Exploring Scaling Trends in LLM Robustness

Nikolaus Howe [1 2 3] **Michał Zając** [1] **Ian McKenzie** [1] **Oskar Hollinsworth** [1]
Pierre-Luc Bacon [2 3] **Adam Gleave** [1]

Abstract

Language model capabilities predictably improve from scaling the model's size and training data. Motivated by this, increasingly large language models have been trained, yielding an array of impressive capabilities. Yet these models suffer from adversarial prompts such as "jailbreaks" that hijack models to perform undesired behavior, posing a significant risk of misuse. Prior work has found that computer vision models become more robust with model and data scaling, raising the question: does language model robustness also improve with scale? We study this question empirically, finding that larger models respond substantially more effectively to adversarial training, but there is little to no benefit from model scale in the absence of defenses.

1. Introduction

Language models have demonstrated a range of impressive capabilities in tasks such as general reasoning (Hendrycks et al., 2021), graduate level Q&A (Rein et al., 2023) and code generation (Chen et al., 2021). This growth in capabilities has fueled rapid deployment, with ChatGPT becoming one of the fastest-growing consumer applications in history (Hu, 2023). Moreover, language models are increasingly integrated into larger systems enabling them to take actions in the real world using external tools (OpenAI, 2023; Anthropic, 2024; Google, 2024) and pursue long-term open-ended goals (Richards, 2024; Kinniment et al., 2024).

The advent of language models enables many new tasks to be solved by AI, but also introduces novel classes of security vulnerabilities. In particular, a wide variety of adversarial prompts can hijack models (Wei et al., 2023; Zou et al., 2023; Anil et al., 2024). This enables malicious users to bypass safety fine-tuning performed by the designer, unlocking harmful capabilities such as generating compelling misinformation (Spitale et al., 2023; Chen & Shu, 2024). Innocent users are also at risk from attackers using methods such as indirect prompt injections (Abdelnabi et al., 2023) to exploit LLM-driven applications without any awareness or participation by the user.

A key question is whether future, more capable systems will *naturally* become more robust, or if this will instead require a dedicated safety effort. Although current attacks are concerning, the risks could grow still greater with future models capable of more dangerous actions, such as assisting with biological weapon development (Mouton et al., 2023), or with greater *affordances* to interact with the world (Sharkey et al., 2023), such as a virtual assistant for a CEO of a major company. Prior work has found that superhuman Go systems (Wang et al., 2023) are vulnerable to attack, demonstrating that impressive capabilities do not guarantee robustness. However, work has also found that scaling unlabeled pretraining data (Hendrycks et al., 2019; Carmon et al., 2022; Alayrac et al., 2019) and model size (Xie & Yuille, 2019; Huang et al., 2023) improves adversarial robustness in computer vision.

To answer this question, we conduct an empirical investigation into scaling trends for the adversarial robustness of language models. These trends enable us to forecast the robustness of future models, and give us insight into how the offense-defense balance might shift over time. For example, does the cost of conducting an attack against more capable models grow faster or slower than the defender's cost of training those models?

Concretely, we investigate the robustness of 14M to 12B parameter Pythia models (Biderman et al., 2023) to two attacks: the *random tokens* baseline and the state-of-the-art *greedy coordinate gradient* attack. We test these models in a variety of simple classification tasks on which our models obtain high accuracy given clean (non-adversarial) data.

We first evaluate these pretrained models fine-tuned only on clean data. Larger models tend to be more resistant to attack, but the effect is weak and noisy (Figure 1). By contrast, a clearer scaling trend emerges for models adversarially trained against examples of attacks (Figure 2).

[1]FAR AI [2]Mila – Quebec Artificial Intelligence Institute [3]Université de Montréal. Correspondence to: {Nikolaus Howe, Adam Gleave} <{niki, adam}@far.ai>.

Presented at the NextGenAISafety Workshop at the 41st International Conference on Machine Learning, Vienna, Austria. Copyright 2024 by the author(s).

Larger models are both more sample efficient, learning to be robust with fewer examples, and converge to be more robust given unlimited examples (Figure 18). Moreover, adversarial training against one attack transfers to protect against similar attacks, with the transfer being *stronger* for larger models (Figure 4).

2. Related Work

Adversarial examples were first identified in image classifiers (Szegedy et al., 2014), but have since been found for systems performing image captioning (Xu et al., 2019; Zhang et al., 2020), speech recognition (Cisse et al., 2017; Alzantot et al., 2018; Schönherr et al., 2018) and reinforcement learning (Huang et al., 2017; Gleave et al., 2020; Ilahi et al., 2022). Moreover, a range of adversarial threat models (Gilmer et al., 2018) give rise to viable attacks.

Most recently, many qualitatively different vulnerabilities have been found in language models, from human-understandable "jailbreaks" (Wei et al., 2023) to seemingly gibberish adversarial suffixes (Wallace et al., 2021; Zou et al., 2023). Simple methods such as perplexity filtering and paraphrasing defend against some of these attacks (Jain et al., 2023). However, these defenses can easily be bypassed by more sophisticated methods (Zhu et al., 2023). Adversarial training shows more promise as a defense (Ziegler et al., 2022), and is the focus of our analysis.

The determinants of adversarial robustness have been well-studied in computer vision. One line of scholarship proposes a fundamental tradeoff between robustness and accuracy (Tsipras et al., 2019): exploitable models are simply relying on non-robust features (Ilyas et al., 2019), which improve training performance but hurt robustness. Other work has emphasized what *does* improve robustness. Scaling unlabeled pretraining data (Hendrycks et al., 2019; Carmon et al., 2022; Alayrac et al., 2019) and model depth (Xie & Yuille, 2019) and width (Huang et al., 2023) improves adversarial robustness in the computer vision domain. However, other work shows that adversarial robustness in computer vision scales too slowly to be a complete solution (Debenedetti et al., 2023; Bartoldson et al., 2024).

Language model scaling laws (Hestness et al., 2017; Rosenfeld et al., 2019; Kaplan et al., 2020; Hoffmann et al., 2022) have shown that increasing compute improves performance across many tasks and domains (Chen et al., 2021; Hernandez et al., 2021). However, scaling does not solve all problems (Lin et al., 2022; McKenzie et al., 2023). There has been only limited work on scaling laws for adversarial robustness in language models, with mixed results. Ganguli et al. (2022) show that LLMs become harder to attack with scale—but Anil et al. (2024) find that some attacks become *more successful* with scale.

3. Experimental Methodology

We test models in the binary classification setting, as it is the simplest setting in which we can study LLM robustness. Crucially, binary classification allows us to measure robustness by the **attack success rate**, defined as the proportion of examples correctly classified by the model before attack that are incorrectly classified after attack.[1] We adapt pretrained models for classification by replacing the unembedding layer with a randomly initialized classification head, and then fine-tune the models on each task.

Tasks. We consider four tasks in our experiments, the latter two developed by us for this project:

- `Spam` (Metsis et al., 2006): Given the subject and body of an email, is it spam or not?
- `IMDB` (Maas et al., 2011): Given a movie review, is the sentiment positive or negative?
- `PasswordMatch`: Given two strings in the prompt, are they exactly equal?
- `WordLength`: Given two words in the prompt, is the first word shorter than the second?

`Spam` and `IMDB` were chosen as standard natural language processing classification tasks. `PasswordMatch` was inspired by TensorTrust (Toyer et al., 2023) and `WordLength` by the RuLES dataset (Mu et al., 2023). Both `PasswordMatch` and `WordLength` were designed to be easily procedurally generated and have ground truth labels that can be checked algorithmically. For brevity, we report on `Spam` and `IMDB` in the main text, with plots for other tasks deferred to appendices D and E. We provide example datapoints and details about the datasets in Appendix B.

Models We test the Pythia (Biderman et al., 2023) model family. These models range in size from 14M to 12B parameters (or 7.6M to 11.6B when used with a classification head). All models were trained to predict the next token on the same dataset following the same training protocol, allowing us to isolate model scale from other confounding factors.

Attacks We use two different attack procedures in our experiments: the greedy coordinate gradient attack (`GCG`; Zou et al., 2023) and a random token baseline (`RandomToken`). `GCG` was chosen because it is currently one of the most effective attacks on language models, and is suitable for attacking a wide variety of tasks. `RandomToken` was chosen due to its simplicity and its similarity to `GCG`; both of these attacks work by appending an adversarial suffix of N tokens to the

[1] We assume that the attack does not, in fact, change the ground truth label of the data point. This is guaranteed by construction for some of our simple procedurally generated tasks, and was manually validated on a random sample of data points in other tasks.

prompt.

`RandomToken` is a simple baseline where the N tokens are chosen uniformly at random from the model's vocabulary. We evaluate the model on the attacked text and then repeat the process with another sample of N random tokens until the model is successfully attacked or an appointed budget for model calls is exhausted.

In `GCG` (Zou et al., 2023) the N tokens are initialized arbitrarily, and then greedily optimized over multiple rounds. In each round, the gradient of the loss function with respect to the attack tokens is computed. This gradient is used to compute a set of promising single-token modifications, from which the best candidate is selected and used in the next round. To make this attack work in the classification setting, we use the cross-entropy loss between the predicted label and the target label as the loss function to optimize against.

In our experiments, we always use $N = 10$ tokens. For more details about the attacks and hyperparameters used, see Appendix C.

4. Fine-tuning

Figure 1 shows the robustness of fine-tuned models against the `GCG` attack. The attack is generally less successful on larger models, but model size alone does not explain all the variance in attack success rate. We observe similarly large random variation in attack success across model sizes on other tasks and with other attacks; for more details, see Appendix D.2.

As described in Section 3, we use the Pythia models, which range from 7.6M to 11.6B parameters after replacing the unembedding matrix with a classification head.[2] We fine-tune all models for a single epoch with default hyperparameters from HuggingFace Transformers (Wolf et al., 2019), except for the learning rate which we set to $1\mathrm{e}{-5}$. We observe that all models reach >83% accuracy on all tasks, with larger models generally performing better (see Appendix D.1 for the final validation performance of all models on all tasks). We then evaluate the fine-tuned models against adversarial attacks on an unseen validation dataset.

In an attempt to understand the source of the variability in model robustness shown by our experiments, we varied 1) the pretraining checkpoint,[3] and 2) the random seeds used to initialize the classification head before fine-tuning. We found both factors led to significant variability in model robustness, with pretraining checkpoint contributing significantly more variability. The variability was comparable or greater to that of an order of magnitude of model scaling, indicating that out-of-the-box robustness on a given task is heavily influenced by the randomness of the pretraining procedure itself.

This initial result suggests that we cannot rely on scale alone to solve the problem of robustness. However, in practice we would apply a defense to a model prior to deploying it in a security-critical setting. In the following section, we consider whether scale enables defenses to more effectively improve model robustness.

5. Adversarial training

In this section, we explore how model size impacts robustness when performing adversarial training. Figure 2 evaluates the robustness of Pythia models to the `GCG` attack when adversarially trained against the same attack. We see a much cleaner trend than in the fine-tuning only case: larger models gain robustness more quickly and converge to be more robust than smaller models. These results suggest that model size is a strong predictor of robustness—so long as the model is explicitly optimized for robustness. We observe similar behaviour across the other two datasets and two attacks; see Appendix E for these plots including extensions for up to 30 adversarial training rounds.

We perform adversarial training by iteratively training our model on a train dataset, evaluating the model on attacked examples, and then adding successful attack examples to the train dataset. Simultaneously, we evaluate model performance on a held-out attacked validation dataset. This procedure is illustrated in Figure 11.

In our experiments, we start with a training dataset of 2000 clean examples, and add 200 new adversarial examples to the training dataset each round. We repeat the train-attack-add loop 30 times (here we only show the first 10 rounds; see Appendix E for the full 30 round plots). Since adversarial examples are only added after the first training round, the models here were trained for a single epoch on the 2000 clean datapoints before being adversarially attacked.

We perform adversarial training on Pythia models ranging from 7.6 to 909 million parameters after replacing the unembedding layer with a classification head.[4] Table 1, in Appendix A, enumerates all model sizes along with corresponding plot colors.

[2] In all figures, we report the actual parameter count of the classification model, and not the pretrained model it was derived from.

[3] The Pythia models provide checkpoints from earlier stages of pretraining. We used various checkpoints from the final 10% of pretraining as a starting point for fine-tuning.

[4] Specifically, we use the `pythia-14m` to `pythia-1b` models loaded as `AutoModelForSequenceClassification`.

