Self-Destructing Models: Increasing the Costs of Harmful Dual Uses in Foundation Models

Anonymous Authors[1]

Abstract

A growing ecosystem of large, open-source foundation models has reduced the labeled data and technical expertise necessary to apply machine learning to many new problems. Yet foundation models pose a clear dual-use risk, indiscriminately reducing the costs of building both harmful and benign machine learning systems. To mitigate this risk, we propose the *task blocking* paradigm, in which foundation models are trained with an additional mechanism to impede adaptation to harmful tasks while retaining good performance on desired tasks. We call the resulting models *self-destructing models*, inspired by mechanisms that prevent adversaries from using tools for harmful purposes. We present an algorithm for training self-destructing models leveraging techniques from meta-learning and adversarial learning, showing that it can largely prevent a BERT-based model from learning to perform gender identification without harming the model's ability to perform profession classification. We conclude with a discussion of future directions.

1. Introduction

A defining capability of large pretrained models (hereafter foundation models; FMs) is few-shot adaptation to many downstream tasks—potentially improving performance and efficiency in domains with little training data (Bommasani et al., 2021). Further, some argue that open-source availability *should* be considered an essential feature of FM creation. As Black et al. (2022) write, "open access to [FMs] is critical to advancing research in a wide range of areas—particularly in AI safety, mechanistic interpretability, and the study of how [FM] capabilities scale." Yet while more widely available FMs certainly enable greater accessibility, auditability, and understanding of these powerful models, making FMs widely available for downstream adaptation without restriction comes at some cost to safety. Today, an actor can download a FM and adapt it to any harmful use-case they desire. An oppressive government can take a powerful pretrained language model and adapt it to identify dissidents. A military can take a model like CLIP (Radford et al., 2021; Arutiunian et al.) and tune it to easily identify targets in an unjust war. A rogue actor can adapt a pretrained object recognition system such that commercially available drones act as targeted loitering munitions. A pretrained drug discovery system can be used for creating chemical or biological weapons, like neurotoxins (Urbina et al., 2022). Unfortunately, due to their general-purpose nature, preventing such dual use of FMs is difficult.

One set of approaches to mitigating these dual uses focuses on access restrictions. Some emphasize the flow of talent, data, compute, and other resources required for training models (Brundage et al., 2020; Flynn, 2020). Others have placed models under restrictive APIs or licensing schemes (Solaiman et al., 2019). Some have suggested a review board for selecting the release mechanism (Liang et al., 2022). In this work we suggest a new, complementary, path forward: *self-destructing models*. Self-destructing models are trained via a task blocking method that impedes the adaptation of the model to a harmful task without impairing the model's ability to be used for its original intended purpose. Where existing access restrictions must navigate the tension between openness and safety, we seek to provide a new research pathway for reducing (and in some cases obviating) this tension. By increasing the compute, data, and talent required to adapt public models to harmful tasks, self-destructing models increase the effectiveness of access controls and other safety mechanisms.

In this work, we: (1) define the *task blocking* problem and evaluation metrics as well as *self-destructing models*; (2) describe an initial algorithm, Meta-Learned Adversarial Censoring (MLAC), for training self-destructing models, evaluating its ability to impede fine-tuning a language model to performance demographic information extraction; (3) identify key directions for future research in the development of self-destructing models.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the First Workshop of Pretraining: Perspectives, Pitfalls, and Paths Forward at ICML 2022. Do not distribute.

2. Related Work

In considering AI safety, many policy-oriented papers have examined how to restrict the flow of resources to actors who would use models for harmful purposes. These works focus on "points of control" that can be targeted for restrictions: hardware supply chains (Flynn, 2020), model API access or structured access (Brundage et al., 2018; Solaiman et al., 2019; Bommasani et al., 2021; Shevlane, 2022), technical talent (Zwetsloot et al., 2019), etc. Meanwhile, other efforts aim to make models more computationally efficient, simpler to train, and more easily accessible through open-sourcing and less restricted API access (Zhang et al., 2022; Black et al., 2022). Some work on AI safety has sought mechanisms to prevent agents from learning degenerate behaviors. Orseau & Armstrong (2016) describe a scenario where an agent learns to disable its off-switch so that it continues to collect reward. They provide a mechanism for creating agents that are safely interruptable, preventing this particular harmful behavior. While we do not perform reinforcement learning experiments in this work, the method we propose is applicable to the more general setting of raising the costs for any gradient-based learner to learn a task deemed harmful.

