機械文としての検出されやすさと文章の品質は両立する

齋藤幸史郎 [1] 小池隆斗 [1] 金子正弘 [2,1] 岡崎直観 [1,3,4]
[1] 東京科学大学 [2] MBZUAI [3] 産業技術総合研究所 [4] NII LLMC
{koshiro.saito@nlp., ryuto.koike@nlp., okazaki@}comp.istc.ac.jp
masahiro.kaneko@mbzuai.ac.ae

概要

大規模言語モデル (LLM) の生成文に透かしトークンを埋め込む Watermark は，LLM によって生成された文を高い精度で検出できる．しかし，LLM が計算するトークンの予測確率分布を操作するため，検出性能とタスク性能とのトレードオフが問題視されてきた．本研究では，LLM の被検出性能とタスク性能を同時に最適化する学習フレームワーク D-PUPPET を提案する．評価実験の結果，D-PUPPET を適用した LLM は Long-Form QA，要約，エッセイ生成において被検出性能とタスク性能が共に向上し，その向上はタスクを超えて汎化しうることを確認した．本結果は，被検出性能とタスク性能のトレードオフが十分な最適化によって緩和できることを示唆する．さらに，トレードオフが報告されていた既存の Watermark 手法についても適切な設定下においてはタスク性能を維持可能であることがわかった．

1 はじめに

LLM が生成する文は，人間が読んだだけでは人間が書いた文と見分けが付かないほど高品質なものになりつつある [1]．これを踏まえ，LLM の悪用防止やデータの出自追跡のため，LLM が生成する文を検出することの需要が高まっている．機械文検出は人間が書いた文（人間文）と LLM が生成した文（機械文）を分類するタスクであり，両者の出力分布の違いに基づいて識別を行う [2]．Watermark [3, 4] は，LLM の文生成時にトークンの予測確率分布にバイアスを加え，検出の手掛かりとなる透かしトークンを混ぜることで，高精度な検出を実現する．一方，そうした手法は LLM のトークン出力分布を操作するため，検出性能と生成文の品質（タスク性能）の間でトレードオフが生じる点が問題視されている [5, 6, 7, 8]．しかし，既存手法は検出性能を重視した生成規則に基づいており，両性能を同時に最適化する設計にはなっていない．そのため，報告されているトレードオフが不可避なものかは検証されていない．

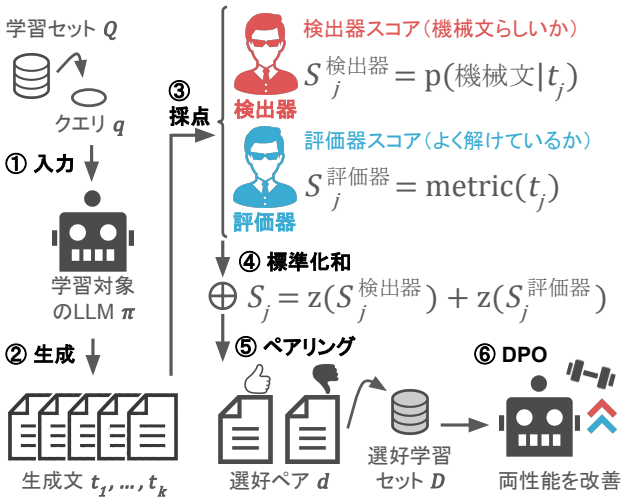

図 1 D-PUPPET の概要．

本研究では，LLM のタスク性能を維持したまま，その生成文を機械文として検出されやすくする学習フレームワーク D-PUPPET を提案する．具体的には，検出器とタスク評価器を用い，生成文に対する機械文クラスの尤度とタスク評価値の二つを報酬として LLM に強化学習を行う．評価実験の結果，D-PUPPET を適用した Llama-3-8B-Instruct [9] と Qwen-3-8B [10] では，Long-form QA，要約，エッセイ生成において，従来問題視されていたトレードオフが緩和されるだけでなく，検出性能とタスク性能の両方が向上した．また，これらの性能向上はタスクを超えて汎化する可能性があることも確認した．さらに，分析では，既存の Watermark 手法 (KGW [3], SynthID [4]) についても，適切な生成設定を選択することで，トレードオフを緩和できることを示した．本結果は，これまで報告されていた Watermark 手法のトレードオフが探索の不十分さに起因していたことを示しており，十分な探索によって高い検出性能とタスク性能が両立する可能性を示唆する．