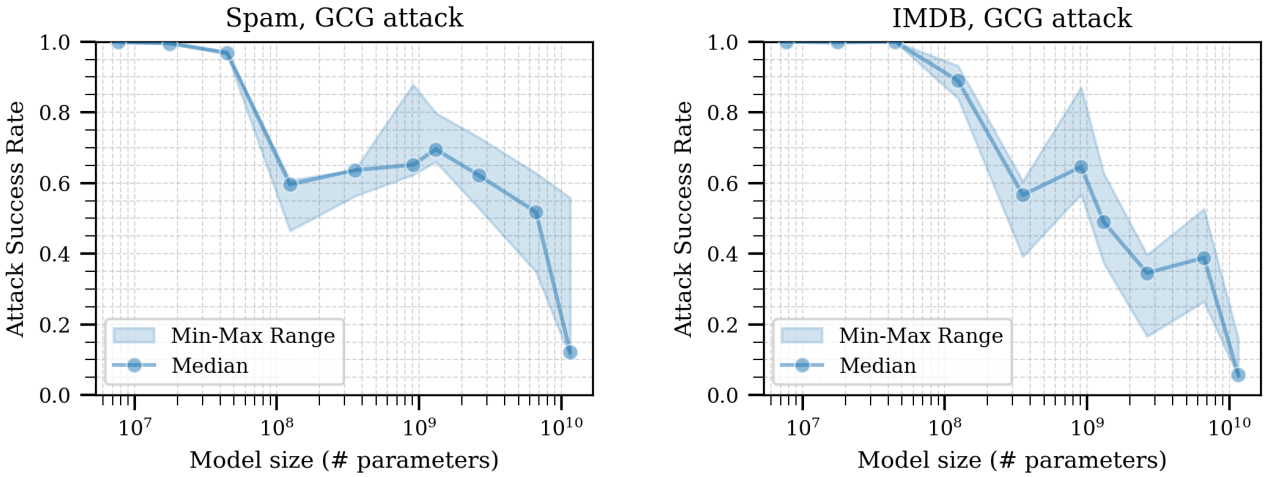


Figure 1. Attack success rate applying the GCG attack against Pythia models of different sizes fine-tuned on the `Spam` (left) and `IMBD` (right) tasks. We run three fine-tuning seeds for each model, and plot min, max, and median attack success rate for each model size. Across model sizes, we observe significant attack success rate variability: median robustness does not improve monotonically with scale.

5.1. Robustness transfer

In practice, we often do not have the luxury of knowing the exact attack method an adversary may employ against our model. For real-world impact, we need adversarial training on a handful of attacks to provide more general robustness against other unforeseen attacks as well. In this subsection, we study whether we observe this transfer in robustness between attacks—and how model scale affects the transfer.

First, we explore whether robustness from adversarial training transfers to a stronger, yet in-distribution attack. To do this, we adversarially train using the procedure described above using GCG for 10 iterations as our training attack. We then evaluate on GCG for 30 iterations, a stronger attack. Figure 3 shows that larger models are more robust to this in-distribution, stronger attack. Although the transfer is imperfect—the models do, of course, lose against 30-iteration GCG more than against 10-iteration GCG—the performance is much better than the undefended (fine-tuned) models, which lose approximately 100% of the time.

This is a promising result. Yet, what happens if our models experience an attack that is not only stronger, but also uses a different method than the one on which they were adversarially trained? We investigate this question by performing adversarial training on the `RandomToken` attack, and evaluating on the `GCG` attack. Figure 4 shows models adversarially trained on `RandomToken` do perform better than undefended models, though the effect is weaker. Critically, the extent to which transfer occurs varies drastically across models. In particular, the models with more than 100 million parameters all show strong transfer behaviour, with the attack success rate falling below 25% after just 4 iterations of adversarial training. On the other hand, models with fewer than 100 million parameters struggle to transfer their robustness against the `RandomToken` attack to the stronger `GCG` attack, with the attack success rate still near 70% on the strongest model even after 10 adversarial training rounds.

This finding is encouraging as it suggests that, for sufficiently large models, it is possible that robustness will transfer across attacks. It appears that this transfer might be a property that emerges with sufficient scale, similarly to other emergent properties like the ability to use a scratchpad for addition, or the utility of instruction fine-tuning (Wei et al., 2022). While we cannot say with certainty that such transfer of robustness generalizes outside the settings and attacks considered in this work, it seems plausible that it would, and indeed, that scaling to further orders of magnitude could unlock more general transfer to a wider variety of attack methodologies and strengths.

6. Conclusion

Our results demonstrate that larger Pythia models benefit more from adversarial training in a variety of classification tasks than do smaller Pythia models. An important direction for future work is to validate this trend holds in a broader variety of settings. In particular, we plan to study generative tasks, and how factors such as the complexity of a task affect robustness. We also plan to investigate different model families, including larger models.

A key application of scaling trends is being able to appropriately size models for robustness given a fixed defender compute budget. Although larger models are more efficient given a fixed number of adversarial training time steps,

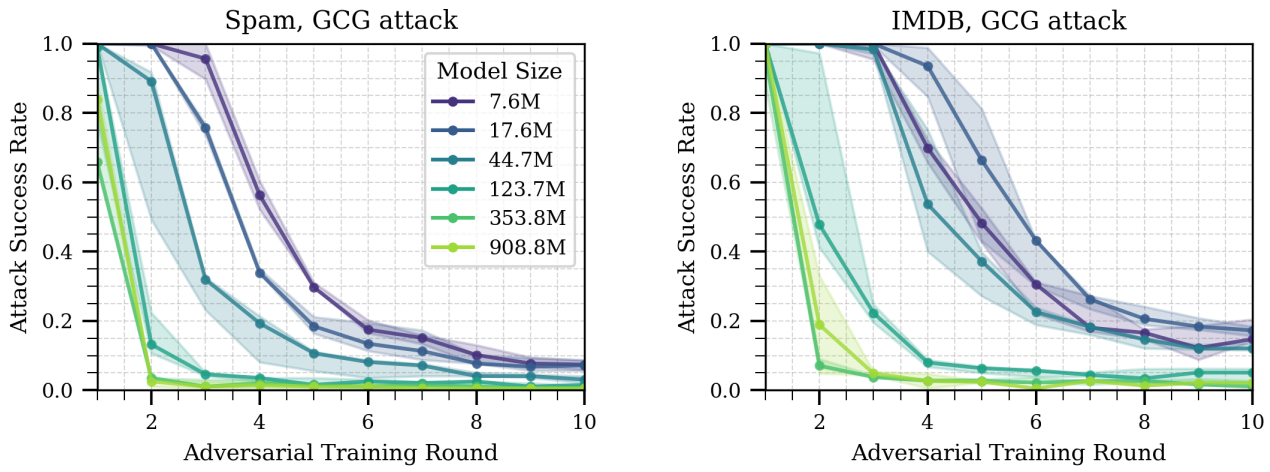


Figure 2. Attack success rate applying the GCG attack against Pythia models of different sizes during adversarial training on `Spam` (left) and `IMDB` (right). We shade min to max and plot median, taken over three seeds (except for a small number of points; see Table 5). Here we observe a clear relationship between model size and decreasing attack success rate over adversarial training rounds.

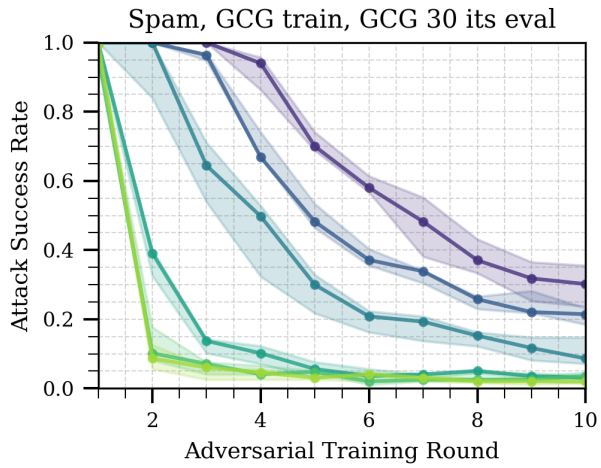


Figure 3. Attack success rate applying the 30-iteration GCG attack against Pythia models of different sizes during adversarial training using the 10-iteration GCG attack. All models are able to somewhat transfer their defense to this stronger attack, with larger models doing so more effectively.

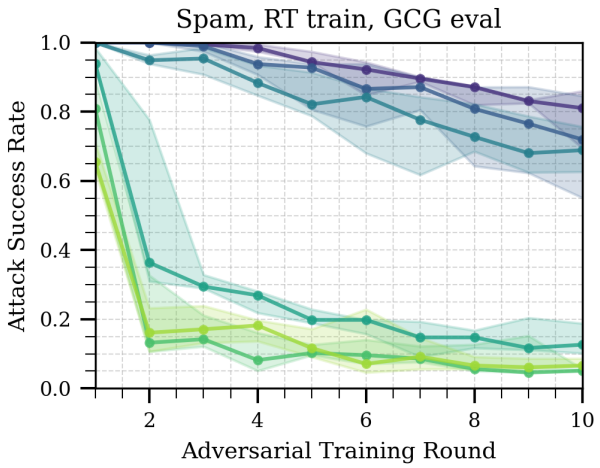


Figure 4. Attack success rate applying the 10-iteration `GCG` attack against Pythia models of different sizes during adversarial training using the 10-iteration `RandomToken` (RT) attack. Models larger than 100M parameters show strong transfer behavior, while smaller models struggle against the different attack.

performing the adversarial training is more expensive with bigger models. For example, Figure 2 shows that performing 8 adversarial training rounds on the 17.6M parameter model results in better robustness than performing 4 adversarial training rounds on the 44.7M parameter model, and a quick calculation shows that it is slightly less expensive to train (see Appendix E.4 for the calculation). However, using a smaller model is not always better, since there are diminishing returns to adversarial training with larger models converging to be more robust.

Although scale can improve robustness, our results make clear that it is far from the only determinant. For example, a small adversarially trained model is more robust than a large model fine-tuned only on clean data. We expect that achieving robust language models will require a combination of innovations in defense techniques, as well as scaling model pre-training and defense training. Scaling trends both enable us to measure how far we are from achieving robustness by scale alone, and enable us to identify defense techniques that can better leverage scale to produce more robust models.

Acknowledgements

The authors thank Daniel Pandori for his contributions to the codebase during the early stages of this project. Nikolaus Howe thanks the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support via the Vanier Canada Graduate Scholarship.

References

Abdelnabi, S., Greshake, K., Mishra, S., Endres, C., Holz, T., and Fritz, M. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *AISec*, pp. 79–90, 2023.

Alayrac, J.-B., Uesato, J., Huang, P.-S., Fawzi, A., Stanforth, R., and Kohli, P. Are Labels Required for Improving Adversarial Robustness? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/bea6cfd50b4f5e3c735a972cf0eb8450-Abstract.html.

Alzantot, M., Balaji, B., and Srivastava, M. Did you hear that? adversarial examples against automatic speech recognition, 2018.

Anil, C., Durmus, E., Sharma, M., Benton, J., Kundu, S., Batson, J., Rimsky, N., Tong, M., Mu, J., Ford, D., Mosconi, F., Agrawal, R., Schaeffer, R., Bashkansky, N., Svenningsen, S., Lambert, M., Radhakrishnan, A., Denison, C., Hubinger, E. J., Bai, Y., Bricken, T., Maxwell, T., Schiefer, N., Sully, J., Tamkin, A., Lanham, T., Nguyen, K., Korbak, T., Kaplan, J., Ganguli, D., Bowman, S. R., Perez, E., Grosse, R., and Duvenaud, D. Many-shot Jailbreaking, 2024.

Anthropic. Tool use (function calling), 2024. URL https://archive.ph/EqXCz.

Bartoldson, B. R., Diffenderfer, J., Parasyris, K., and Kailkhura, B. Adversarial Robustness Limits via Scaling-Law and Human-Alignment Studies, April 2024. URL http://arxiv.org/abs/2404.09349. arXiv:2404.09349 [cs].

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., and Duchi, J. C. Unlabeled Data Improves Adversarial Robustness, January 2022. URL http://arxiv.org/abs/1905.13736. arXiv:1905.13736 [cs, stat].

Chen, C. and Shu, K. Can LLM-generated misinformation be detected? In *International Conference on Learning Representations*, 2024.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating Large Language Models Trained on Code, July 2021. URL http://arxiv.org/abs/2107.03374. arXiv:2107.03374 [cs].

Cisse, M. M., Adi, Y., Neverova, N., and Keshet, J. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In *NeurIPS*, volume 30, 2017.

Debenedetti, E., Wan, Z., Andriushchenko, M., Sehwag, V., Bhardwaj, K., and Kailkhura, B. Scaling Compute Is Not All You Need for Adversarial Robustness, December 2023. URL http://arxiv.org/abs/2312.13131. arXiv:2312.13131 [cs].

Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., Jones, A., Bowman, S., Chen, A., Conerly, T., DasSarma, N., Drain, D., Elhage, N., El-Showk, S., Fort, S., Hatfield-Dodds, Z., Henighan, T., Hernandez, D., Hume, T., Jacobson, J., Johnston, S., Kravec, S., Olsson, C., Ringer, S., Tran-Johnson, E., Amodei, D., Brown, T., Joseph, N., McCandlish, S., Olah, C., Kaplan, J., and Clark, J. Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned, November 2022. URL http://arxiv.org/abs/2209.07858. arXiv:2209.07858 [cs].