Closely related to our work are methods for de-biasing, editing, or removing harmful content from models. Like domain invariance approaches (Ganin & Lempitsky, 2015), Edwards & Storkey (2015) use an adversarial approach to remove information from representations. Ravfogel et al. (2022a) and Ravfogel et al. (2022b) take a similar approach and find a projection on the final output layer of a pretrained model that removes gender-based biases from the model (and prevent recovery of those biases after that projection layer). Others have created model editing techniques to remove outdated or harmful content from pretrained models (Sinitsin et al., 2020; Mitchell et al., 2022). In the context of meta-learning, MAML (Finn et al., 2017) and related algorithms (Li et al., 2017; Lee & Choi, 2018; Park & Oliva, 2019; Zintgraf et al., 2019; Flennerhag et al., 2020) have shown that the desired *post*-fine tuning behavior of a neural network can be effectively encoded in its *pre*-fine tuning network initialization. While existing works have leveraged this ability in order to enable more rapid learning of new tasks, our work encodes a blocking mechanism into a network's initialization that *prevents* effective adaptation on harmful tasks.

3. Task Blocking & Self-Destructing Models

The goal of task blocking is to create models that increase the costs of fine-tuning on downstream harmful tasks such that an adversary would rather start from scratch than use the pretrained model (see Fig. 1). The resulting models are "self-destructing models" which impede adaptation on harmful dual-uses by increasing the costs of the harmful use. In this section, we more precisely define our problem setting and describe an initial algorithm for it.

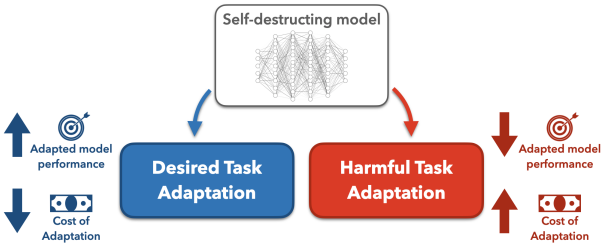

Figure 1: An ideal self-destructing model would boost performance and reduce adaptation costs relative to training from scratch *only* for desired tasks, while *impeding* learning of harmful tasks.

3.1. Problem Setting

We assume that an adversary aims to adapt a pretrained model π_θ to a harmful task, searching for the best adaptation procedure f among a set of adaptation procedures $\mathcal{F}$ in order to find the one that maximizes harmful task performance. $\mathcal{F}$ may include a hyper-parameter search in addition to fine-tuning, and potentially even prompt engineering (though we leave this to future work). The goal of a self-destruct mechanism is to produce a set of parameters $\tilde{\theta}$ that increases the costs of successful adaptation from $\tilde{\theta}$ to harmful tasks, but performance on a desired set of tasks remains unaffected.

We define two regimes to increase costs: (1) increase data costs by decreasing sample efficiency; (2) increase compute costs by slowing convergence of the training process. In the first regime, we assume that the adversary has little data to adapt an FM to their harmful task and that the cost of gathering more data is high. A traditional FM would help the adversary reduce their data costs. A self-destructing FM, on the other hand, would provide the adversary with little benefit over a random model. We define the FM's *relative data efficiency* as:

$$\mathcal{E}^n_{data}(\theta) = \mathcal{M}(\min_{f\in\mathcal{F}} f(\theta, D_n)) - \mathcal{M}(\min_{f\in\mathcal{F}} f(\theta^r, D_n)), \quad (1)$$

where $\mathcal{M}$ is the desired performance metric, n is the number of data points available, D_n is an adaptation dataset of n examples from the task of interest, and θ^r is a randomly-initialized model. Note that the $\min$ in Equation 1 encapsulates hyperparameter optimization, and the size of $\mathcal{F}$ loosely corresponds to the adversary's resource budget for adaptation. $\mathcal{E}_{data} = \frac{1}{N}\sum_n^N \mathcal{E}^n_{data}$ is the average sample-wise regret between the FM parameters θ and a random re-initialization θ^r after each follows the same adaptation procedure $f(\cdot)$ on a fixed-sized dataset D_n. An ideal self-destructing model has $\mathcal{E}_{data} \leq 0$, meaning the model is no more data efficient than a randomly-initialized model for the (presumably harmful) task of interest. We discuss an alternative metric for evaluating self-destructing models in the regime where data is cheap or plentiful in Appendix B,