学習データ	Llama-3-8B-Instruct								Qwen3-8B							
	ELI5		Multi-News		OUTFOX		平均		ELI5		Multi-News		OUTFOX		平均	
	R-L	AUC	R-L	AUC	Jdg.	AUC	Perf.	AUC	R-L	AUC	R-L	AUC	Jdg.	AUC	Perf.	AUC
–	0.229	0.937	0.265	0.928	3.970	0.947	0.429	0.937	0.245	0.937	0.240	0.833	**4.360**	0.802	0.452	0.857
ELI5	**0.258**†	**0.999**	0.276†	0.971	**4.325**†	0.897	**0.466**	0.956	**0.266**†	**0.999**	0.246†	0.912	4.298	0.897	0.457	0.936
Multi-News	0.245†	0.990	**0.281**†	**0.990**	4.253†	0.968	0.459	0.983	0.254†	0.990	**0.267**†	**0.964**	4.305	**0.968**	**0.461**	**0.974**
OUTFOX	0.240†	**0.999**	0.267	0.977	4.030	**0.999**	0.438	**0.992**	0.252†	0.987	0.244	0.941	4.273‡	0.971	0.450	0.966

表 1　D-PUPPET 適用前後での検出性能とタスク性能の比較．R-L：ROUGE-L，AUC：AUROC，Jdg.：LLM-as-a-Judge，Perf.：タスク性能．**太字**：各モデル・各指標の最良値．平均算出時には各指標を 0–1 に正規化．†，‡：D-PUPPET 適用前（–）に比べて優位に改善／劣化（平均以外のタスク性能について $\alpha = 0.05$ の Paired T 検定）．

2　関連研究

機械文検出には，1. 生成文に対する検出器を開発する，2. 文生成時に生成文に透かしトークンを埋め込む，という二つの大きな方向性が存在する．

生成文に対する検出器の開発　この方向性で最も一般的なアプローチでは，人間文と機械文のラベル付きデータを収集し，教師あり分類器を学習する．ベースモデルとしては RoBERTa [11] が採用されることが多く [12, 13, 14, 15]，特に OpenAI が開発した RoBERTa ベースの OpenAI Detector [12] は複数の先行研究 [16, 17] で高い検出性能が報告されている．

生成文への透かしトークンの埋め込み　この方向性には，LLM の生成分布に操作を加える Logit ベース手法と，サンプリングで選ばれるトークンを操作する Sampling ベース手法が存在する．

Logit ベース手法で代表的なものが Kirchenbauer-Geiping-Wen Watermark (KGW) [3] である．KGW ではある位置のトークンを生成する際に，まず，その位置までのトークンの情報から，モデルの語彙のうち γ の割合だけを透かしトークンにランダムに割り当てる．そして，透かしトークンの尤度を δ だけ上昇させることにより，生成文中に透かしトークンを γ の割合ほど出現させる．これにより，KGW が適用された生成文には γ 程度の割合で透かしトークンが存在すると期待できるので，対象文の透かしトークン率との差分から検出を行うことが可能になる．

Sampling ベース手法の一つに Google が考案した SynthID [4] が存在する．SynthID ではある位置のトークンを生成する際に，まず，N^M の候補を生成する．次にそれぞれの候補に対して M 個のランダムなスコアを付与する．そして候補を N 個ごとの組に分け，スコアが最大の候補を勝ち抜けさせるようなトーナメントを M 回行い，最後の一つを透かしトークンとして採用する．これにより，SynthID が適用された生成文は高いスコアを持っていると期待でき，対象文のスコアから検出が可能になる．

Logit ベース手法はトークンの出力分布を直接操作するため堅牢な検出性能が期待できるが，文の品質が劣化しうる．一方，Sampling ベース手法は元の出力分布を壊しにくいため反対の性質を持つ．