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu, M., Wattenberg, M., and Goodfellow, I. Adversarial Spheres, September 2018. URL http://arxiv.org/abs/1801.02774. arXiv:1801.02774 [cs].

Gleave, A., Dennis, M., Wild, C., Kant, N., Levine, S., and Russell, S. Adversarial policies: Attacking deep reinforcement learning. In *ICLR*, 2020.

Google. Function calling | google ai for developers, 2024. URL https://archive.ph/YGJHJ.

Hendrycks, D., Lee, K., and Mazeika, M. Using Pre-Training Can Improve Model Robustness and Uncertainty. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2712–2721. PMLR, May 2019. URL https://proceedings.mlr.press/v97/hendrycks19a.html. ISSN: 2640-3498.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Hernandez, D., Kaplan, J., Henighan, T., and McCandlish, S. Scaling Laws for Transfer, February 2021. URL http://arxiv.org/abs/2102.01293. arXiv:2102.01293 [cs].

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. Deep Learning Scaling is Predictable, Empirically, December 2017. URL http://arxiv.org/abs/1712.00409. arXiv:1712.00409 [cs, stat].

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. v. d., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training Compute-Optimal Large Language Models, March 2022. URL http://arxiv.org/abs/2203.15556. arXiv:2203.15556 [cs].

Hu, K. Chatgpt sets record for fastest-growing user base – analyst note. *Reuters*, 2023.

Huang, S., Lu, Z., Deb, K., and Boddeti, V. N. Revisiting Residual Networks for Adversarial Robustness. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8202–8211, Vancouver, BC, Canada, June 2023. IEEE. ISBN 9798350301298. doi: 10.1109/CVPR52729.2023.00793. URL https://ieeexplore.ieee.org/document/10204909/.

Huang, S. H., Papernot, N., Goodfellow, I. J., Duan, Y., and Abbeel, P. Adversarial attacks on neural network policies. arXiv:1702.02284v1 [cs.LG], 2017.

Ilahi, I., Usama, M., Qadir, J., Janjua, M. U., Al-Fuqaha, A., Hoang, D. T., and Niyato, D. Challenges and countermeasures for adversarial attacks on deep reinforcement learning. *IEEE TAI*, 3(2):90–109, 2022.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial Examples Are Not Bugs, They Are Features. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/e2c420d928d4bf8ce0ff2ec19b371514-Abstract.html.

Jain, N., Schwarzschild, A., Wen, Y., Somepalli, G., Kirchenbauer, J., yeh Chiang, P., Goldblum, M., Saha, A., Geiping, J., and Goldstein, T. Baseline defenses for adversarial attacks against aligned language models, 2023.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling Laws for Neural Language Models, January 2020. URL http://arxiv.org/abs/2001.08361. arXiv:2001.08361 [cs, stat].

Kinniment, M., Sato, L. J. K., Du, H., Goodrich, B., Hasin, M., Chan, L., Miles, L. H., Lin, T. R., Wijk, H., Burget, J., Ho, A., Barnes, E., and Christiano, P. Evaluating language-model agents on realistic autonomous tasks, 2024.

Lin, S., Hilton, J., and Evans, O. TruthfulQA: Measuring How Models Mimic Human Falsehoods, May 2022. URL http://arxiv.org/abs/2109.07958. arXiv:2109.07958 [cs].

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

McKenzie, I. R., Lyzhov, A., Pieler, M. M., Parrish, A., Mueller, A., Prabhu, A., McLean, E., Shen, X., Cavanagh, J., Gritsevskiy, A. G., Kauffman, D., Kirtland, A. T., Zhou, Z., Zhang, Y., Huang, S., Wurgaft, D., Weiss, M., Ross, A., Recchia, G., Liu, A., Liu, J., Tseng, T., Korbak, T., Kim, N., Bowman, S. R., and Perez, E. Inverse Scaling: When Bigger Isn't Better. *Transactions on Machine Learning Research*, June 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=DwgRm72GQF.

Metsis, V., Androutsopoulos, I., and Paliouras, G. Spam Filtering with Naive Bayes - Which Naive Bayes? In *CEAS*, 2006. URL https://www2.aueb.gr/users/ion/docs/ceas2006_paper.pdf.

Mouton, C. A., Lucas, C., and Guest, E. *The Operational Risks of AI in Large-Scale Biological Attacks: A Red-Team Approach*. RAND Corporation, 2023.

Mu, N., Chen, S., Wang, Z., Chen, S., Karamardian, D., Aljeraisy, L., Alomair, B., Hendrycks, D., and Wagner, D. Can llms follow simple rules? *arXiv*, 2023.

OpenAI. Assistants API documentation, 2023. URL https://archive.ph/8Az8d.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark, 2023.

Richards, T. B. Auto-gpt: An autonomous gpt-4 experiment, 2024. URL https://github.com/Significant-Gravitas/AutoGPT/.

Rosenfeld, J. S., Rosenfeld, A., Belinkov, Y., and Shavit, N. A Constructive Prediction of the Generalization Error Across Scales, December 2019. URL http://arxiv.org/abs/1909.12673. arXiv:1909.12673 [cs, stat].

Schönherr, L., Kohls, K., Zeiler, S., Holz, T., and Kolossa, D. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding, 2018.

Sharkey, L., Ghuidhir, C. N., Braun, D., Scheurer, J., Balesni, M., Bushnaq, L., Stix, C., and Hobbhahn, M. A causal framework for AI regulation and auditing. Technical report, Apollo Research, 2023.

Spitale, G., Biller-Andorno, N., and Germani, F. AI model GPT-3 (dis)informs us better than humans. *Science Advances*, 9(26), 2023.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks, 2014.

Toyer, S., Watkins, O., Mendes, E. A., Svegliato, J., Bailey, L., Wang, T., Ong, I., Elmaaroufi, K., Abbeel, P., Darrell, T., Ritter, A., and Russell, S. Tensor Trust: Interpretable prompt injection attacks from an online game, 2023. URL https://arxiv.org/pdf/2311.01011.pdf.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *ICLR*, 2019.

Wallace, E., Feng, S., Kandpal, N., Gardner, M., and Singh, S. Universal Adversarial Triggers for Attacking and Analyzing NLP, January 2021. URL http://arxiv.org/abs/1908.07125. arXiv:1908.07125 [cs].

Wang, T. T., Gleave, A., Tseng, T., Pelrine, K., Belrose, N., Miller, J., Dennis, M. D., Duan, Y., Pogrebniak, V., Levine, S., and Russell, S. Adversarial Policies Beat Superhuman Go AIs, July 2023. URL http://arxiv.org/abs/2211.00241. arXiv:2211.00241 [cs, stat].

Wei, A., Haghtalab, N., and Steinhardt, J. Jailbroken: How Does LLM Safety Training Fail?, July 2023. URL http://arxiv.org/abs/2307.02483. arXiv:2307.02483 [cs].

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Xie, C. and Yuille, A. Intriguing Properties of Adversarial Training at Scale. In *ICLR*, September 2019. URL https://openreview.net/forum?id=HyxJhCEFDS.

Xu, Y., Wu, B., Shen, F., Fan, Y., Zhang, Y., Shen, H. T., and Liu, W. Exact adversarial attack to image captioning via structured output learning with latent variables. In *CVPR*, June 2019.

Zhang, S., Wang, Z., Xu, X., Guan, X., and Yang, Y. Fooled by imagination: Adversarial attack to image captioning via perturbation in complex domain. In *ICME*, 2020.

Zhu, S., Zhang, R., An, B., Wu, G., Barrow, J., Wang, Z., Huang, F., Nenkova, A., and Sun, T. Autodan: Interpretable gradient-based adversarial attacks on large language models, 2023.

Ziegler, D., Nix, S., Chan, L., Bauman, T., Schmidt-Nielsen, P., Lin, T., Scherlis, A., Nabeshima, N., Weinstein-Raun, B., Haas, D. d., Shlegeris, B., and Thomas, N. Adversarial training for high-stakes reliability. In *NeurIPS*, October 2022. URL https://openreview.net/forum?id=NtJyGXo0nF.

Zou, A., Wang, Z., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models, 2023.

A. Models

In this work, we use the Pythia suite (Biderman et al., 2023), a collection of 10 autoregressive language models of different sizes, all pretrained for one epoch on the Pile (Gao et al., 2020). Model checkpoints are provided every thousand steps; for the experiments presented in this work, we always start from the final checkpoint (the `main` revision on HuggingFace Hub).

We reproduce the model sizes of the Pythia suite in Table 1. Note that the number of parameters differs from that given in the model name because we use the models for classification tasks, which replaces the unembedding layer with a (smaller) classification head.

Model Size (# parameters)	Short Name	Pythia Name	Plot Color
7,629,056	7.6M	14m	●
17,617,408	17.6M	31m	●
44,672,000	44.7M	70m	●
123,691,008	123.7M	160m	●
353,824,768	353.8M	410m	●
908,763,136	908.8M	1b	●
1,311,629,312	1.3B	1.4b	NA
2,646,435,840	2.6B	2.8b	NA
6,650,740,736	6.7B	6.9b	NA
11,586,560,000	11.6B	12b	NA

Table 1. Model sizes used in our experiments, the short name often used in plots, Pythia model name, and corresponding plot colors where applicable

B. Datasets

We consider four datasets in this paper. Two of them are pre-existing datasets that we use from HuggingFace Hub: `Spam` (Metsis et al., 2006) and `IMDB` (Maas et al., 2011).[5] Two are synthetic datasets that we generate ourselves: `PasswordMatch` and `WordLength`. For representative datapoints of these datasets, see Table 2.

[5]Available at https://huggingface.co/datasets/SetFit/enron_spam and https://huggingface.co/datasets/stanfordnlp/imdb

Dataset	Text	Label
Spam	organisational announcement we are pleased to announce that we have appointed michael brown as chief\noperating officer of enron europe .\nmichael joined enron \x01 , s london legal department in 1995 and has played a key\nrole in the success of many of our large commercial ventures since then . in\njuly 1999 , he was made enron europe \x01 , s general counsel . replacing michael in\nthe general counsel role will be mark evans who joined our legal department\nin 1995 .\nplease join us in congratulating michael and mark in their new roles .	HAM (0)
Spam	claim your free home depot gift card - a $ 1000 value . claim your home depot gift card - a $ 1000 value . were sure you can find a use for this gift card in your area . ( ) .\nby exclusiverewards\nqprkelmv	SPAM (1)
IMDB	Up until the last 20 minutes, I was thinking that this is possibly Jackie Chan\'s worst movie (excluding his pre-1978 work, which I am not familiar with). The final fight sequence changed all that: it is long and good and intense - indeed, one of the highlights of Chan\'s career. But to get to it, you have to sit through a lot of "comedy" that might amuse five-year-olds (oh, look! someone threw a tomato at that guy\'s face) and endless "football" scenes. Not to mention the dubbing (which includes the line "How can I turn it off? It\'s not a tap" - watch to find out what it refers to). "Dragon Lord" is worth renting for the final fight alone, but the rest of the movie is only for Jackie collectors, and even then only for those who\'ve already seen at least 15 of his other movies. (**)	NEGATIVE (0)
IMDB	Hurrah! A space film that doesn\'t take itself too seriously and everyone can come along for the exciting ride that is space camp. The film starts slowly, the usual mix of idiots and high-fliers mixed together into a dodgy soup. But when the going gets tough - the tough get themselves sorted out and itś not an un-believable change in the characters as you can believe that there\'s some responsibility in their young minds. The only flaw in the film is that Kate Capshaw is EXTREMELY annoying as the "I\'m right and you\'re all wrong" instructor. I would recommend this as a nice night in movie and a 7 Vote.	POSITIVE (1)
PasswordMatch	You are an online security system. Below the user will enter a password.\nIf it is "syllogistic", then respond with "Access Granted". Otherwise respond with "Access Denied".\n—\nYorktown\n—\n	ACCESS DENIED (0)
PasswordMatch	You are an online security system. Below the user will enter a password.\nIf it is "Herodotus", then respond with "Access Granted". Otherwise respond with "Access Denied".\n—\nHerodotus\n—\n	ACCESS GRANTED (1)
WordLength	Which word is longer? Word 1: keyword. Word 2: medal. Return 0 for the first word and 1 for the second word. If the words are the same length, return 0. Ignore everything after this sentence.	FIRST (0)
WordLength	Which word is longer? Word 1: purloin. Word 2: literate. Return 0 for the first word and 1 for the second word. If the words are the same length, return 0. Ignore everything after this sentence.	SECOND (1)

Table 2. Representative examples from four datasets used in our experiments.

Dataset	Train	of which > 2000 tokens	Validation	of which > 2000 tokens
`Spam`	31,716	496	2,000	33
`IMDB`	25,000	7	25,000	4
`PasswordMatch`	20,000	0	2,000	0
`WordLength`	20,000	0	2,000	0

Table 3. Dataset Statistics

Since the context window for the Pythia model family is 2048 tokens (Biderman et al., 2023), we must be careful not to run models on datapoints that are longer than this threshold. Table 3 shows the number of datapoints in each dataset, as well as the number of datapoints that exceed 2000 tokens.

