Language Model Detectors Are Easily Optimized Against

Anonymous authors
Paper under double-blind review

Abstract

The fluency and general applicability of large language models (LLMs) has motivated significant interest in detecting whether a piece of text was written by a language model. While both academic and commercial detectors have been deployed in some settings, particularly education, other research has highlighted the fragility of these systems. In this paper, we demonstrate a data-efficient attack that fine-tunes language models to confuse existing detectors, leveraging recent developments in reinforcement learning of language models. We use the 'human-ness' score (often just a log probability) of various open-source and commercial detectors as a reward function for reinforcement learning, subject to a KL-divergence constraint that the resulting model does not differ significantly from the original. For a 7B parameter Llama-2 model, fine-tuning for under a day reduces the AUROC of the OpenAI RoBERTa-Large detector from 0.84 to 0.62, while perplexity on OpenWebText increases from 8.7 to only 9.0; with a larger perplexity budget, we reduce AUROC to 0.30 (worse than random), with a perplexity increase to 9.9. Similar to traditional adversarial attacks, we find that this increase in 'detector evasion' generalizes to other detectors not used during training. In light of our empirical results, we advise against continued reliance on LLM-generated text detectors.

1 Introduction

Large language models (LLMs) can produce high-quality text in a wide variety of settings (3; 4). Access to such powerful LLMs has expanded rapidly; for anyone hoping to generate machine-written text, a plethora of free and low-cost options exist. The usage of such models has become endemic to classrooms, news outlets, social media platforms, and other domains. This rapid development has led to several objections to widespread use of LLMs, including moral qualms with data procurement, applications, questions regarding the quality of machine-generated text and issues with LLMs outputting inaccurate information (hallucinations) among others. These concerns have led to a significant amount of research and commercial product offerings for detecting machine-generated text (e.g. 6; 19; 15). In this work we present an adversarial attack that utilizes reinforcement learning (RL) that directly optimizes an LLM to minimize it's detectability. We seek to answer the following questions: How does detectability trade-off with other metrics like perplexity? How does detectability scale with the query budget to a detector, and is it feasible to train available LLMs to be undetectable against commercial detectors on a limited budget? Does training against one detector reduce detectability under other detectors? We should note that in contrast to prior work, our approach does not require the use of human paraphrasers or a paraphrasing model (11) and adds no overheard during inference.

The main contribution of this paper is a detailed empirical study on the ease of evading language model detectors by optimizing against them. Our experiments find that a simple DPO-based pipeline produces consistent reduction in detectability against various detectors. We can achieve AUROC metrics below 0.5 against several strong public and commercial detectors, indicating worse than random chance detector performance on the fine-tuned model and close to random chance performance on several additional detection algorithms at the cost of only small increases in perplexity. Moreover, in many cases optimizing against one detector yields a model that is also less detectable under other detectors. In particular, we find that models pre-trained against the public **RoBERTa-large** achieve an average of 0.15 reduction in AUROC when evaluated by a number of black-box commercial detectors. These results hold even at longer sequences, such as generating essays, where our fine-tuned Llama-7b-chat model achieves a RoBERTA-large AUROC of 0.26.

The results of our red-teaming effort suggest that fine-tuning language models to be less detectable is both easy and cheap, which makes it feasible for a wide slate of malicious actors, even against the best open-source and commercial detectors available. Based on these results, we advise various stakeholders (educators, policymakers etc) against reliance on the current suite of text-detectors, and to suitably account for less-detectable LLM generated text.

2 Optimizing Against Language Model Detectors

We leverage recent advantages in fine-tuning language models with reinforcement learning to directly optimize for maximum detector confusion. We present our pipeline below.

Reinforcement Learning for Language Modelling. We consider a language model π_θ that is conditioned on a prompt x and auto-regressively generates an output sequence y. The desired objective is expressed through a reward function $r(x, y)$ that assigns higher rewards to more desirable responses. In practice, the most commonly-used objective includes an additional KL-divergence penalty between the language model and its initialization:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}_p, y\sim\pi_\theta(y|x)}\left[r(x, y) - \beta\mathbb{D}_{\text{KL}}\left[\pi_\theta(y \mid x) \mid\mid \pi_{\text{ref}}(y \mid x)\right]\right] \tag{1}$$

where $\mathcal{D}_p$ is some dataset of prompts π_{ref} is the reference model and β is a coefficient that controls the trade-off between reward and divergence (16; 1; 20). The objective above seeks to align the model with the reward function, while not deviating too far from the pre-trained model.