先行研究との差分　先行研究である PUPPET [18] も，検出器の尤度とタスク評価値を報酬として LLM に強化学習を行うことで被検出性能とタスク性能の両立を図っている．これに対し，本研究ではより幅広いモデル・データセットを対象とし，In-domain に加えて Out-of-domain での有効性の検証を行っている．また，PUPPET が高い被検出性能とタスク性能の維持を目標としていたことに対し，本研究では両者のトレードオフを緩和することを目標とし，より高い被検出性能に加え，タスク性能の改善を実現している．さらに，既存の Watermark 手法に対するトレードオフの緩和についても知見を得ている．

3　提案手法：D-PUPPET

図 1 に提案手法 D-PUPPET の概要図を示す．まず入力クエリ q ($\in Q$) に対して，学習前の学習対象モデル π から k 個の生成文をサンプリングする．

$$\{t_1, \ldots, t_k\} \sim \pi(\cdot \mid q).$$

次に各生成文 t_j ($j \in \{1, ..., k\}$) について二つのスコアを計算する．一つ目のスコアは，各生成文に対して検出器を適用して得ることができる機械文クラスの確率である．

$$S_j^{\text{検出器}} = S_{\text{検出器}}(\text{機械文} \mid t_j).$$

二つ目のスコアは，各生成文に対して評価器を適用して得ることができるタスク評価値である．

$$S_j^{\text{評価器}} = \text{metric}(t_j).$$

ここで metric(·) には，入力クエリのタスクに応じた評価指標（例：ROUGE-L，LLM-as-a-Judge）が使用

される．最後に，これら二つのスコアをそれぞれ k 個内で標準化 $z(\cdot)$ したのちに足し合わせたものを生成文 t_j に対する最終的なスコアとする．

$$S_j = z\left(S_j^{\text{検出器}}\right) + z\left(S_j^{\text{評価器}}\right).$$

こうして得られたスコア S_j に基づいて，クエリ q に対する選好ペア $d = (q, t_{j^+}, t_{j^-})$ を構築する．ここのペアリングは様々な設計が可能であるが，本研究では最大の報酬を持つ生成文を正例，最小の報酬を持つ生成文を負例とした．

$$j^+ = \underset{j\in\{1,\ldots,k\}}{\operatorname{argmax}}\ S_j, \quad j^- = \underset{j\in\{1,\ldots,k\}}{\operatorname{argmin}}\ S_j.$$

D-PUPPET では，以上のように構築した選好学習セット $D := \{d \mid q \in Q\}$ を用いて，モデル π に対して DPO (Direct Preference Optimization [19]) を行い，検出性能とタスク性能を同時に最適化する．

4 実験設定

本章では，D-PUPPET の有効性検証に用いた「データセット」「学習対象 LLM」「検出器」「評価器」を紹介する．さらに詳細な学習や評価の設定については付録 A を参照されたい．

データセット Q データセットには次の 3 つを採用した． 1. ELI5 [20] (Long-Form QA)，2. Multi-News [21]（複数記事要約），3. OUTFOX [22]（エッセイ生成）．全てのデータセットについて，学習セットから 5,000 件，評価セットから 200 件を乱択し，それぞれ学習と評価に用いた．

学習対象 LLM π 学習対象の LLM による影響を調べるため，モデルのアーキテクチャや学習データの構成などといった設計思想の異なる二つの LLM，Llama-3-8B-Instruct と Qwen-3-8B を採用した．

検出器 先行研究 [16, 17] で高い検出性能が報告されている OpenAI Detector [12] を採用する．検出性能の評価指標としては AUROC を用いる．

評価器 タスク評価値の特徴による影響を調べるため，ELI5 と Multi-News については記号的情報により計算される ROUGE-L ([0, 1])，OUTFOX については意味的情報が考慮される LLM-as-a-Judge ([0, 5], 0.5 刻み）を用いる．