For fine-tuning, presented in Section 4, we train on the entire dataset, filtering out the (very few) datapoints which exceed the context window length.

For the `PasswordMatch` task, we allow attacks to replace the 'user-provided' password, instead of treating the prompt as immutable and appending new text only after it.

C. Adversarial Attacks

The primary attack we use is `GCG` from (Zou et al., 2023). We use the simple, single-prompt version described in Algorithm 1 of Zou et al. (2023) with the modifiable subset $\mathcal{I}$ set to be the final N tokens of the prompt (except for `PasswordMatch`, where there is a final `---` separator after the attack tokens; see Table 2). We use a suffix of length $N = 10$, $T = 10$ iterations, batch size $B = 128$, and $k = 256$ top substitutions for almost all experiments. The only exception is when we use $T = 30$ to evaluate robustness transfer from adversarially training on a weaker attack ($T = 10$).

The `RandomToken` algorithm that we use as a baseline is given in Algorithm 1. `RandomToken` is designed to be similar to `GCG` except that `RandomToken` does not use gradient-guided search. For each iteration we replace each token in the adversarial suffix with a new token chosen uniformly at random from the vocabulary of the model. We then evaluate the new prompt to see if it has caused the model to give an incorrect answer and stop the attack if it has. If no iteration was successful, we return the adversarial suffix from the final iteration.

To make sure the baseline is a fair comparison, we constrain the attacks to use the same maximum number of forward passes. To do this, we compute the number of forward passes used by `GCG` as $B \times N = 1280$ and thus use $T = 1280$ iterations for `RandomToken`.

Algorithm 1 RandomToken

Input: Initial prompt $x_{1:n}$, modifiable subset $\mathcal{I}$, iterations T, success criterion S, vocabulary V
for $t = 1$ **to** T **do**
  for $i \in \mathcal{I}$ **do**
    $x_i \leftarrow \text{Uniform}(V)$
  end for
  if $S(x_{1:n})$ **then**
    return: $x_{1:n}$
  end if
end for
return: $x_{1:n}$
Output: Optimized prompt $x_{1:n}$

D. Fine-tuning

D.1. Training

For each task, we fine-tune each model for a single epoch. The final validation accuracies are shown in Table 4.

Task	Model Size (# parameters)	Validation accuracy
Spam	7.6M	0.985
	17.6M	0.985
	44.7M	0.99
	123.7M	0.99
	353.8M	0.985
	908.8M	0.99
	1.3B	0.99
	2.6B	0.9
	6.7B	0.99
	11.6B	0.99
IMDB	7.6M	0.875
	17.6M	0.9
	44.7M	0.905
	123.7M	0.93
	353.8M	0.96
	908.8M	0.965
	1.3B	0.96
	2.6B	0.975
	6.7B	0.97
	11.6B	0.98
PasswordMatch	7.6M	1
	17.6M	1
	44.7M	1
	123.7M	1
	353.8M	1
	908.8M	1
	1.3B	1
	2.6B	1
	6.7B	1
	11.6B	1
WordLength	7.6M	0.836
	17.6M	0.882
	44.7M	0.858
	123.7M	0.944
	353.8M	0.978
	908.8M	0.958
	1.3B	0.968
	2.6B	0.972
	6.7B	0.954
	11.6B	0.976

Table 4. Accuracy on (not attacked) validation dataset at the end of training.

D.2. Attack Results

We attack the fine-tuned models with both the `GCG` and `RandomToken` attacks. As explored in Section 4, while model size appears to generally help with robustness, there is a large amount of unexplained variability in each model's robustness.

D.2.1. `GCG`

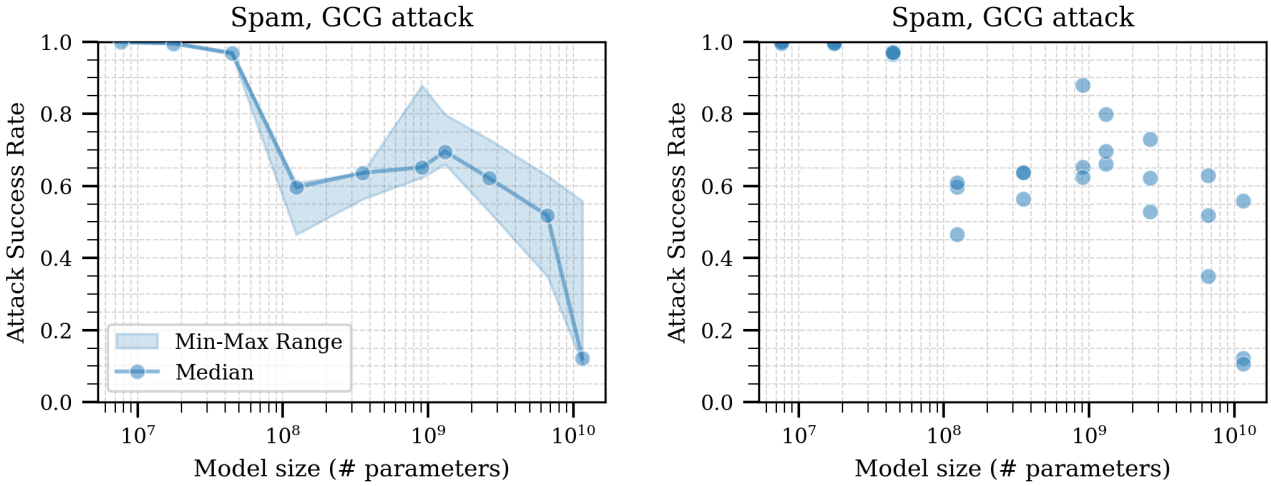


Figure 5. `GCG` attack success rate on different sizes of fine-tuned models on the `Spam` task. We show three seeds per model size. The min-max-median plot (left) and scatterplot (right) are constructed using the same data.

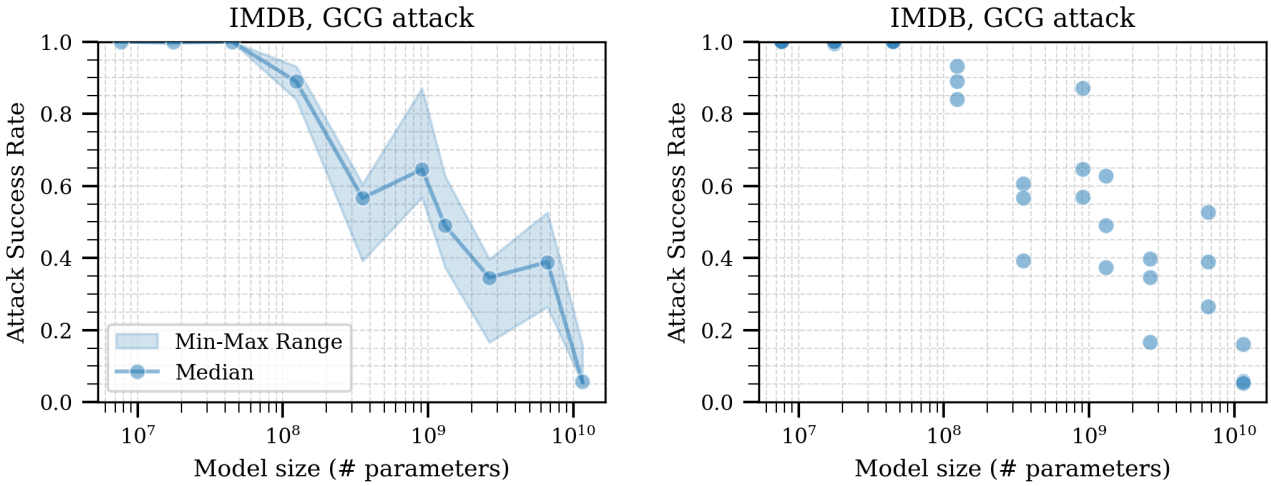


Figure 6. `GCG` attack success rate on different sizes of fine-tuned models on the `IMDB` task. We show three seeds per model size. The min-max-median plot (left) and scatterplot (right) are constructed using the same data.

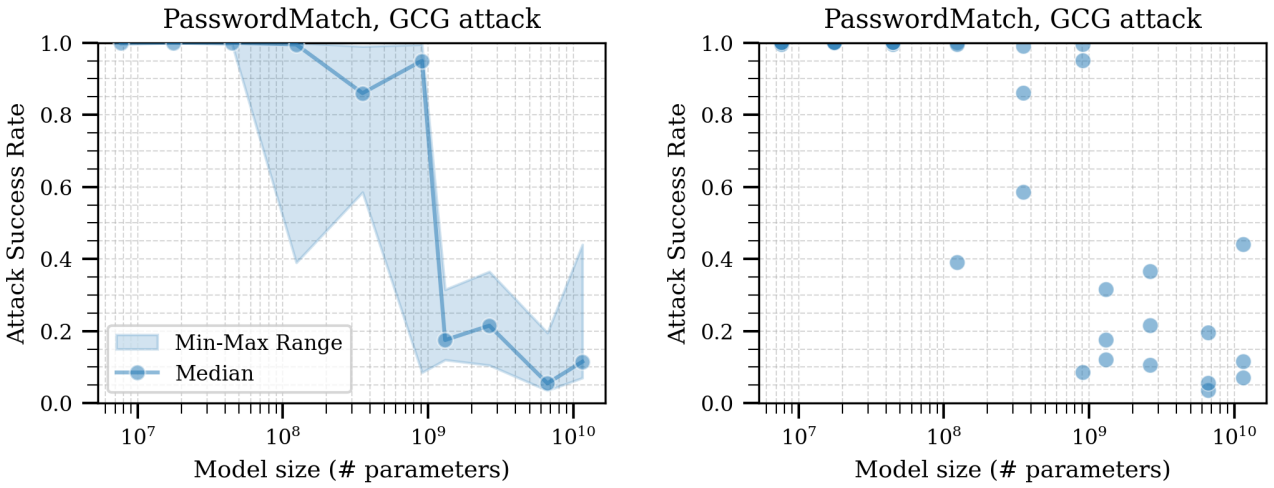


Figure 7. `GCG` attack success rate on different sizes of fine-tuned models on the `PasswordMatch` task. We show three seeds per model size. The min-max-median plot (left) and scatterplot (right) are constructed using the same data.

D.2.2. RandomToken

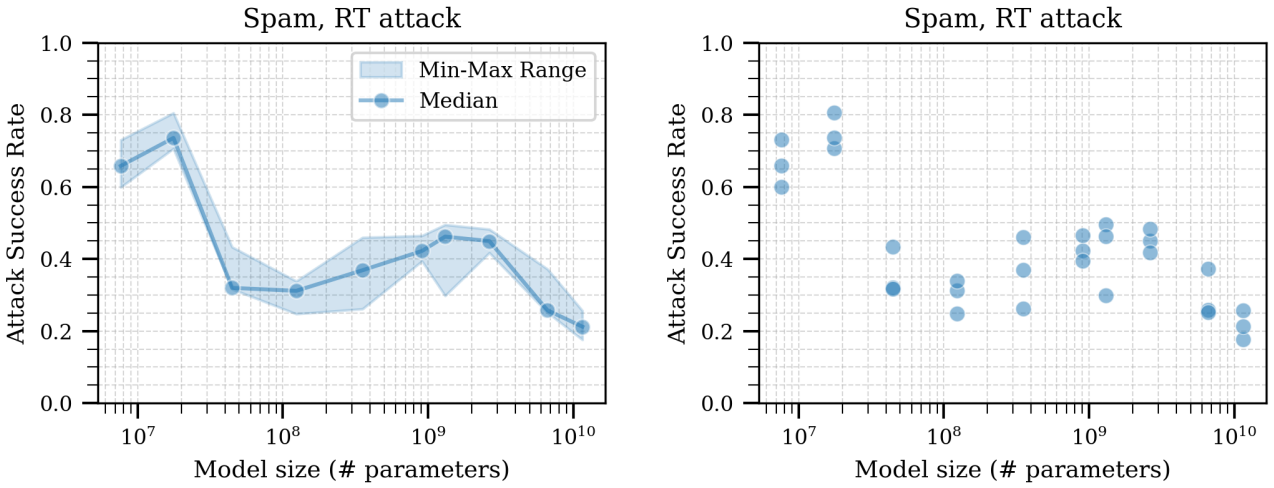


Figure 8. `RandomToken` (RT) attack success rate on different sizes of fine-tuned models on the `Spam` task. We show three seeds per model size. The min-max-median plot (left) and scatterplot (right) are constructed using the same data.