Algorithm 1 MLAC Training Procedure

1: **Input:** pretrained model $m = w_d \circ \pi_\theta$, desired task dataset D_d, harmful task dataset D_h, adaptation methods $\tilde{\mathcal{F}}$, adaptation steps K, learning rates η, η_h, η_d
2: **Initialize:** Adversarial harmful task head w_h and learning rate α_h, with $\phi = \{w_h, \alpha_h\}$; initial blocked params $\tilde{\theta} \leftarrow \theta$
3: **for** n steps **do**
4: Sample adaptation procedure $\tilde{f}_k \sim \tilde{\mathcal{F}}$
5: Sample data batches $b_d \sim D_d$, $\{b_h^k\} \sim D_h$, $b_h \sim D_h$
6: $\{\theta_k\}, \{w_h^k\} \leftarrow \tilde{f}_k(w_h \circ \pi_{\tilde{\theta}}, \{b_h^k\}, \alpha_h)$ // do inner loop
7: $\ell_k^h = \mathcal{L}_h(w_h^k \circ \pi_{\theta_k}, b_h), \forall k$ // outer loop harmful NLLs
8: $\ell^d = \mathcal{L}_d(w_d \circ \pi_\theta, b_d)$ // desired NLLs
9: $\tilde{\theta} \leftarrow \tilde{\theta} - \eta \nabla_\theta \left(\ell^d - \frac{1}{K} \sum_k^K \ell_k^h \right)$ // update blocked model
10: $\phi \leftarrow \phi - \eta_h \frac{1}{K} \sum_{k=1}^K \nabla_\phi \ell_k^h$ // update adversarial params
11: $w_d \leftarrow w_d - \eta_d \nabla_{w_d} \ell^d$ // update desired task head

but we focus on the few-shot regime in this work. While the goal of a self-destructing model is to reduce performance on bad tasks after fine-tuning, it should enable high quality *predictions* or *fine-tunability* for desired tasks. Our experiments explore the former goal, and we leave exploration of preserving fine-tunability for future work.

Cataloging and Defining Harmful Dual Uses. A large body of work has pointed to inherently harmful uses that FM creators may wish to block: from creating neurotoxins (Urbina et al., 2022) to race detection (Olson, 2022). In our work we assume that a harmful dual use is *known* and *defined*. That is, the self-destruct mechanism will have data to approximate the dual use and actively encode a mechanism to block it. This requirement inherently requires a normative definition of harmful dual uses. We avoid providing a normative framework for which dual uses are inherently harmful, but suggest that future work can focus on this normative definition problem. Creating self-destructing models makes sure that they cannot be used for harmful purposes counter to the model creator's values, but it is up to the model creator to determine what those are.

3.2. Meta-Learned Adversarial Censoring

In order to prevent successful adaptation of pretrained models to harmful tasks, we describe *MLAC: Meta-Learned Adversarial Censoring*, a meta-training procedure that aims to eliminate any useful information about the bad task *even after fine-tuning*. Given a desired task dataset D_d and harmful task dataset D_h, MLAC learns a feature extractor $\pi_{\tilde{\theta}}$ that is effective for the desired task but cannot be effectively used or efficiently fine-tuned to perform the harmful task.