5 結果

表 1 に，D-PUPPET 適用前後の LLM の被検出性能およびタスク性能の変化を示した．この結果から次の二つのことがわかる．

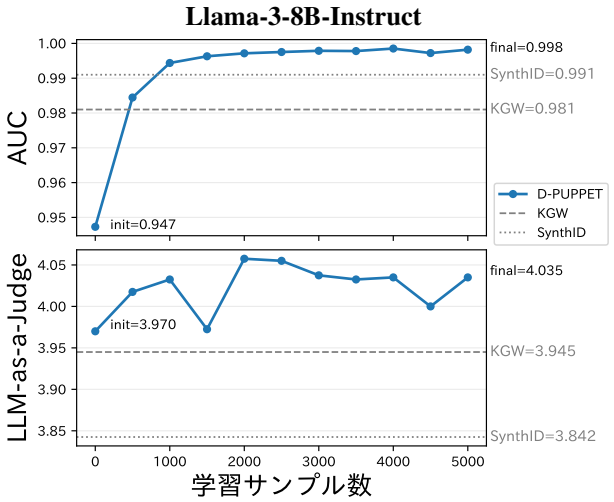

図 2 OUTFOX データセットにおける D-PUPPET を用いた学習時の検出性能とタスク性能の推移．

まず，学習したタスクと同じタスクで評価をした結果から，検出性能とタスク性能の両方が概ね全てのケースで改善しており，学習前に比べてそれぞれ最大で 21%，13% の向上が確認できた．本結果は D-PUPPET が先行研究で問題視していた両性能間のトレードオフを効果的に緩和したことを表す．

次に，学習したタスクと異なるタスクで評価をした結果から，学習していないタスクについても両性能の向上が見られ，学習前に比べてそれぞれ最大で 21%，9% の向上が確認できた．本結果は D-PUPPET がタスク間で汎化する可能性を示唆する．

6 分析

6.1 D-PUPPET に必要な学習サンプル数

本分析では，D-PUPPET の実用性を検証することを目的として，本手法で LLM の被検出性能とタスク性能を十分に高めるために必要な学習サンプル数を調査する．

図 2 に，OUTFOX データセットを対象に Llama-3-8B-Instruct を D-PUPPET で学習させたときの，500 サンプルごとの両性能の推移を示す．まず被検出性能の推移は，1,000 サンプルで既存の Watermark 手法の検出性能をいずれも超え，2,000 サンプルあたりで値が飽和していることがわかる．次にタスク性能の推移は，未学習の点から緩やかに上昇をしており，特に 1,000 サンプル目以降は伸びは僅かであることがわかる．

この観察から，D-PUPPET の学習には多くとも 2,000 件程度の学習サンプルがあれば被検出性能と

		Llama-3-8B-Instruct								Qwen3-8B							
		ELI5		Multi-News		OUTFOX		平均		ELI5		Multi-News		OUTFOX		平均	
Watermark	CT	R-L	AUC	R-L	AUC	Jdg.	AUC	Perf.	AUC	R-L	AUC	R-L	AUC	Jdg.	AUC	Perf.	AUC
–	w/	0.229	–	0.265	–	3.970	–	0.429	–	0.245	–	0.240	–	4.360	–	0.452	–
KGW	w/o	0.235†	**0.998**	0.251‡	**0.995**	3.685‡	0.989	0.408	**0.994**	0.248	**0.995**	0.230‡	0.997	3.788‡	**1.000**	0.412	**0.997**
SynthID	w/o	**0.239**†	0.975	0.260	0.984	3.695‡	**0.997**	0.410	0.985	**0.251**†	0.948	0.226‡	0.976	3.928‡	0.996	0.424	0.973
KGW	w/	0.233†	0.995	**0.268**	0.972	**3.945**	0.981	**0.430**	0.983	0.245	**0.995**	0.239	**0.998**	**4.373**	0.988	**0.453**	0.994
SynthID	w/	0.235†	0.948	0.265	0.965	3.843‡	0.991	0.423	0.968	0.246	0.945	**0.242**	0.980	4.340	0.976	0.452	0.967

表 2　チャットテンプレート (CT) の有無ごとの Watermark 適用による性能変化．R-L：ROUGE-L，AUC：AUROC，Jdg.：LLM-as-a-Judge，Perf.：タスク性能．**太字**：各モデル・各指標の最良値．平均算出時には各指標を 0–1 に正規化．†，‡：Watermark なしの設定（–）に比べた優位な改善／劣化（平均以外のタスク性能について $\alpha = 0.05$ の Paired T 検定を実施）．