Direct Preference Optimization (DPO). Rafailov et al. (17) recently proposed the DPO algorithm with the goal of enabling simpler, stabler optimization of the above KL-constrained objective in the case where the reward function is *learned* from a dataset of preference pairs. Assume a dataset of preference pairs $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^N$ of prompts x and two generations $y_w \succ y_l$, where y_w is preferred over y_l. Under suitable assumptions, the DPO algorithm (17) shows that the exact optimal policy π^* for the problem in Eq. 1 can be directly optimized through the MLE objective:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right] \tag{2}$$

This supervised loss results in a simple and stable training objective, so we adopt DPO as our optimization algorithm. See Appendix A for a complete discussion of related work.

Preference Data Generation and Optimization for Detector Evasion. To apply DPO to our detector-evasion setting, we construct a preference dataset using Llama-2-7B samples. We generate a pair of samples $y^{(i)}, \bar{y}^{(i)}$ for each prompt $x^{(i)}$ in the dataset, using temperature 1.0.[1] Preference labels are generated by comparing the detector's 'human-ness' score $s(x, y)$ for a pair of responses, assigning the label $y^{(i)} \succ \bar{y}^{(i)}$ if $s(x, y^{(i)}) > s(x, \bar{y}^{(i)})$; otherwise we have $\bar{y}^{(i)} \succ y^{(i)}$. Once we have generated the preference dataset, we fine-tune Llama-2-7B using the DPO objective in Eq. 2.

3 Experiments

We conduct a wide variety of experimental evaluations in order to understand the extent to which optimizing against detectors is feasible and cost-effective. In Section 3.1, we investigate *the extent to which training against one detector provides evasion from other detectors*, using both open-source and commercial detectors available only through APIs. Section 3.2 studies how many queries to a detector are necessary to collect a dataset sufficient for evasion. Section 3.3 evaluates whether sampling longer sequences from the evasion-tuned model degrades evasion. Finally, we explore detector evasion in language models fine-tuned for dialogue in Section 3.4, which explore evasion using off-policy data in a case study in essay generation.

[1] The number of prompts varies across experiments, but is typically on the order of 10k; the specific value is noted in the relevant experimental sections.

		Detector Trained Against						
		None	**RoB-lg**	**RoB-base**	**Log Prob**	**Log Rank**	**DetectGPT**	**DetectLLM**
	Perplexity	8.7	9.0	8.9	9.0	9.7	9.5	9.5
Eval Detector	**RoB-lg**	0.84	0.62	0.68	0.87	0.90	0.87	0.88
	RoB-base	0.78	0.56	0.53	0.78	0.81	0.78	0.80
	Log Prob	0.69	0.59	0.58	0.32	0.14	0.50	0.55
	Log Rank	0.75	0.64	0.64	0.42	0.22	0.57	0.61
	DetectGPT	0.81	0.82	0.80	0.70	0.62	0.48	0.53
	DetectLLM	0.82	0.83	0.83	0.77	0.73	0.56	0.60
	Median AUROC	0.80	0.63	0.66	0.74	0.68	0.57	0.61

Table 1: **Cross-detector generalization for open-source detectors.** For each detector, we train **3 models** with different KL constraints ($\beta \in \{0.05, 0.5, 5\}$). We report the perplexity of the strongest resulting model **within a perplexity budget of 9.7** (an increase of 1) on OpenWebText, as well as the AUROC for all detectors.

Models. Our experiments use the open-source 7B parameter Llama-2-base model (23) in Sections 3.1-3.3 and the corresponding Llama-2-chat model in Section 3.4. For the off-policy data experiment in Section 3.4, we sample demonstration responses from `gpt-3.5-turbo-0613`.[2]

Datasets. In Sections 3.1-3.4, we generate texts for detection that continue short 2 to 16-token prefixes of OpenWebText documents (7). For the experiments with chat-tuned models in Section 3.4's essay-generating case study, we use prompts from the Alpaca instruction dataset (22) for generic evasion tuning and a collection of free, human-written essays for essay-specific evasion-tuning (25).

Detectors. We use strong open-source detectors from prior work, including language models trained for detection such as the **RoBERTa-large** and **RoBERTa-base** models trained in Solaiman et al. (19) as well as 'zero-shot' detectors (15) such as **Log Rank** and **Log Probability** thresholding. Finally, we include zero-shot perturbation-based methods **DetectGPT** (15) and **DetectLLM** (21). For zero-shot detectors, we make the assumption that the detector knows that samples are being generated by a Llama-2 model; that is, for these detectors, we use Llama-2-base (the model we fine-tune) to compute log probabilities and ranks. This configuration represents an optimistic case for the detector. In addition to open-source detectors, we train against four popular commercial detectors, **GPTZero**, **Sapling**, **Originality.ai**, and **Winston AI**.[3] All commercial detectors advertise strong performance against widely used LLMs.