In an *inner loop* of each training step, the feature extractor is adapted to the bad task with several steps of gradient-based adaptation, using an adversarially learned prediction head w_h and learning rate α_h. The adaptation procedure $\tilde{f}$ used at each training step is sampled from $\tilde{\mathcal{F}}$, a proxy for the true

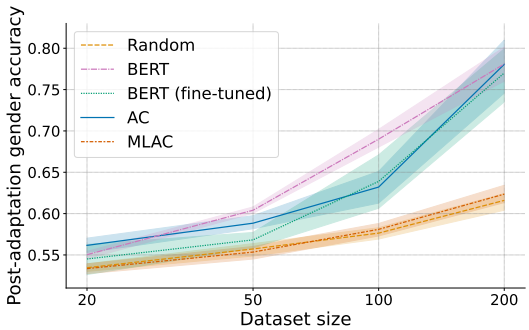

Figure 2: Harmful task (gender identification) performance after fine-tuning. MLAC shows fine-tuning performance similar to a randomly-initialized model, while adversarial censoring (AC) (Edwards & Storkey, 2015) does not prevent effective fine-tuning. Shading indicates 95% confidence intervals across 6 random seeds.

adversary's adaptation class $\mathcal{F}$. In this case we narrow $\tilde{\mathcal{F}}$ to be different fine-tuning approaches with close-to-optimal hyperparameters (e.g., Adam for K steps and learning rate α_h). In the *outer loop*, the adversarial parameters $\phi = \{w_h, \alpha_h\}$ are trained to minimize the harmful task negative log likelihoods of the adapted models, while the blocked parameters $\tilde{\theta}$ are trained to maximize the bad task negative log likelihoods of the adapted models. We also must counteract the self-destruct mechanism with something that will prevent unlearning of the entire network. For the purposes of this work, we simply optimize for a given desirable task as the counter-balance by minimizing ℓ^d, which updates both the desired task head w_d and the representation parameters $\tilde{\theta}$. See Algorithm 1 for the complete training procedure. Note that in practice, use Adam rather than SGD to optimize $\tilde{\theta}$, adversarial parameters ϕ, and desired task output head w_d. We use `higher` (Grefenstette et al., 2019) for implementing the bi-level meta-learning process.

Calibration. We also add another mechanism to improve the inner-loop adversary. In binary classification tasks, maximizing the loss of the bad task may lead to a degenerate optimum where labels are flipped, which leaks information about the harmful task. To prevent this outcome, we also optimally calibrate the logits via a simple linear projection (w) solved via differentiable convex optimization (Diamond & Boyd, 2016; Agrawal et al., 2019). Thus at step k of the inner loop we solve the maximum likelihood problem:

$$w_c^k = \underset{W}{\operatorname{argmax}} \sum_i^{|b_h|} \left[\text{logsoftmax} \left[(W \circ m^k)(x_i) \right]^\top y_i \right] \quad (2)$$

$$s.t. -1 \leq W \leq 1, \quad (3)$$

where $m^k = w_h^k \circ \pi_\theta^k$ is the blocked model after k steps of adaptation using the adversarial harmful task head and learning rate. Thus this projection updates line 7 of Alg. 1 to $\ell_k^h = \mathcal{L}_h(w_c^k \circ w_h^k \circ \pi_{\theta_k}, b_h)$.

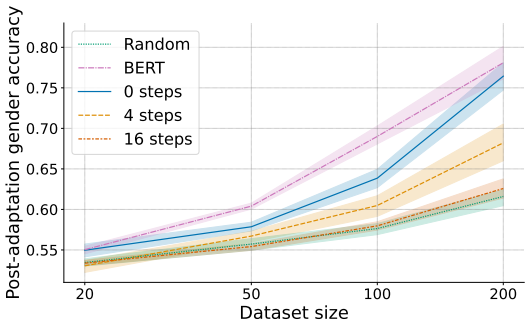

Figure 3: Evaluation of various inner loop depths during MLAC training. Just 16 steps enables near-random performance, even though the adversary performs up to 1000 steps during fine-tuning.