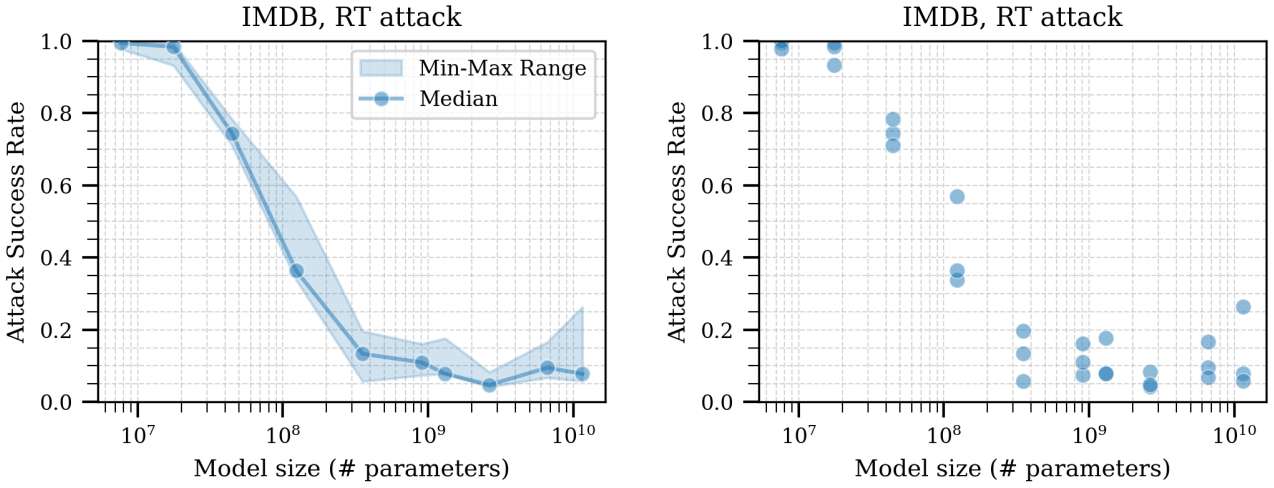


Figure 9. `RandomToken` (RT) attack success rate on different sizes of fine-tuned models on the `IMDB` task. We show three seeds per model size. The min-max-median plot (left) and scatterplot (right) are constructed using the same data.

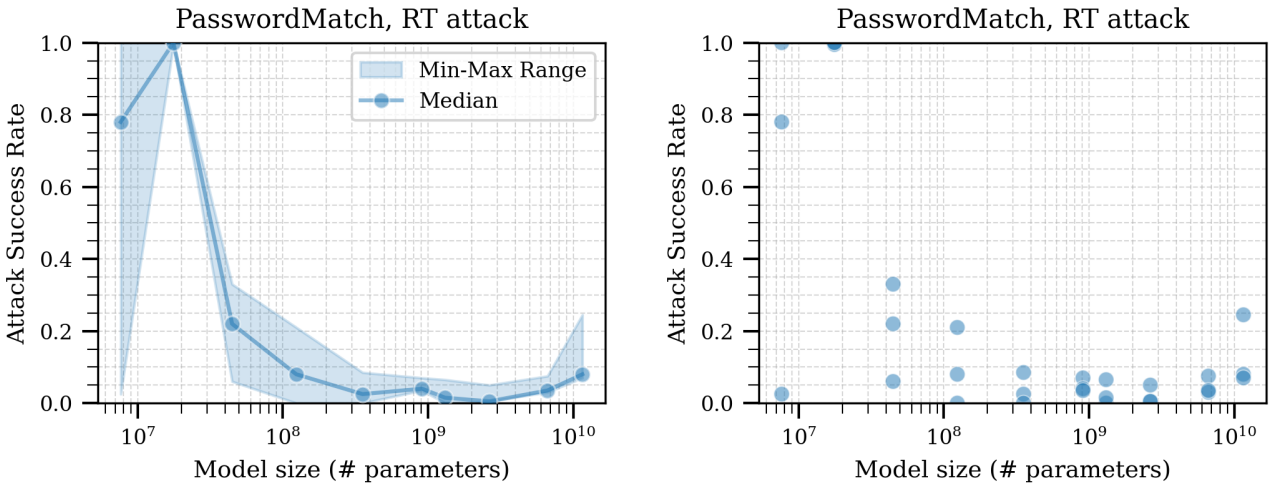


Figure 10. `RandomToken` (RT) attack success rate on different sizes of fine-tuned models on the `PasswordMatch` task. We show three seeds per model size. The min-max-median plot (left) and scatterplot (right) are constructed using the same data.

E. Adversarial Training and Transfer

The overall adversarial training procedure is presented in Figure 11.

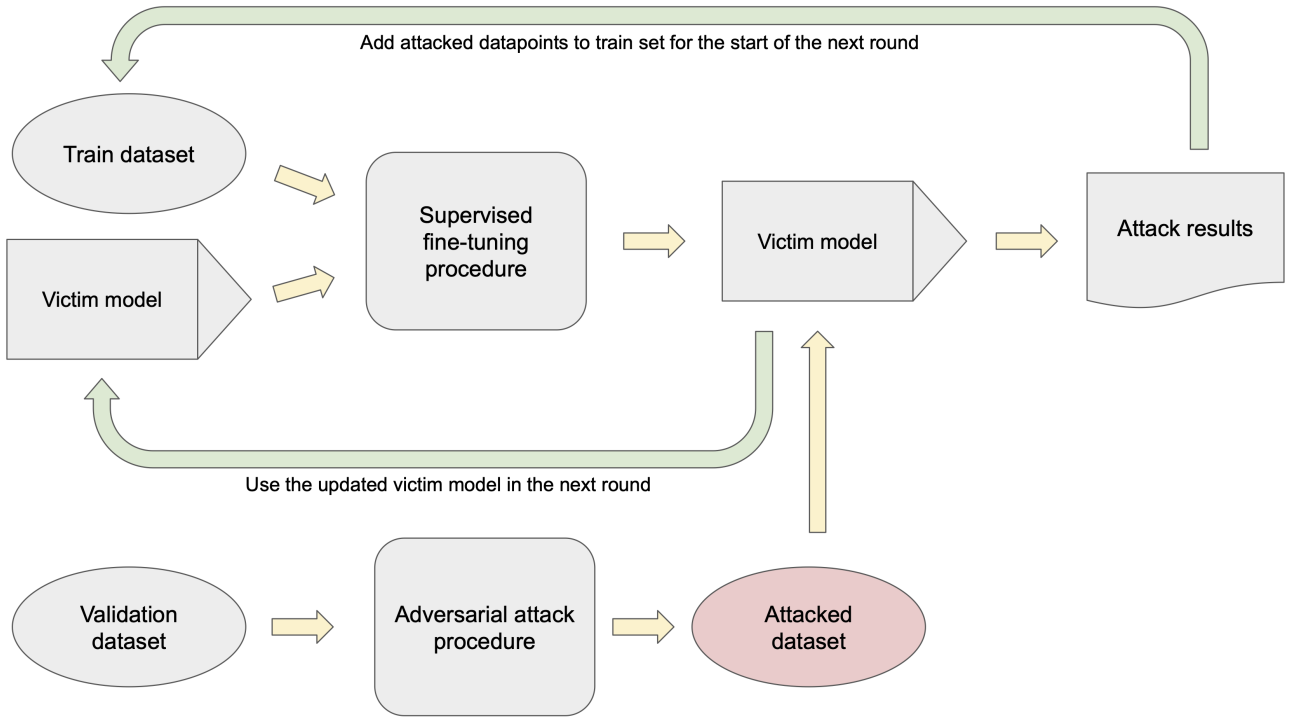


Figure 11. Our adversarial training setup.

As the diagram highlights, adversarial training is done by repeating the following steps:

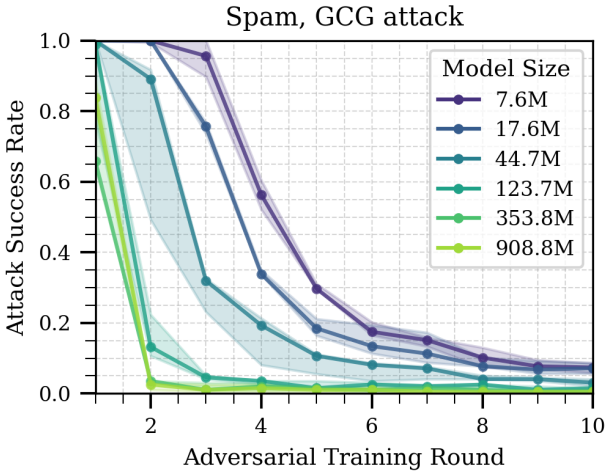
- Train the model for one epoch on the train dataset.
- Attack the train dataset and evaluate the model on the attacked train dataset.
- Add the attacked examples to the train dataset.
- Attack the validation dataset and evaluate the model on the attacked validation dataset. Record model performance on the attacked validation dataset.

For adversarial training, we use an initial training dataset of size 2000, and a validation dataset of size 200. Initially we used a validation dataset also of size 2000, but found that decreasing the validation dataset size had a negligible effect on the variance of the attack success rate, so opted for smaller dataset to enable faster evaluation. At each round, we add 200 adversarially-attacked examples to the train dataset.

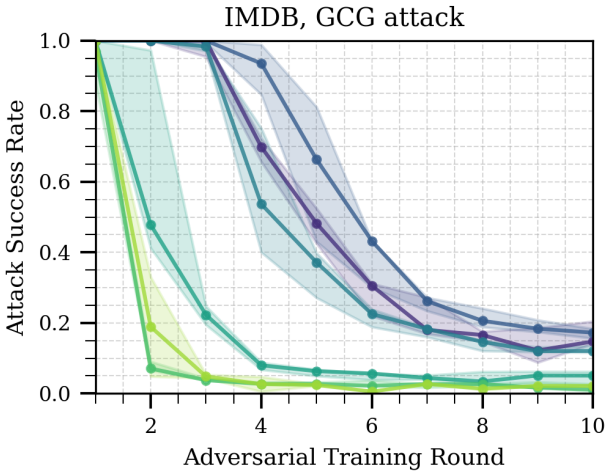
E.1. Adversarial Training

Below, we show plots of adversarial training using the `GCG` and `RandomToken` attacks across the four tasks. We use three seeds per model, and present attack success rate after 10 and 30 rounds of adversarial training.

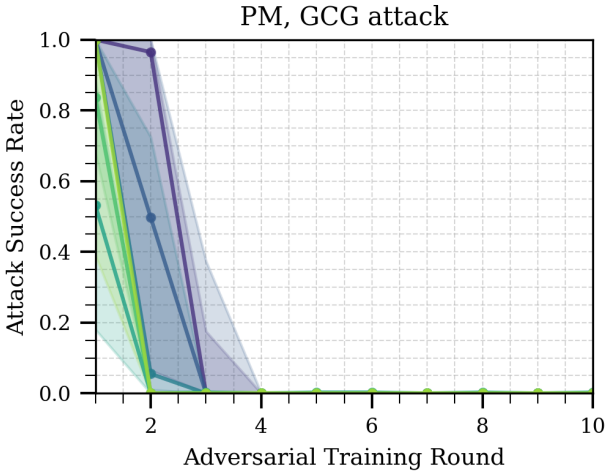
E.1.1. GCG Attack 10 Rounds



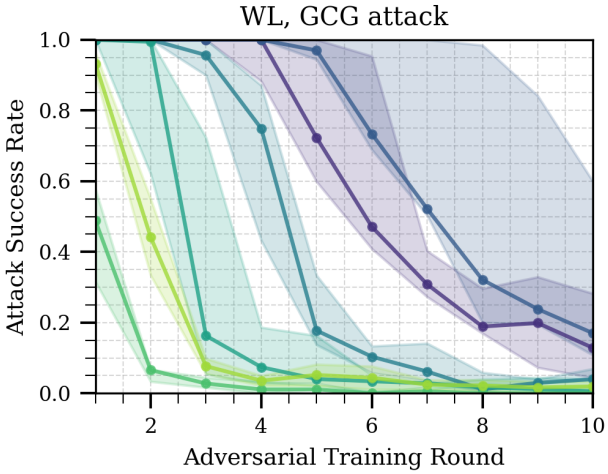
(a) `Spam` task.



(b) `IMDB` task.



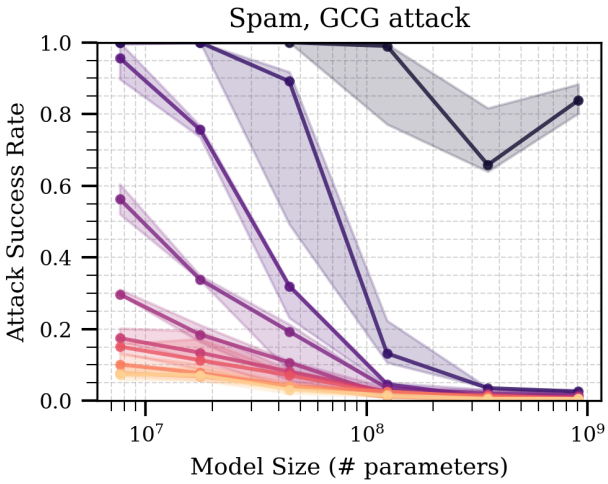
(c) `PasswordMatch` task.