3.1 Evaluating Generalization of Detector Evasion Across Detectors

In our first experiment, we study the basic question of the feasibility of optimizing language models against language model detectors without significantly harming sample quality. We fine-tune three Llama-2 7B models on preferences computed from a variety of open source (Table 1) and commercial (Table 2) detectors.

We find that optimizing against both open source and commercial detectors is effective, driving the AUROC of the detector trained against to 0.62 or below in every case. In addition, we find strong generalization between detectors when training against the OpenAI RoBERTa-large detector, DetectGPT and Originality. Performance is, intuitively, correlated for open-source detectors with similar mechanisms: RoB-lg and RoB-base are both fine-tuned RoBERTa models; log prob and log rank but threshold token-level statistics of the observed text; DetectGPT and DetectLLM both measure a discrepancy that appears after perturbing the given text. Similary, among the commercial detectors, we note similar performance from the RoBERTa and Originality detectors; this result makes sense because Originality also uses a fine-tuned RoBERTa-like model for detection.[4] However, we also observe asymmetry in generalization between evasion-tuned models; the model trained against RoBERTa-large also evades log probability and log rank detectors, but not vice versa. Among commercial detectors, Originality proves most resistant to evasion-tuning on other detectors. However, optimizing against Originality itself produces relatively low AUROC for it as well as the

[2] https://platform.openai.com/docs/models/gpt-3-5.

[3] https://gptzero.me/; https://sapling.ai/; https://originality.ai/; https://gowinston.ai/

[4] https://originality.ai/blog/ai-content-detection-accuracy

		Detector Trained Against					
		None	**RoB-lg**	**GPTZero**	**Winston**	**Originality**	**Sapling**
	Perplexity	8.7	9.0	9.5	9.6	9.7	9.3
Eval Detector	**RoB-lg**	0.84	0.62	0.79	0.83	0.64	0.76
	GPTZero	0.57	0.46	0.16	0.32	0.42	0.42
	Winston	0.53	0.45	0.30	0.21	0.41	0.43
	Originality	0.99	0.68	0.77	0.98	0.62	0.78
	Sapling	0.75	0.64	0.67	0.72	0.64	0.60
	Median AUROC	0.75	0.62	0.67	0.72	0.62	0.60

Table 2: **Cross-detector generalization for commercial detectors.** We use the same criteria as in Table 1.

other commercial detectors. Overall, training against RoBERTa-large or Originality produces the strongest evasion of commercial detectors.

3.2 Impact of Detector Query Quota on Evasion

In a real-world scenario, a malicious actor is likely to be constrained by the amount of paired preference data used in training, especially if optimizing against a commercial detector. We demonstrate that strong performance of detectors can be maintained even when training data is limited. We trained five Llama2-7b models against OpenAI's large RoBERTa-based detector, using $\beta = 0.5$ for all models. Each model was trained for up to 100k steps on a different training set size (1k, 5k, 10k, 25k, and 100k preference pairs). We observe several significant findings, which are presented in Figure 1. First, detector evasion is possible with only a small number ($<$10k) queries to the detector. For the commercial detectors we study, this number of queries typically costs less than \$150 (in several cases significantly so), making detector evasion a very accessible procedure even for small budgets. These results suggest that preventing an adversary from collecting a dataset of detector evaluations large enough to train an undetectable model may be extremely difficult or impossible.

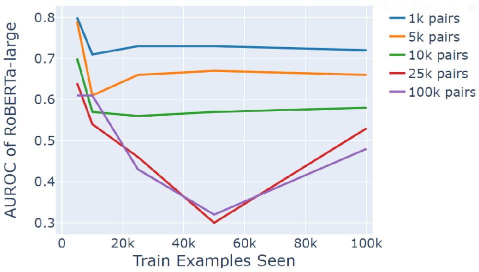

Figure 1: **Only a small number of detector queries are needed to produce a dataset large enough to confuse even a strong detector (RoBERTa-large).**

3.3 Robustness of Evasion to Longer Sequence Length

Prior work has shown that detector accuracy improves as the length of the generated data increases (10). We therefore conduct an experiment to assess whether this pattern holds true after a language model has been optimized against a given detector. The results are shown in Table 3. We draw samples from a Llama-2 7B base model before and after it has been fine-tuned with $\beta = 0.5$ to evade RoBERTa-large for 100k preference pairs. While the pre-evasion model generates samples that are increasingly easy to detect as sequence length increases, after fine-tuning, doubling the sequence length does not lead to increased AUROC, but rather a slight decrease. This result shows that detector evasion fine-tuning can generalize to longer sequences than it was trained for, further increasing its general applicability.