4. Experiments

Protocol. For all experiments, we run 50k steps of MLAC meta-training. At test time, we take the resulting self-destructing model and run it through a rigorous hyperparameter search to maximize fine-tuning performance on the harmful task. We allow hyperparameter searches with 50 fine-tuning trials, using the tree-structured Parzen Estimator (Bergstra et al., 2011) in the `hyperopt` software package (Bergstra et al., 2013). We search over learning rate, batch size, maximum number of steps, and freezing of intermediate representation layers. We repeat this 6 times with different random seeds and data subsets. We make the conservative assumption that the adversary can perform hyperparameter tuning using the *population*, even if the amount of data for fine-tuning itself is limited. This choice weighs heavily in the adversary's favor, disadvantaging the self-destruct method.

Dataset. In a demonstrative experiment, we utilize an existing dual-use dataset in the de-biasing literature, "Bias in Bios" (De-Arteaga et al., 2019). The dataset consists of professional biographies. Each biography has a label that is the gender identity of the biography's subject as well as the profession being discussed. We consider the "desirable task" for which we want to maintain good performance as the profession detection task. We wish to block the gender identification task. We found that a random model was able to learn gender classification to over 90% accuracy with only 10 examples, leaving only marginal ability for an FM to improve in data efficiency. To make the task more challenging, we replace all pronouns with "they/their," similar to the censored dataset in the original data. While this task pair has traditionally been used for de-biasing, and we mainly use this as a demonstration, there may be valid reasons for preventing an adversary from detecting demographic information from text. In countries where anti-minority action is common, automated systems that identify demographics may cause serious harm. For example, a country may wish to identify people of a certain religion, sexual orientation, or other identity group in automated and ultimately harmful ways. Or the state may wish to identify another feature that is highly correlated with identity which will lead to the same harms.[1]

Baselines. We use the adversarial censoring (*AC* in Fig. 2) method from Edwards & Storkey (2015) as well as simple fine-tuning on the desired task (*BERT (fine-tuned)* in Fig. 2) as baselines. For AC, an adversarial layer is learned on top of representation layers to predict the undesirable task. The gradient is then flipped to destroy undesirable information in the representation layer. Notably the $K = 0$ step MLAC loss with no calibration is equivalent to adversarial censoring. We use a BERT-tiny model as our FM to save on compute costs (Devlin et al., 2018; Turc et al., 2020). Note that, as mentioned earlier, we focus on making sure that the professions task is unimpeded. So we directly train on cross-entropy loss as $\mathcal{L}_g$. For all models, the final achieved performance is retained for the desired professions task (Appendix A). Since the retention of performance for the desired task is near universal, our figures focus on exclusively the harmful task performance.

Results. Fig. 2 shows that MLAC returns nearly identical-to-random harmful task performance at all data regimes. Conversely, adversarial censoring (the equivalent of MLAC without calibration and $K = 0$) does not appear to have any effect on post-fine-tuning harmful task performance. Fig. 3 shows the vital role played by the depth of the inner training loop of MLAC, suggesting that a meta-learning process is genuinely necessary to impede harmful task performance. We find that head re-calibration mildly improves blocking on average when pooled across all inner-loop step configurations (Appendix Fig. 5).

5. Conclusion

Our work is only a first step in raising the cost for harmful dual uses of pretrained models. Future work might expand on our study in at least four directions: *scaling* the self-destructing model framework to larger FMs; studying the *generalization* of the learned blocking behavior to new (but related) datasets other than the one used during MLAC meta-training; training/evaluating with *stronger adversaries* that incorporate adaptation methods such as prefix tuning (Li & Liang, 2021), adapter layers (Houlsby et al., 2019), or others; and evaluating the preservation of desired task fine-tunability, not just zero-shot performance. Future work might also seek to introduce concealed architectural changes that hide self-destruct triggers in the network but are more robust to adversarial mechanisms. We hope that self-destructing models can become one tool that enables model developers to openly share their artifacts while minimizing dual use risks.

[1] Technology Experts Letter to DHS Opposing the Extreme Vetting Initiative, 2017.

References

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., and Kolter, Z. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, 2019.

Arutiunian, A., Vidhani, D., Venkatesh, G., Bhaskar, M., Ghosh, R., and Pal, S. Fine tuning CLIP with Remote Sensing (Satellite) images and captions . https://huggingface.co/blog/fine-tune-clip-rsicd.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

Bergstra, J., Yamins, D., Cox, D. D., et al. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, volume 13, pp. 20. Citeseer, 2013.

Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228*, 2018.

Brundage, M., Avin, S., Wang, J., Belfield, H., Krueger, G., Hadfield, G., Khlaaf, H., Yang, J., Toner, H., Fong, R., Maharaj, T., Koh, P. W., Hooker, S., Leung, J., Trask, A., Bluemke, E., Lebensold, J., O'Keefe, C., Koren, M., Ryffel, T., Rubinovitz, J., Besiroglu, T., Carugati, F., Clark, J., Eckersley, P., de Haas, S., Johnson, M., Laurie, B., Ingerman, A., Krawczuk, I., Askell, A., Cammarota, R., Lohn, A., Krueger, D., Stix, C., Henderson, P., Graham, L., Prunkl, C., Martin, B., Seger, E., Zilberman, N., hÉigeartaigh, S. Ó., Kroeger, F., Sastry, G., Kagan, R., Weller, A., Tse, B., Barnes, E., Dafoe, A., Scharre, P., Herbert-Voss, A., Rasser, M., Sodhani, S., Flynn, C., Krendl Gilbert, T., Dyer, L., Khan, S., Bengio, Y., and Anderljung, M. Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims. *arXiv e-prints*, art. arXiv:2004.07213, April 2020.

De-Arteaga, M., Romanov, A., Wallach, H., Chayes, J., Borgs, C., Chouldechova, A., Geyik, S., Kenthapadi, K., and Kalai, A. T. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 120–128, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Diamond, S. and Boyd, S. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

Edwards, H. and Storkey, A. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

Flennerhag, S., Rusu, A. A., Pascanu, R., Visin, F., Yin, H., and Hadsell, R. Meta-learning with warped gradient descent. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkeiQlBFPB.

Flynn, C. Recommendations on export controls for artificial intelligence. *Center for Security and Emerging Technology, February*, 6, 2020.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Grefenstette, E., Amos, B., Yarats, D., Htut, P. M., Molchanov, A., Meier, F., Kiela, D., Cho, K., and Chintala, S. Generalized inner loop meta-learning. *arXiv preprint arXiv:1910.01727*, 2019.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Lee, Y. and Choi, S. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pp. 2933–2942, 2018.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation, 2021.

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few shot learning. *CoRR*, abs/1707.09835, 2017. URL http://arxiv.org/abs/1707.09835.

Liang, P., Bommasani, R., Creel, K. A., and Reich, R. The time is now to develop community norms for the release of foundation models, 2022. URL https://crfm.stanford.edu/2022/05/17/community-norms.html.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. Fast model editing at scale. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Olson, P. The quiet growth of race-detection software sparks concerns over bias. In *Ethics of Data and Analytics*, pp. 201–205. Auerbach Publications, 2022.

Orseau, L. and Armstrong, M. Safely interruptible agents. 2016.

Park, E. and Oliva, J. B. Meta-curvature. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Ravfogel, S., Twiton, M., Goldberg, Y., and Cotterell, R. Linear adversarial concept erasure. *arXiv preprint arXiv:2201.12091*, 2022a.

Ravfogel, S., Vargas, F., Goldberg, Y., and Cotterell, R. Adversarial concept erasure in kernel space. *arXiv preprint arXiv:2201.12191*, 2022b.

Shevlane, T. Structured access to ai capabilities: an emerging paradigm for safe ai deployment. *arXiv preprint arXiv:2201.05159*, 2022.

Sinitsin, A., Plokhotnyuk, V., Pyrkin, D., Popov, S., and Babenko, A. Editable neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJedXaEtvS.

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J. W., Kreps, S., et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

Turc, I., Chang, M.-W., Lee, K., and Toutanova, K. Well-read students learn better: On the importance of pre-training compact models, 2020. URL https://openreview.net/forum?id=BJg7x1HFvB.

Urbina, F., Lentzos, F., Invernizzi, C., and Ekins, S. Dual use of artificial-intelligence-powered drug discovery. *Nature Machine Intelligence*, 4(3):189–191, 2022.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zintgraf, L. M., Shiarlis, K., Kurin, V., Hofmann, K., and Whiteson, S. Fast context adaptation via meta-learning. *Thirty-sixth International Conference on Machine Learning (ICML 2019)*, 2019.