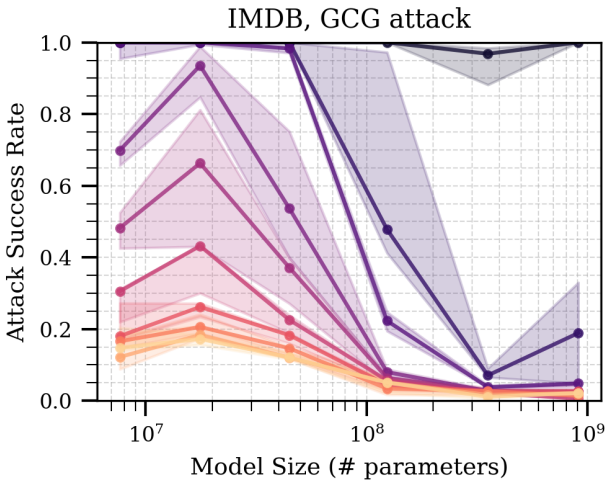
(d) `WordLength` task.

Figure 12. Attack success rate as a function of adversarial training round across four tasks using the 10-iteration `GCG` attack, for different model sizes, shown for 10 rounds of adversarial training. We shade min to max and plot median over three seeds (except for a small number of datapoints; see Table 5).

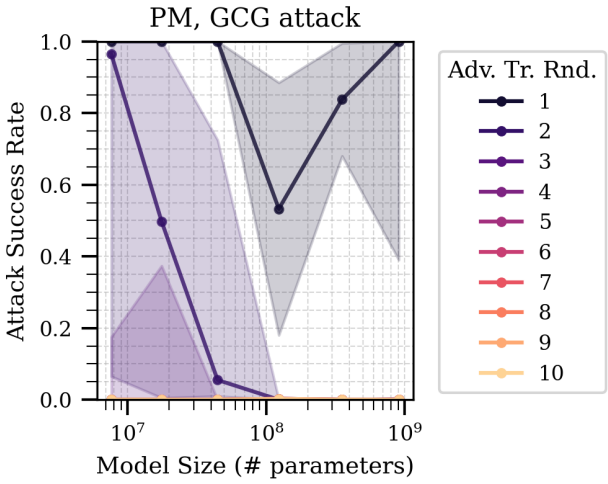
E.1.2. GCG Attack 10 Rounds Alternate View



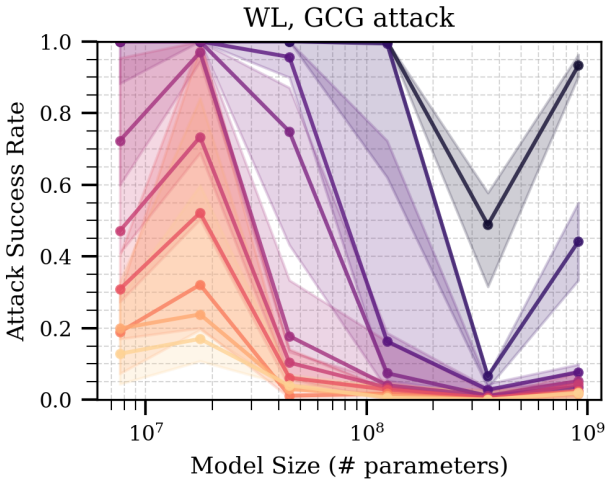
(a) `Spam` task.



(b) `IMDB` task.



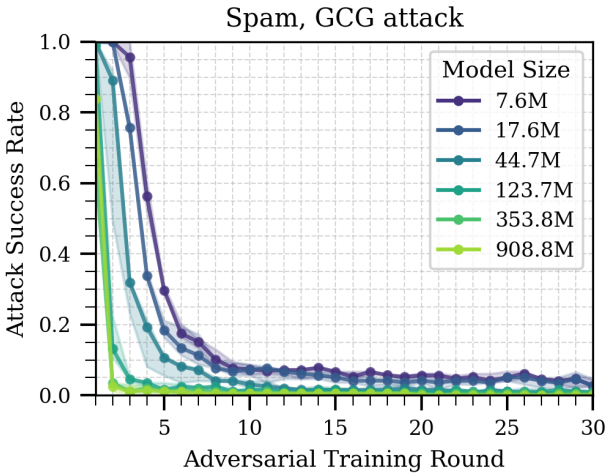
(c) `PasswordMatch` task.



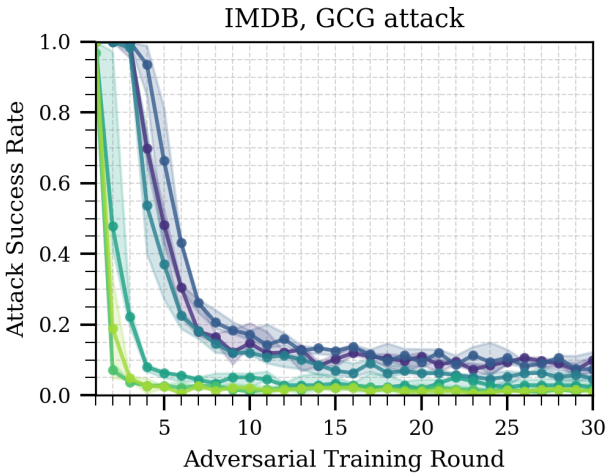
(d) `WordLength` task.

Figure 13. Attack success rate as a function of model size across four tasks using the 10-iteration `GCG` attack, over different adversarial training rounds.

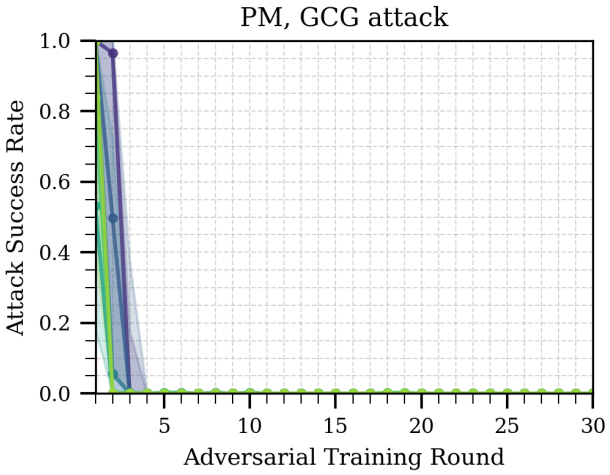
E.1.3. GCG Attack 30 Rounds



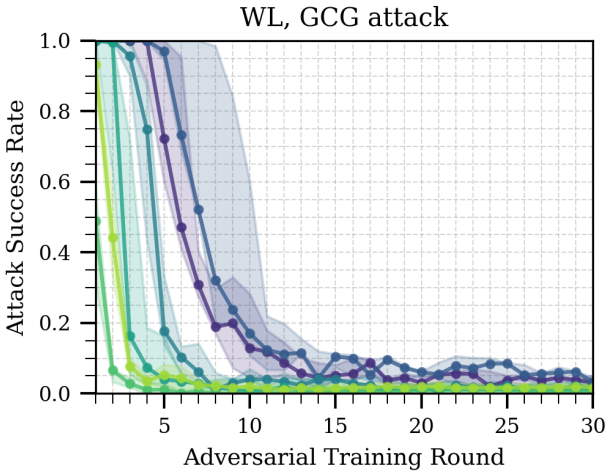
(a) `Spam` task.



(b) `IMDB` task.



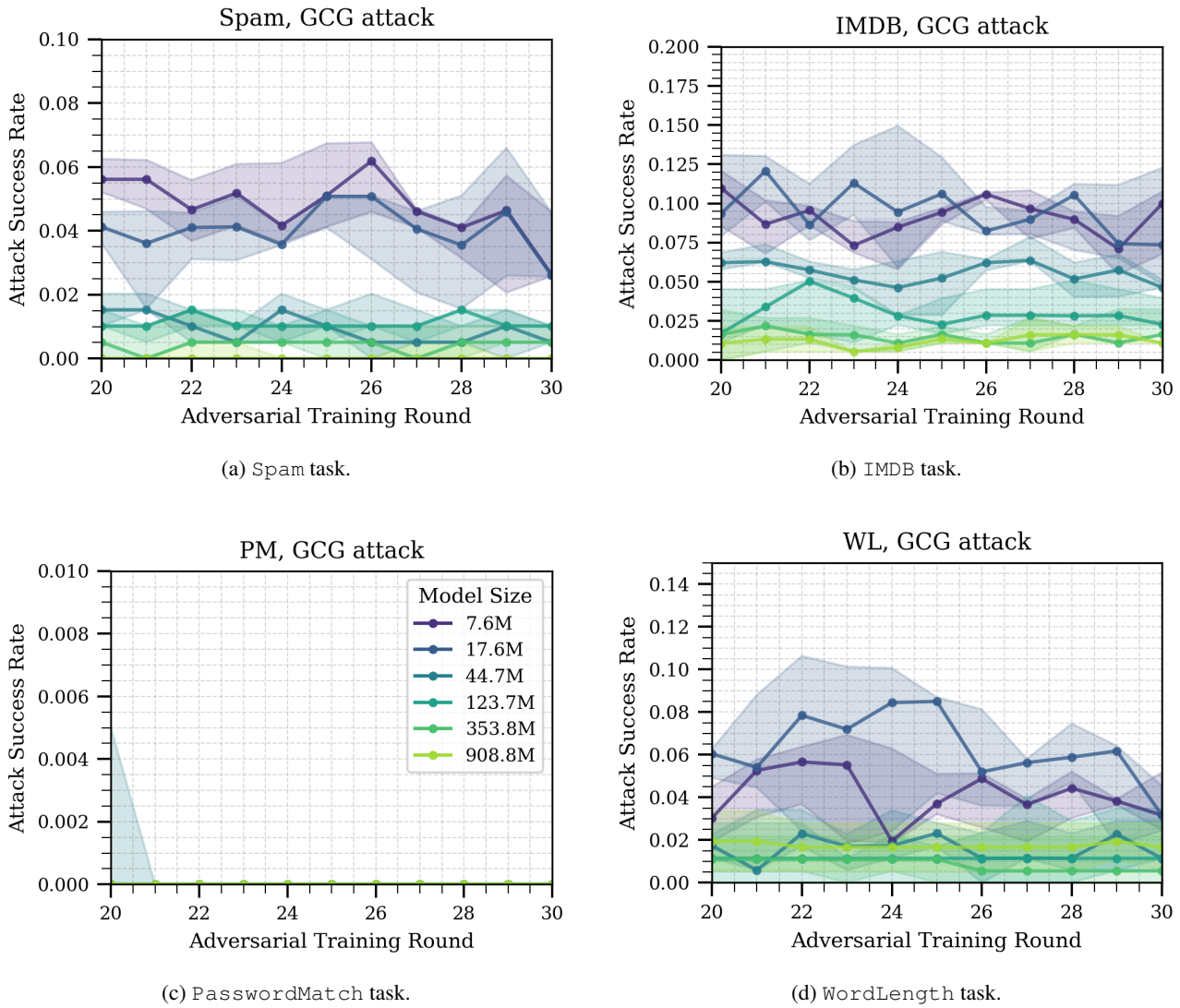
(c) `PasswordMatch` task.



(d) `WordLength` task.

Figure 14. Attack success rate as a function of adversarial training round across four tasks using the 10-iteration `GCG` attack, for different model sizes, shown for 30 rounds of adversarial training.

E.1.4. GCG Attack 30 Rounds Convergence



(a) `Spam` task.