	Source Model	
Seq. length	**Base**	**Post-evasion**
$\bar{n} = 120$	0.84	0.63
$\bar{n} = 243$	0.92	0.61

Table 3: **Sampling longer responses from an evasion-tuned model does not improve detector AUROC for RoBERTa-large.**

3.4 Case Study: Essay Generation

Finally, to analyze the feasibility of optimizing against existing detectors in a domain more closely aligned with real-world usage, we explore evasion tuning in the context of generating essays. We ask two questions: *Can we evasion-tune a chat model to generate essays that confuse a strong detector?* and a more challenging question *Can we evasion-tune a general-purpose dialogue model that still produces essays that confuse a strong detector, without gathering new detector-annotated preferences or fine-tuning specifically for essay generation?* While an affirmative answer to either question is cause for concern, an affirmative answer to the second is much more serious: in this case, we do not need to evasion tune again for each new domain in which we would like to evade a detector, and further, in order to do so, we can re-use a single set of preference data generated by another model (in this case, ChatGPT).

To answer these questions, we perform detector evasion on a LLama-2-chat 7B (23), using two different datasets, one specifically essay prompts and essays generated by Llama-2-chat, and the other more general instruction-following prompts from Alpaca (22) and preference data over samples from ChatGPT, rather than the Llama-2-chat model. We fine-tune both models for 30k steps using $\beta = 0.5$; the Llama-generated preference samples are 250 tokens long; we prompt ChatGPT to write a 'single mid-length paragraph' on the given topic, discarding samples less than 100 Llama-2 tokens. **The results in Table 4 show that optimizing against RoBERTa-large is successful in both of these cases: fine-tuning a general-purpose chat model to evade a detector using general-purpose instruction following prompts and off-policy samples nonetheless can evade a detector in the specific case of generating essays.**

	Source Model		
Metric	**Base**	**Essay training**	**Dialogue training**
AUROC	0.83	0.26	0.43
Perplexity	6.0	7.0	7.0

Table 4: **A case study in generating difficult-to-detect essays from Llama-7b-chat.** We perform detector evasion tuning on preferences generated by RoBERTa-large.

4 Discussion

Motivated by the increasingly widespread use of large language model detectors, we have shown that it is straightforward to fine-tune a model to evade these detectors while still maintaining high performance. The fine-tuned models produce text that is almost completely undetectable by two out of four commercial detectors. For the other two commercial detectors, the fine-tuned models have an AUROC of less than 0.5, indicating that they produce text that is judged to be statistically *more* likely to be human than the human-written corpus itself. Moreover, generating long-form text, such as essays, does not increase detectability.

We emphasize that the training pipeline that we consider is straightforward and easy for adversaries to replicate. It uses easily accessible public models and an open-source training codebase. The entire data acquisition and training process cost a few hundred dollars. For data, the process only requires limited, black-box query access to the detector with a budget of a few thousand prompts and does not need human annotators, paraphrasing, or teacher models. For compute, we used widely available consumer hardware and only a few hours of training time. We further expect that, with more extensive data, training, and resources, a fine-tuned model may be even more evasive.

We expect that this direct kind of attack is hard to protect against. Indeed, our results showed meaningful transfer between strong detectors. Thus, in light of these results, we argue that the current generation of machine-generated text detectors is not robust to adversaries and may even favor machine-generated text over actual human-generated content. This includes both public detectors and closed black-box commercial ones. Furthermore, we argue that the problem of robust machine-generated text detection may be unsolvable in practical settings. Any new detection algorithm can be subject to the adversarial training process in this paper. New detection algorithms will be rendered ineffective by further model fine-tuning, which would then require the development of new detection algorithms. Hence, we argue against continued use of machine-generated text detectors.