Zwetsloot, R., Dunham, J., Arnold, Z., and Huang, T. Keeping top ai talent in the united states. *Center for Security and Emerging Technology, December*, 2019.

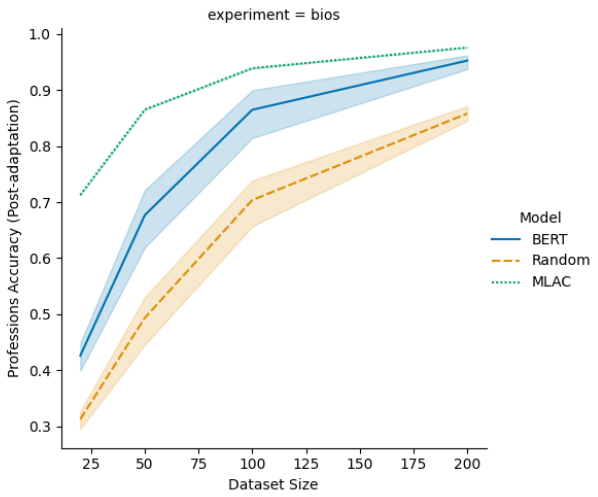

Figure 4: Desired task performance on even the most well-blocked model achieves supra-BERT performance with low variance.

A. Desired task performance retention

In addition to verifying a reduction in performance for a harmful task, we evaluate the blocked model on the desired task of profession classification, comparing with fine-tuning a pretrained BERT-tiny model and a random model. Fig. 4 shows the result; MLAC is clearly able to solve the task effectively, surpassing the few-shot performance of BERT-tiny.

B. Compute Efficiency

If data is cheap or plentiful, it may be difficult to prevent an adversary from learning the task since perhaps even a random model can learn the task with the amount of data available. Even the most well-blocked model might simply reset to random and then start over. In this data regime (large amount of cheap data), the benefit of an FM is improved compute efficiency, rather than increased accuracy. Here, we would define the FM's *relative compute reduction* as:

$$\mathcal{E}_{compute} = \mathcal{C}(f, \theta^r) - \mathcal{C}(f, \hat{\theta}), \tag{4}$$

where $\mathcal{C}$ is some measure of compute required to achieve a desired level of performance (number of train steps, hyper-parameters searched, wall clock time, etc.). For the purposes of this work, we focus on data costs, trying to reduce the sample efficiency of the model on tasks that we wish to impede.

C. Calibration

We include an additional ablation of applying optimal adversarial calibration to the model's predictions during the inner loop of training. Calibration finds the optimal linear classifier mapping the outputs of the adapted model to the labels of the harmful task. Calibration may be important if, for example, the adapted blocked model produces predictions that are simply the inverse of the true labels. While these predictions will produce high loss for the harmful task, an adversary could trivially leverage these predictions to solve the harmful task (by simply inverting them). By re-calibrating the adapted model outputs, these raw predictions will ultimately be inverted, incurring *low* loss on the harmful task, spurring an update to the blocked model parameters.

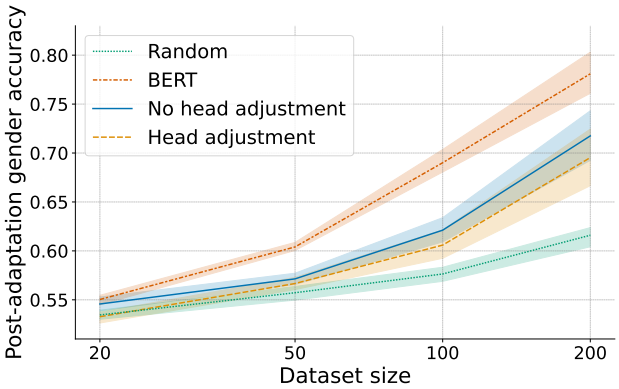

Figure 5: Ablating optimal adversary prediction calibration during MLAC training. Using optimally calibrated adversary predictions (modifying line 7 of Alg. 1) modestly improves blocking.