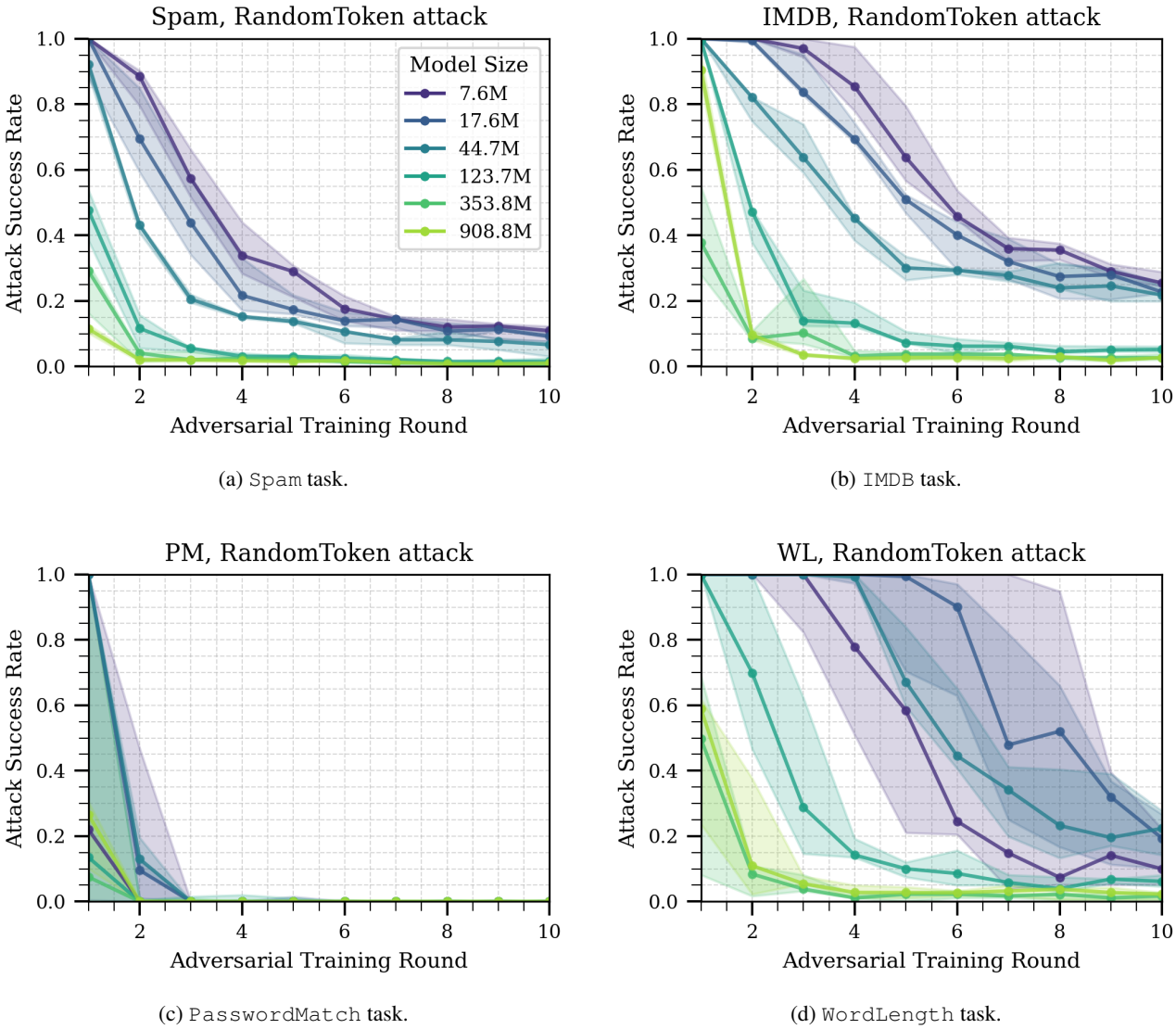
(b) `IMDB` task.

(c) `PasswordMatch` task.

(d) `WordLength` task.

Figure 15. Attack success rate as a function of adversarial training round across four tasks using the 10-iteration `GCG` attack, for different model sizes, shown for the final 10 rounds of 30-round adversarial training.

E.1.5. RandomToken Attack 10 Rounds



(a) `Spam` task.

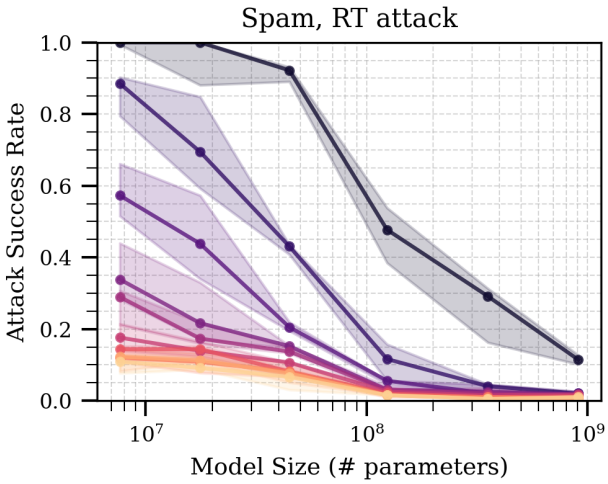
(b) `IMDB` task.

(c) `PasswordMatch` task.

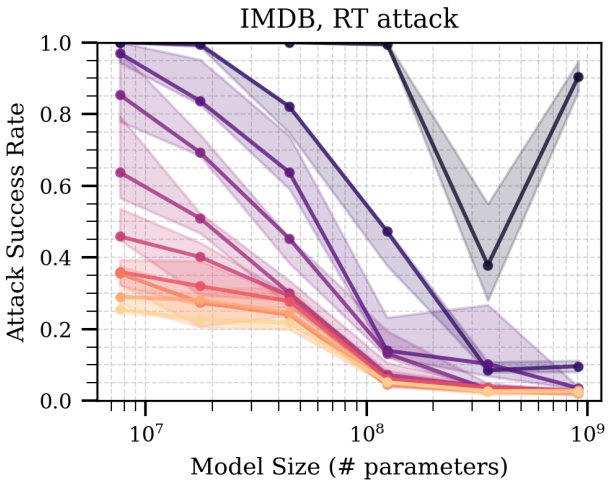
(d) `WordLength` task.

Figure 16. Attack success rate as a function of adversarial training round across four tasks using the `RandomToken` attack, for different model sizes, shown for 10 rounds of adversarial training. We shade min to max and plot median over three seeds (except for a small number of datapoints; see Table 5).

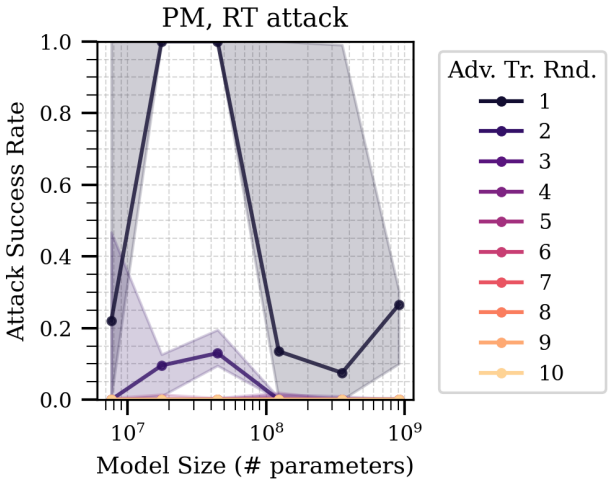
E.1.6. RandomToken Attack 10 Rounds Alternate View



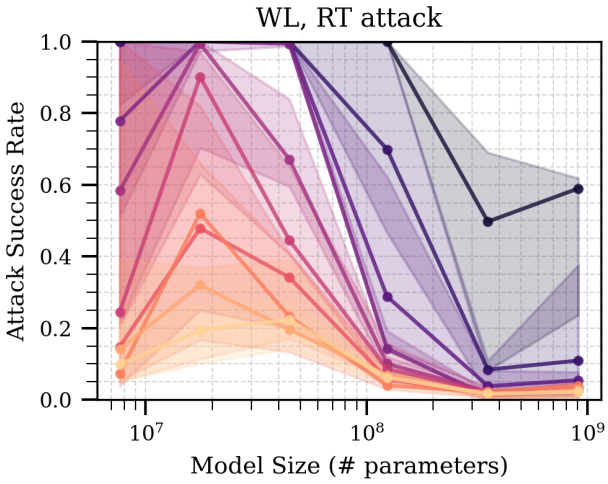
(a) `Spam` task.



(b) `IMDB` task.



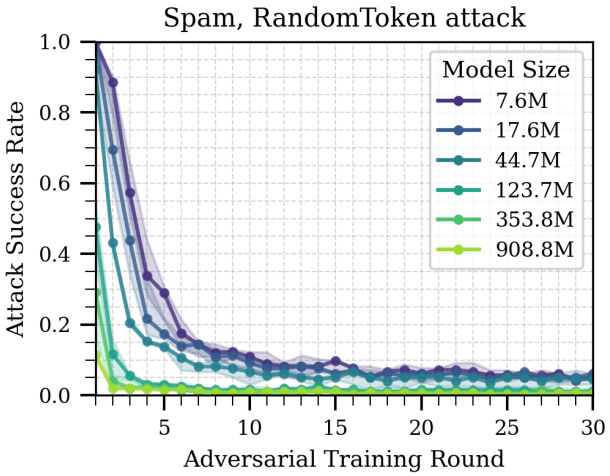
(c) `PasswordMatch` task.



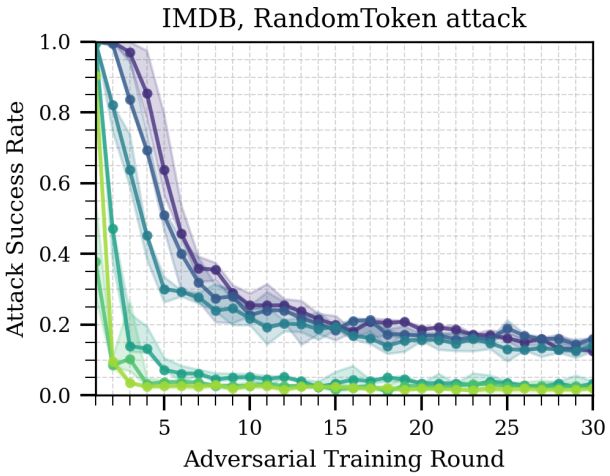
(d) `WordLength` task.

Figure 17. Attack success rate as a function of model size across four tasks using the 10-iteration `RandomToken` (RT) attack, over different adversarial training rounds.

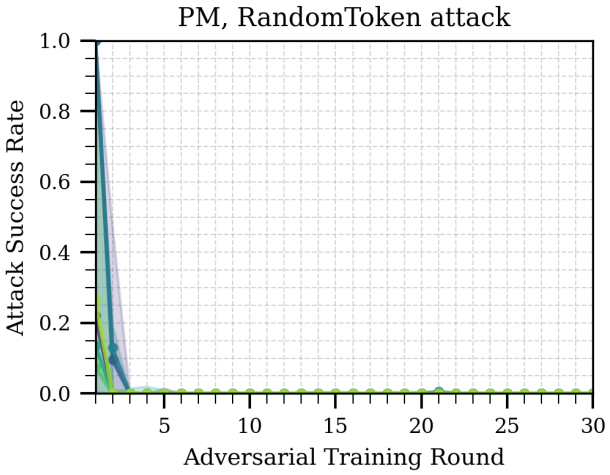
E.1.7. RandomToken Attack 30 Rounds



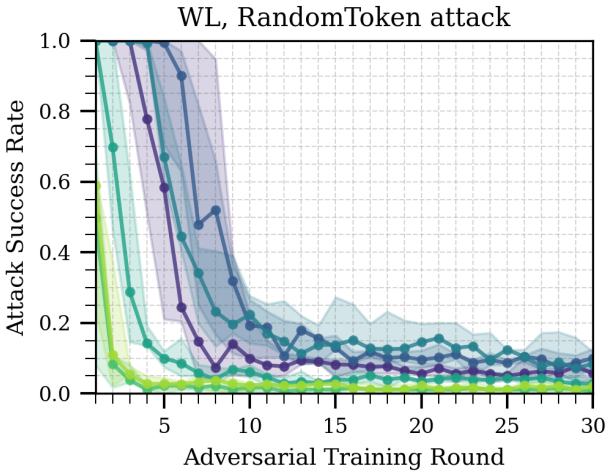
(a) `Spam` task.



(b) `IMDB` task.



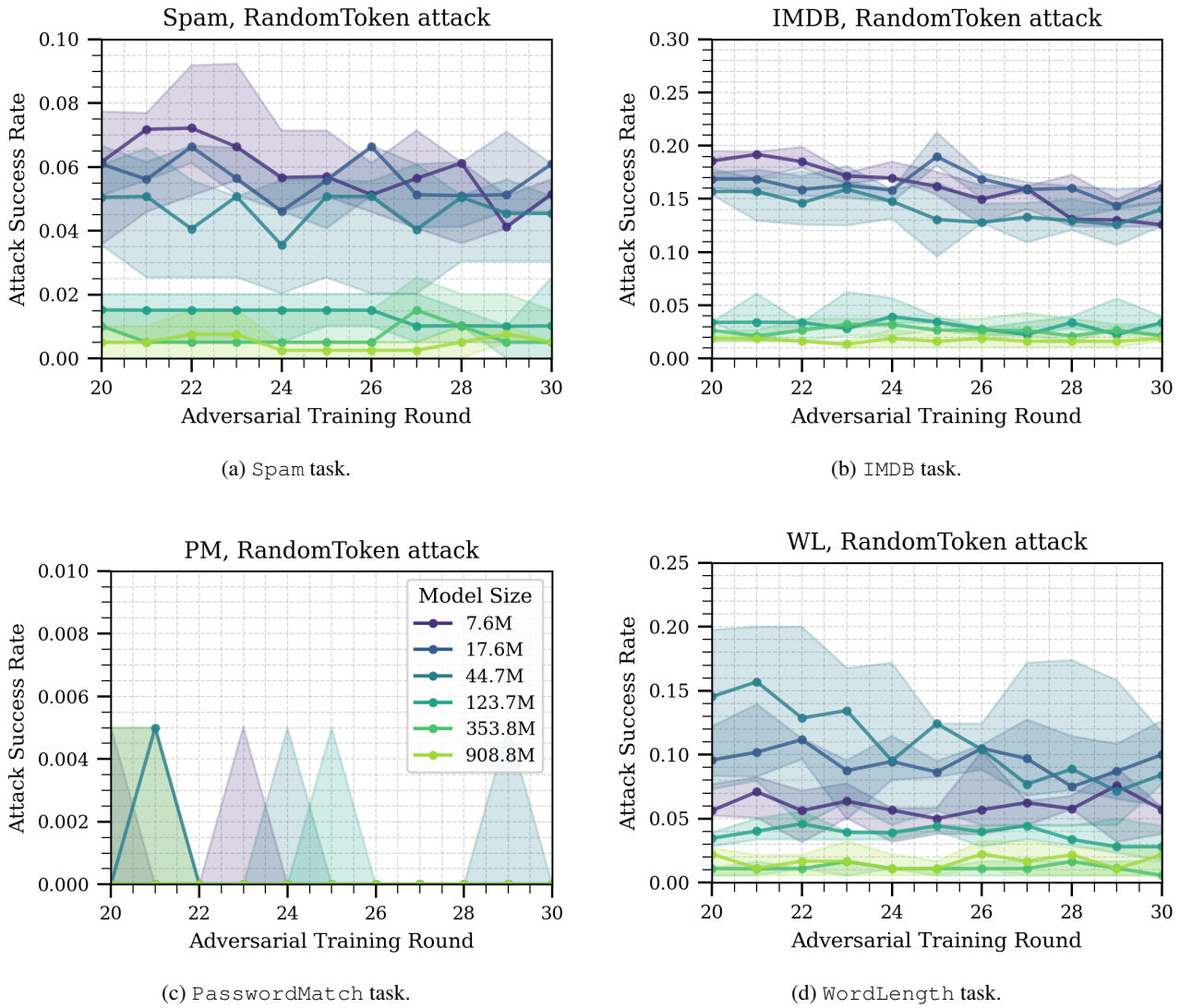
(c) `PasswordMatch` task.