タスク性能を十分に向上できることが示唆された．一般的に DPO が数十万件オーダーの学習サンプルを要する [23, 24] ことを考慮すると，2,000 件は非常に小さい規模であり，D-PUPPET が実用的な手法であることが示唆された．

6.2　Watermark のトレードオフ

Watermark を適用すると文の品質が低下する，というトレードオフを主張する既存研究では，多くの場合，その品質を測る指標として尤度のみが用いられている [3, 6, 4]．さらに，ROUGE-L などのタスク指標を用いている場合であっても，評価対象のモデルが古い（例：T5 [25]，Llama 2 [26]），あるいは事後学習済みモデルにチャットテンプレートが付与されず，学習時と異なる設定で評価されている [8, 5, 7]．

そこで本分析では，比較的新しい Llama-3-8B-Instruct および Qwen3-8B を対象に，チャットテンプレートを正しく付与した設定で，トレードオフが存在すると報告されている Watermark 手法を再検証する．調査対象の Watermark 手法としては，検出アルゴリズムの異なる 2 手法，KGW と SynthID，を採用した．なお，Watermark の実装には MarkLLM [27] を利用し，チャットテンプレートの適用に関する部分のみを改変した．

表 2 に Watermark を適用した時の検出性能とタスク性能の変化を，チャットテンプレートの有無ごとに示した．まず，既存研究で使われていた チャットテンプレートを付与しない設定に着目すると．検出性能は 0.948 から最大で 1.000 と非常に高いことが分かる．一方，タスク性能は Watermark を適用していない時と比べてほとんどのケースで優位に劣化していることが分かり，最大で 13% 劣化している．ここで，チャットテンプレートを正しく付与した設定に着目する．平均の検出性能は チャットテンプレートを付与していないときにわずかに劣る．しかし，タスク性能は Watermark を適用していない時に比べて優位な劣化が概して見られないことがわかる．

以上から，従来の研究で問題視されていた KGW と SynthID における検出性能とタスク性能のトレードオフは，最新のモデルにチャットテンプレートを正しく付与させた上で生成させることによって効果的に緩和できることがわかった．ここで，D-PUPPET はタスク性能を学習の対象に含めていることから，トレードオフが緩和された KGW と SynthID と比べても平均タスク性能が高いことは留意されたい．

7　結論

本研究では，検出性能とタスク性能を同時に最適化するような探索を行うことにより，既存研究で問題視されていた両者間のトレードオフを解消することを目標とし，検出器報酬と評価器報酬を用いた学習フレームワーク D-PUPPET の有用性を検証した．評価実験から，D-PUPPET を適用することによって，品質を保持させたまま機械文として検出されやすい生成文を書くように，LLM を学習できることを確認した．さらに，この学習の効果がタスクを超えて汎化する可能性があることを確かめた．また，検出性能とタスク性能の間でトレードオフがあると報告されていた既存の Watermark 手法について，チャットテンプレートが正しく付与された最新のモデルについてはタスク性能が劣化しないことを発見した．これらの知見はこれまでトレードオフの関係にあるとされてきた検出性能とタスク性能を同時に最適化できる可能性を示唆するものであり，実用的な機械文検出の開発に大きく寄与すると期待している．

謝辞

本研究は，JST 経済安全保障重要技術育成プログラム JPMJKP24C3 の支援を受けたものです．

参考文献

[1] Yuxia Wang, Rui Xing, Jonibek Mansurov, et al. Is human-like text liked by humans? multilingual human detection and preference against AI. arXiv:2502.11614, 2025.

[2] Ryuto Koike, Liam Dugan, Masahiro Kaneko, et al. Machine text detectors are membership inference attacks. arXiv:2510.19492, 2025.

[3] Yuxin Wen John Kirchenbauer, Jonas Geiping, et al. A watermark for large language models, 2023.

[4] Sumanth Dathathri, Abigail See, Sumedh Ghaisas, et al. Scalable watermarking for identifying large language model outputs. **Nature**, Vol. 634, No. 8035, pp. 818–823, 2024.