References

[1] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. 2

[2] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. Real or fake? Learning to discriminate machine from human generated text. *arXiv*, 2019. URL http://arxiv.org/abs/1906.03351. 9

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf. 1

[4] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. 1

[5] William Fedus, Ian Goodfellow, and Andrew M. Dai. Maskgan: Better text generation via filling in the______, 2018. 9

[6] Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 111–116, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3019. URL https://aclanthology.org/P19-3019. 1, 9

[7] Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019. 3

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf. 9

[9] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1808–1822, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.164. URL https://www.aclweb.org/anthology/2020.acl-main.164. 9

[10] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html. 4, 9

[11] Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *arXiv preprint arXiv:2303.13408*, 2023. 1, 9

[12] Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models, 2023. 9

[13] Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. Gpt detectors are biased against non-native english writers. *arXiv preprint arXiv:2304.02819*, 2023. 9

[14] Fatemehsadat Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. Smaller language models are better black-box machine-generated text detectors. *arXiv preprint arXiv:2305.09859*, 2023. 9

[15] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html. 1, 3, 9

[16] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 2

[17] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023. 2, 9

[18] Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023. 9

[19] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. Release strategies and the social impacts of language models, 2019. URL https://arxiv.org/ftp/arxiv/papers/1908/1908.09203.pdf. 1, 3, 9

[20] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. *Neural Information Processing Systems*, 18, 2020. 2

[21] Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*, 2023. 3, 9

[22] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023. 3, 5

[23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin

Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 3, 5

[24] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8384–8395, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.673. URL https://aclanthology.org/2020.emnlp-main.673. 9

[25] Michael Vechtomov. qwedsacf/ivypanda-essays dataset, 2023. URL https://huggingface.co/datasets/qwedsacf/ivypanda-essays/viewer/default/train. Available from Hugging Face Dataset Hub. 3

[26] Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. Ghostbuster: Detecting text ghostwritten by large language models, 2023. 9

[27] KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. Robust multi-bit natural language watermarking through invariant features. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2092–2115, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.117. URL https://aclanthology.org/2023.acl-long.117. 9

[28] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient, 2017. 9

[29] Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text, 2023. 9

A Related Work

Machine-generated text detection methods often either train a classifier using a dataset of LM- and human-generated text (2; 19; 24; 9; 26) or detect zero-shot by leveraging the suspected language model or a proxy (19; 6; 15; 21). Prior works have called into question the robustness of these detectors, finding that detectors are susceptible to paraphrasing attacks (18; 11). and can perform poorly for text written by non-native speakers (13). Further, Mitchell et al. (15) and Mireshghallah et al. (14) show that zero-shot detectors show significantly reduced performance when the generating model is not known. By showing that it is straightforward to optimize against current detectors, our results complement these prior studies, while continuing to suggest that machine-generated text detectors are not robust.

Another class of works have aimed to train language models that produce subtle 'watermarks,' i.e. indications that they were generated by a machine (10; 29; 12; 27). However, the premise of watermarking relies on the fact that all strong models in the LLM ecosystem are watermarked (i.e., hosted behind APIs that enforce watermarking); a single strong LLM with freely-available weights violates this threat model. We consider a stronger threat model where an adversary is fine-tuning the model to be undetectable.

Finally, Solaiman et al. (19) train a detector to discriminate between human samples and samples generated by a pre-trained model (GPT-2). In our case, fine-tuning that pre-trained model to maximize the 'human' probability of the detector with DPO (17) is very similar to performing one round of generator improvement in a generative adversarial network (GAN; Goodfellow et al. (8)). While adversarial objectives are typically avoided for text data due to the difficulty of differentiating through the discrete sampling step, Yu et al. (28) and Fedus et al. (5) show that GAN language models can produce more realistic-looking samples than MLE when evaluated by humans. There results provide some precedent for our GAN-like training procedure for our use case of generating human-looking samples.

B Social Impacts Statement

Evading language model detectors is a type of red-teaming exercise that we carry out in order to call attention to the serious risks of relying on any machine-generated text detection technologies. We categorically do **not** advocate for evading language model detectors for the purpose of carrying out harmful activities with LLMs. Rather, we hope that in demonstrating the ease with which the effectiveness of existing detectors can be severely degraded, we can spur a conversation about these technologies. Ultimately, we believe swift action to revise institutional norms, particularly standards in classrooms around student assessment, is warranted.

C Reproducibility

Section 2 covers the details of the direct preference optimization algorithm, and an open-source implementation is available in the cited paper. An anonymized implementation of our pipeline and experiments can be made available during the review process. Precise descriptions of the fine-tuning and model selection process for non-chat models are available in sections 3.1, 3.2, and 3.3, while the corresponding information for chat models can be found in section 3.4. Anonymized datasets of preference pairs and detector scores for all open-source models can also be made available during the review process, though releasing datasets for commercial detectors violates their terms of use.