBERTa-Base [32] ♠	0.5096	0.5000	0.0000	0.0021	0.4165	0.5000	0.0000	0.0052	0.4212	0.5000	0.0000	0.0080
RoBERTa-Large [32] ♠	0.2154	0.5000	0.0000	0.0010	0.3700	0.5000	0.0000	0.0091	0.4054	0.5000	0.0000	0.0307
LRR [46]	0.3225	0.5000	0.0000	0.0124	0.3549	0.5000	0.0000	0.0143	0.3031	0.5000	0.0000	0.0068
DNA-GPT [58] ◇	0.3538	0.5124	0.0716	0.0031	0.3982	0.5019	0.0255	0.0156	0.3457	0.5011	0.0337	0.0102
NPR [46] ◇	0.6025	0.6304	0.3242	0.0093	0.5336	0.5617	0.1718	0.0325	0.4818	0.5392	0.1312	0.0148
DetectGPT [34] ◇	0.6970	0.6781	0.3832	0.0569	0.5918	0.5903	0.2013	0.0325	0.5411	0.5557	0.1456	0.0353
Fast-DetectGPT [7]	0.6280	0.5958	0.1930	0.1159	0.4422	0.5006	0.0255	0.0377	0.4078	0.5000	0.0000	0.0250
ImBD [9] ♠	0.9103	0.8318	0.6658	0.5663	0.7506	0.6981	0.4292	0.3714	0.7357	0.6894	0.3923	0.3265
DetectAnyLLM(ours) ♠	**0.9589**	**0.9105**	**0.8238**	**0.8023**	**0.9478**	**0.8929**	**0.7860**	**0.8091**	**0.9444**	**0.9135**	**0.8309**	**0.8749**
Imp.	+54.24%	+46.77%	+47.28%	+54.42%	+79.07%	+64.52%	+62.50%	+69.63%	+78.97%	+72.16%	+72.17%	+81.42%
MIRAGE-SIG, QwQ-Plus-32B												
Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood [44]	0.4263	0.5098	0.0658	0.0021	0.4292	0.5026	0.0321	0.0166	0.3755	0.5000	0.0000	0.0045
LogRank [25]	0.4024	0.5062	0.0526	0.0010	0.4040	0.5006	0.0253	0.0179	0.3530	0.5000	0.0000	0.0068
Entropy [14]	0.6143	0.5973	0.2357	0.0566	0.5402	0.5320	0.0976	0.0730	0.5835	0.5600	0.1951	0.1459
RoBERTa-Base [32] ♠	0.4835	0.5000	0.0000	0.0000	0.4174	0.5000	0.0000	0.0051	0.4329	0.5000	0.0000	0.0113
RoBERTa-Large [32] ♠	0.2083	0.5000	0.0000	0.0000	0.3718	0.5000	0.0000	0.0371	0.4072	0.5023	0.0476	0.0294
LRR [46]	0.3548	0.5000	0.0000	0.0051	0.3391	0.5000	0.0000	0.0154	0.3005	0.5000	0.0000	0.0057
DNA-GPT [58] ◇	0.3783	0.5113	0.0684	0.0031	0.3868	0.5006	0.0253	0.0115	0.3430	0.5000	0.0000	0.0057
NPR [46] ◇	0.6165	0.6483	0.3357	0.0175	0.5389	0.5621	0.1705	0.0205	0.4724	0.5373	0.1186	0.0158
DetectGPT [34] ◇	0.7068	0.6874	0.4064	0.0319	0.6029	0.6031	0.2159	0.0359	0.5327	0.5486	0.1323	0.0283
Fast-DetectGPT [7]	0.6228	0.5906	0.1863	0.1123	0.4595	0.5000	0.0000	0.0371	0.3927	0.5000	0.0000	0.0170
ImBD [9] ♠	0.9079	0.8357	0.6718	0.5757	0.7660	0.7138	0.4416	0.3880	0.7527	0.7025	0.4140	0.3575
DetectAnyLLM(ours) ♠	**0.9550**	**0.9212**	**0.8429**	**0.8260**	**0.9400**	**0.8899**	**0.7805**	**0.8092**	**0.9256**	**0.8857**	**0.7730**	**0.8077**
Imp.	+51.15%	+52.04%	+52.12%	+58.98%	+74.38%	+61.52%	+60.69%	+68.83%	+69.92%	+61.60%	+61.27%	+70.07%