[5] Anirudh Ajith, Sameer Singh, and Danish Pruthi. Downstream trade-offs of a family of text watermarks. In **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 14039–14053, 2024.

[6] Yu Fu, Deyi Xiong, and Yue Dong. Watermarking conditional text generation for AI detection: Unveiling challenges and a semantic-aware watermark remedy. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 38, No. 16, pp. 18003–18011, 2024.

[7] Yidan Wang, Yubing Ren, Yanan Cao, and et al. From trade-off to synergy: A versatile symbiotic watermarking framework for large language models. In **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 10306–10322, 2025.

[8] Pierre Fernandez, Antoine Chaffin, Karim Tit, et al. Three bricks to consolidate watermarks for large language models. In **2023 IEEE International Workshop on Information Forensics and Security (WIFS)**, 2023.

[9] AI@Meta. Llama 3 model card, 2024.

[10] Qwen Team. Qwen3 technical report. arXiv:2505.09388, 2025.

[11] Yinhan Liu, Myle Ott, Naman Goyal, and et al. Roberta: A robustly optimized BERT pretraining approach. arXiv:1907.11692, 2019.

[12] Irene Solaiman, Miles Brundage, Jack Clark, et al. Release strategies and the social impacts of language models. arXiv:1908.09203, 2019.

[13] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, and et al. SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In **Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)**, pp. 2057–2079, 2024.

[14] Eduard Tulchinskii, Kristian Kuznetsov, Kushnareva Laida, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. Intrinsic dimension estimation for robust detection of AI-generated texts. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[15] Yafu Li, Qintong Li, Leyang Cui, and et al. 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)**, pp. 36–53, 2024.

[16] Charlotte Nicks, Eric Mitchell, Rafael Rafailov, et al. Language model detectors are easily optimized against. In **The Twelfth International Conference on Learning Representations**, 2024.

[17] Laida Kushnareva, Tatiana Gaintseva, Dmitry Abulkhanov, et al. Boundary detection in mixed AI-human texts. In **First Conference on Language Modeling**, 2024.

[18] 齋藤幸史郎, 小池隆斗, 金子正弘, 直観. PUPPET：タスク性能を維持しながら LLM として検出されやすくする学習フレームワーク. 言語処理学会第 31 回年次大会, pp. 2791–2796, 2025.

[19] Rafael Rafailov, Archit Sharma, Eric Mitchell, et al. Direct preference optimization: Your language model is secretly a reward model. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[20] Angela Fan, Yacine Jernite, Ethan Perez, and et al. ELI5: Long form question answering. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 3558–3567, 2019.

[21] Alexander Fabbri, Irene Li, Tianwei She, and et al. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 1074–1084, 2019.

[22] Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. OUTFOX: LLM-generated essay detection through in-context learning with adversarially generated examples. In **Proceedings of the 38th AAAI Conference on Artificial Intelligence**, pp. 21258–21266, 2024.

[23] Team Olmo, Allyson Ettinger, Amanda Bertsch, Amanda Bertsch, et al. Olmo 3, 2025.

[24] Chunting Zhou, Pengfei Liu, and Puxin Xu et al. LIMA: Less is more for alignment. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[25] Colin Raffel, Noam Shazeer, Adam Roberts, and et al. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv:1910.10683, 2019.

[26] Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288, 2023.

[27] Leyi Pan, Aiwei Liu, Zhiwei He, and et al. MarkLLM: An open-source toolkit for LLM watermarking. In **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 61–71, 2024.

[28] Leandro von Werra, Younes Belkada, Lewis Tunstall, et al. TRL: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

[29] Biyang Guo, Xin Zhang, Ziyuan Wang, and et al. How close is ChatGPT to human experts? comparison corpus, evaluation, and detection. arXiv:2301.07597, 2023.

[30] Shangqing Tu, Yuliang Sun, Yushi Bai, and et al. WaterBench: Towards holistic evaluation of watermarks for large language models. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1517–1542, 2024.

[31] Jifan Yu, Xiaozhi Wang, Shangqing Tu, et al. KoLA: Carefully benchmarking world knowledge of large language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[32] Hao Peng, Xiaozhi Wang, Shengding Hu, and et al. COPEN: Probing conceptual knowledge in pre-trained language models. In **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 5015–5035, 2022.