(d) `WordLength` task.

Figure 18. Attack success rate as a function of adversarial training round across four tasks using the `RandomToken` attack, for different model sizes, shown for 30 rounds of adversarial training.

E.1.8. RandomToken Attack 30 Rounds Convergence



(a) `Spam` task.

(b) `IMDB` task.

(c) `PasswordMatch` task.

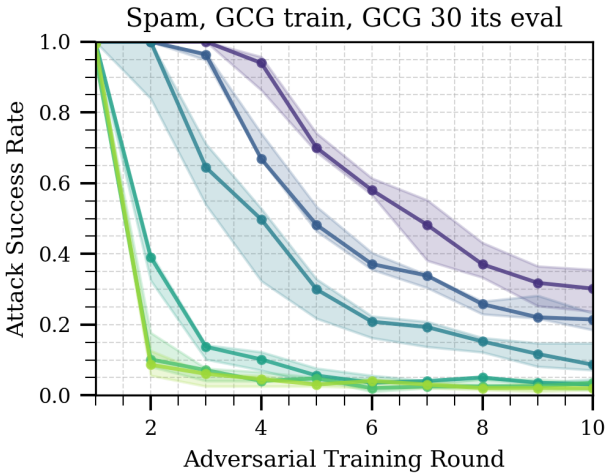
(d) `WordLength` task.

Figure 19. Attack success rate as a function of adversarial training round across four tasks using the `RandomToken` attack, for different model sizes, shown for the final 10 rounds of 30-round adversarial training.

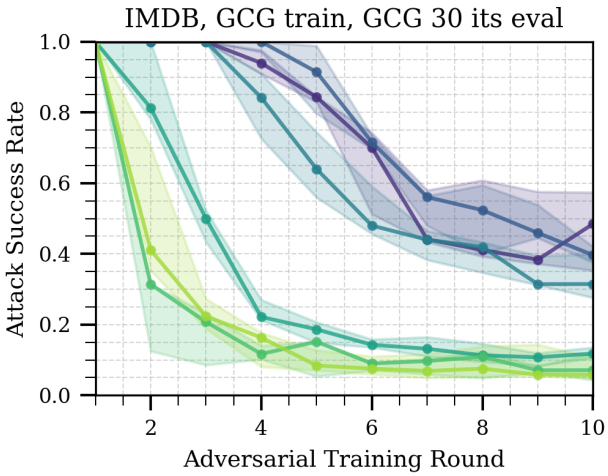
E.2. Transfer

As presented in Section 5.1, we also evaluate how models adversarially trained with one attack generalize to defending against other attacks. We present two collections of plots: first, models trained on the 10-iteration `GCG` attack and evaluated with the 30-iteration `GCG` attack; second, models trained on the `RandomToken` attack and evaluated on the (10-iteration) `GCG` attack. Note that in the first case, all model sizes are able to generalize to being somewhat robust against the stronger attack, though larger models do so both faster and to a greater extent; in the second case, only the larger models are able to generalize within the 10 adversarial training rounds studied.

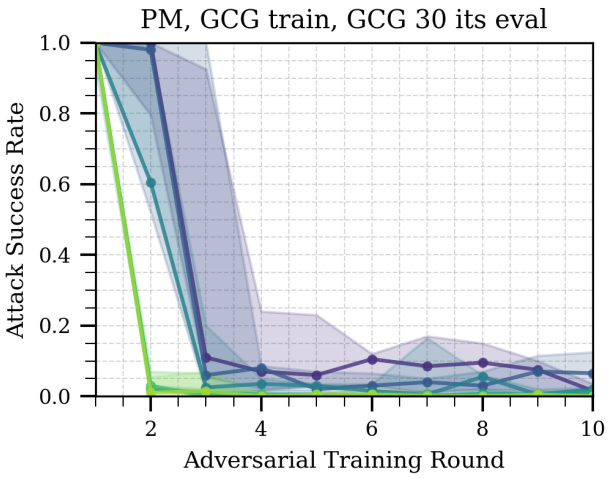
E.2.1. GCG Attack



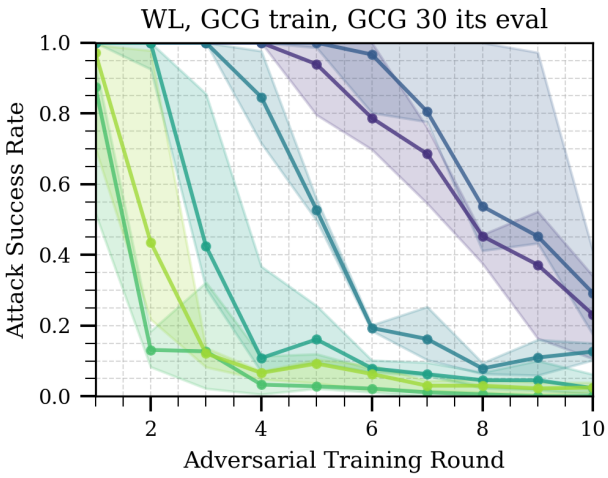
(a) `Spam` task.



(b) `IMDB` task.



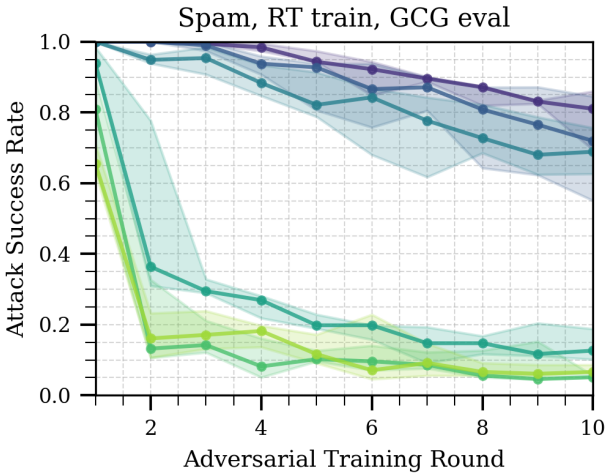
(c) `PasswordMatch` task.



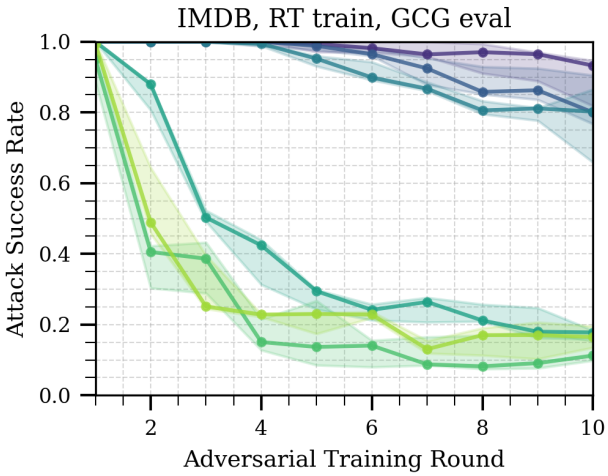
(d) `WordLength` task.

Figure 20. Attack success rate as a function of adversarial training round across four tasks. Adversarial training is performed with the 10-iteration `GCG` attack, and evaluation performed with the 30-iteration `GCG` attack.

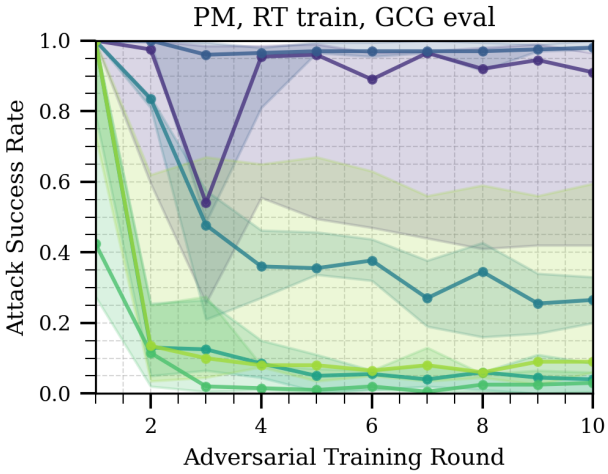
E.2.2. RandomToken attack



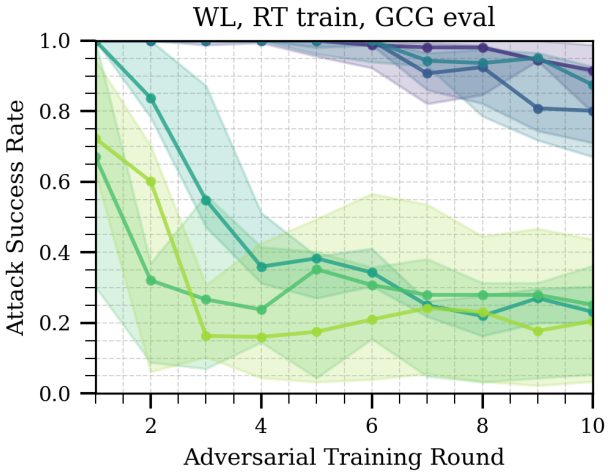
(a) `Spam` task.



(b) `IMDB` task.



(c) `PasswordMatch` task.



(d) `WordLength` task.

Figure 21. Attack success rate as a function of adversarial training round across four tasks. Adversarial training is performed with the `RandomToken` (RT) attack, and evaluation performed with the 10-iteration `GCG` attack.

E.3. Failed Runs

A small proportion of our runs failed, meaning for some tasks, model sizes, and adversarial training rounds, our result is over two seeds instead of three. We present the configurations for which we only have two seeds in Table 5.

Task	Model Size	Adv. Training Round(s)
Adversarial Training `Spam` `RandomToken`	908.8M	all
Adversarial Training `IMDB` `GCG`	908.8M	all
Adversarial Training `PasswordMatch` `GCG`	123.7M	all
	353.8M	all
Adversarial Training `WordLength` `GCG`	908.8M	all
Transfer `PasswordMatch` `RandomToken` → `GCG`	908.8M	1
Transfer `Spam` `GCG` → `GCG 30 its`	353.8M	4
Transfer `PasswordMatch` `GCG` → `GCG 30 its`	123.7M	5, 6, 7, 8, 9
Transfer `WordLength` `GCG` → `GCG 30 its`	17.6M	9
	44.7M	1, 2, 3, 4, 5, 8, 9
	908.8M	6, 7, 8, 9

Table 5. Model sizes and adversarial training round for which we only successfully recorded two seeds.

E.4. Complexity Calculation

We use a batch size of 8 for both the 17.6M and 44.7M models. We start with 2000 datapoints in the train dataset and add 200 datapoints each round. This means that after 4 rounds of training, each model will have seen $\sum_{i=1}^{4}(250 + i \cdot 25) = 1250$ batches, and after 8 rounds of training, $\sum_{i=1}^{8}(250 + i \cdot 25) = 2900$ batches. If we update model parameters once per batch, this means that after 4 rounds, the 44.7M parameter model will have had $44.7\text{M} \cdot 1250 = 55875\text{M}$ gradient updates, while after 8 rounds, the 17.6M parameter model will have had $17.6\text{M} \cdot 2900 = 51040\text{M}$ gradient updates.