[33] Zhilin Yang, Peng Qi, Saizheng Zhang, and et al. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 2369–2380, 2018.

[34] Mark Chen, Jerry Tworek, Heewoo Jun, and et al. Evaluating large language models trained on code. arXiv:2107.0337, 2021.

[35] Yann Dubois, Xuechen Li, Rohan Taori, et al. AlpacaFarm: A simulation framework for methods that learn from human feedback. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[36] Macedo Maia, Siegfried Handschuh, André Freitas, et al. WWW'18 Open Challenge: Financial opinion mining and question answering. In **Companion Proceedings of the The Web Conference 2018**, p. 1941–1942, 2018.

[37] Ming Zhong, Da Yin, Tao Yu, and et al. QMSum: A new benchmark for query-based multi-domain meeting summarization. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 5905–5921, 2021.

[38] OpenAI. gpt-oss-120b & gpt-oss-20b model card. arXiv:2508.10925, 2025.

[39] Yupei Wang, Renfen Hu, and Zhe Zhao. Beyond agreement: Diagnosing the rationale alignment of automated essay scoring methods based on linguistically-informed counterfactuals. In **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 8906–8925, 2024.

[40] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.

パラメータ	値
learning_rate	1×10^{-4}
num_train_epochs	1
per_device_train_batch_size	16
data_seed	42
max_length	ELI5:800, Multi-News:8192, OUTFOX:8192
max_completion_length	ELI5:300, Multi-News:512, OUTFOX:512

表 3　デフォルト値から変更した主なパラメータ一覧．

	訓練（5,000 件）	評価（200 件）
ELI5	Hello-SimpleAI/HC3, reddit_eli5 [29]	THU-KEG/WaterBench, 2-1_longform_qa [30]
Multi-News	alexfabbri-/multi_news [21]	THU-KEG/WaterBench, 4-1_multi_news [30]
OUTFOX	ryuryukke-/OUTFOX [22]	ryuryukke-/OUTFOX [22]

表 4　データの出典．（訓練は train，評価は test から乱択．）

A　実験設定の詳細

A.1　生成設定

本研究の洞察を明瞭にするため，すべての実験について Qwen3-8B の推論機能 (thinking mode) は無効化した．

A.2　学習の実装

学習には TRL[28] の DPOTrainer を用いた．用いたパラメータについては表 3 を参考にされたい．また計算資源を考慮して LoRA を採用している．

A.3　学習・評価のデータセット

ELI5 と Multi-News については WaterBench [30] に収録されている 200 件 のサンプルをそのまま評価用に利用した．訓練用のデータは同フレームワークで用いられていたデータの入手先と前処理を参考に作成した．また評価指標である ROUGE-L の実装についても同フレームワークの計算方法を模倣した．なお，この WaterBench には ELI5 と Multi-News の他に KoLA [31]，Copen [32]，HotpotQA [33]，LCC [34] といった一問一答やコード生成のデータセットが含まれている．しかし，それらのタスクは想定される回答の多様性が小さい（例：コードスニペットの次行予測）ことや，想定される回答の文長が極端に短い（例：“True”，“1”，“A.”）ことが考えられたため，検出の必要性が大きくないと考え，除いた．また，AlpacaFarm [35]（指示追従）については検出の必要性こそあるものの人間文が提供されていないため断念した．そして，FiQA [36]（金融に関する Long-Form QA タスク）と QMsum [37]（query ベースの要約タスク）については同様のタスクのデータセットである ELI5 と Multi-News を採用しているため除いた．一方，OUTFOX はエッセイ生成タスクであり，ELI5 や Multi-News のタスクと異なる上，検出の必要性も高いと判断したため採用した．OUTFOX は評価指標を提供していないデータセットであったため，性能と計算資源を考慮して gpt-oss-20b [38] を用いた LLM-as-a-Judge を採用した．プロンプトについてはエッセイを自動採点する先行研究 [39] で使用されていたものを placeholder {} の部分のみ改変して利用した（図 3）．なお，全てのデータセットについて 1 サンプルあたりの生成数は 1 サンプルあたりに提供されている人間文の数（ELI5：3 件，Multi-News：1 件，OUTFOX：1 件）に合わせている．また選考ペアの候補数 k は先行研究 [40] に倣って 5 個とした．

```
Read and evaluate the essay written in response to the prompt: "{context}"

Essay:
"{prediction}"

Please assign it a score from 1 to 5 (in increments of 0.5 points) based on rubric below:
- A 5-point essay [...].
- A 4-point essay [...].
- A 3-point essay [...].
- A 2-point essay [...].
- A 1-point essay [...].

Your response should be a JSON object containing only one key: 'score', which should be a numeric value representing the score you gave.
```

図 3　OUTFOX の LLM-as-a-Judge に用いたプロンプト．

	ELI5		Multi -News		OUT -FOX		Avg.	
Watermark	R-L	AUC	R-L	AUC	Jdg.	AUC	Perf.	AUC
Llama-3-8B-Instruct								
KGW ($\delta = 2$)	0.233	0.995	**0.268**	0.972	3.945	0.981	**0.430**	0.983
KGW ($\delta = 5$)	0.223‡	**1.000**	0.253‡	**1.000**	**3.995**	**1.000**	0.425	**1.000**
KGW ($\delta = 10$)	0.207‡	**1.000**	0.221‡	**1.000**	2.748‡	**1.000**	0.326	**1.000**
SynthID ($N = 2$)	**0.235**	0.948	0.265	0.965	3.843	0.991	0.423	0.968
SynthID ($N = 3$)	0.233	0.989	0.266	0.992	3.893	**1.000**	0.426	0.994
SynthID ($N = 5$)	0.215‡	0.998	0.231‡	**1.000**	3.198‡	**1.000**	0.362	0.999
Qwen3-8B								
KGW ($\delta = 2$)	0.245	0.995	0.239	0.998	4.373	0.988	0.453	0.994
KGW ($\delta = 5$)	0.241‡	**1.000**	0.230‡	**1.000**	4.350	**1.000**	0.447	**1.000**
KGW ($\delta = 10$)	0.231‡	**1.000**	0.207‡	**1.000**	3.823‡	**1.000**	0.401	**1.000**
SynthID ($N = 2$)	**0.246**	0.945	**0.242**	0.980	4.340	0.976	0.452	0.967
SynthID ($N = 3$)	0.244	0.981	0.241	0.996	**4.383**	0.997	**0.454**	0.991
SynthID ($N = 5$)	0.242‡	0.996	0.231‡	**1.000**	4.340	**0.999**	0.447	0.998

表 5　異なる透かし強度における性能比較．R-L：ROUGE-L，AUC：AUROC，Jdg.：LLM-as-a-Judge，Perf.：タスク性能．**太字**：各モデル・各指標の最良値．平均算出時には各指標を 0–1 に正規化．†，‡：デフォルトの強度に比べた優位な改善／劣化（平均以外のタスク性能について $\alpha = 0.05$ の Paired T 検定を実施）．

A.4　Watermark のパラメータ

付録 B を除き，KGW と SynthID に関する全てのパラメータは MarkLLM のデフォルト値を採用している．

B　透かし強度とトレードオフ

Watermark には透かしの強度を主に司るパラメータが存在する．KGW なら Green Tokens に対する尤度の上昇幅である δ，SynthID なら 1 トーナメントあたりのサンプル数 N が相当する．なお，詳細な説明は §2 を参考されたい．ここではそれらの透かし強度について MarkLLM のデフォルト値：$\delta = 2; N = 2$ よりも強い設定：$\delta = 5, 10; N = 3, 5$ にした場合に性能がどのように変化するかを調査する．

結果（表 5）を見ると，透かし強度を強くした場合，いずれの手法でも検出性能は 1.000 かその近傍の値まで向上した．一方でタスク性能はデフォルトの透かし強度の設定に比べて多くのケースで優位な劣化が確認され，KGW で最大 30%，SynthID で最大 18%も劣化することが分かった．KGW は Logit ベースであるためか劣化幅が大きい．

以上から，課題の採点や書類の審査などの High-Stakes な設定で Watermark を適用した LLM を使用する際には，より精緻なパラメータ探索が必要であると考えられる．