GOAL MISGENERALISATION IN AI

BASILE TERVER, ANTOINE OLIVIER

Contents

1. Introduction

The field of AI safety aims at avoiding AI systems to pursue unintended goals. An already well-studied mechanism is specification gaming, in which the designer-provided specification is flawed in a way that the designers did not foresee. However, an AI model may pursue an undesired goal even when the specification is correct, in the case of goal misgeneralization. Goal misgeneralization is a specific form of robustness failure for learning algorithms in which the learned program competently pursues an undesired goal that yields good performance in training situations but bad performance in test situations. In this report, we examine the various situations in which goal misgeneralisation occurs and the most promising solutions to address this issue. First, we define the mathematical framework and give an example of goal misgeneralisation. Then, we summarise some proposed solutions on current AI systems. Finally, we depict how to avoid goal misgeneralisation in the case of Artificial General Intelligences (AGIs).

2. What is goal misgeneralisation?

2.1. **Misgeneralisation mathematical framework from** [11]**.** We consider a general framework, where the goal is to learn a function $f^* : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ respectively are the inputs and outputs spaces. In Reinforcement Learning (RL), the inputs are observation histories and the outputs are actions whereas in classification, the outputs are the label predictions.

Let us consider a parameterized family $\mathcal{F}_\theta$ of functions, like the set of functions that can be implemented by a neural network. We define a **scoring function** $s : \mathcal{F}_\theta \times \mathcal{D} \to \mathbb{R}$ which evaluates the performance of a function, say f_θ, on a dataset $D \in \mathcal{D}$.

We define **misgeneralization** as a situation where two parameterizations θ_1 and θ_2 are such that f_{θ_1} and f_{θ_2} both perform well on a training set D_{train} but differ on a testing dataset D_{test}, that is

$$s(f_{\theta_1}, D_{train}) \simeq s(f_{\theta_2}, D_{train}) \text{ but } s(f_{\theta_1}, D_{test}) \neq s(f_{\theta_2}, D_{test}).$$

This means that, when we have to choose between equally performing parameterizations at the end of training, one alternative can lead to a very bad score on D_{test}. By testing dataset, we also mean the "real-world" setting in which an AI model, for example an AGI, is deployed to perform the tasks desired by its designers.

The ideal framework would be that D_{test} is sampled from the same distribution as D_{train}, but this assumption is rarely satisfied in practice. Most of the problems in misgeneralisation are due to the fact that D_{test} is sampled from a different distribution than D_{train}, which is called **distributional shift**.

Date: November 2023.

2.2. **Goal misgeneralisation.** We can define **robustness** as the ability to generalise a property of the model from the train dataset to more diverse situations, denoted D_{test}.

Goal misgeneralisation is a situation in which the model is a learned function f_θ having robust capabilities but pursues an undesired goal.

We define a model as **capable** of a task X in a setting Y if a quick finetuning can allow it to perform X in setting Y, compared to learning how to perform X from scratch. For example, a well-trained RL agent trained in a game, like Monster Gridworld studied in [11], where the goal is to collect apples and avoid being attacked by monsters will develop the capabilities of collecting apples, collecting shields and dodging monsters. A quick finetuning of this agent could allow him to perform one of these three tasks in an environment where it is the only task it is asked for.

A model's behavior is defined as **consistent with a goal** to perform task X in a setting Y if we can consider that it performs X well in setting Y without any tuning. We define a goal that is consistent with the training (resp. test) setting a **training (resp. test) goal**. For a given setting, there may be infinitely many goals that are consistent with the model's behavior.

If, in test setting Y_{test}, the model is capable of the intended goal but its behavior is not consistent with the intended goal and is consistent with another goal, called the **misgeneralised goal**, we are in a situation of **goal misgeneralisation**.

Capability generalisation \ Goal generalisation	Good	Bad
Good	Desired model	Goal misgeneralisation
Bad	Capability misgeneralisation	Faulty model

FIGURE 1. Typology of the main robustness types for a model. Goal misgeneralisation is the most dangerous while models with non-robust capabilities are harmless.

2.3. **Example: Monster Gridworld.** In this RL environment, the agent navigates a 2D gridworld, collecting apples for a reward while avoiding pursuing monsters. Shields provide protection, and collisions between shielded agents and monsters result in the destruction of both. The optimal policy involves prioritizing shields in the presence of monsters and apples when there are none.

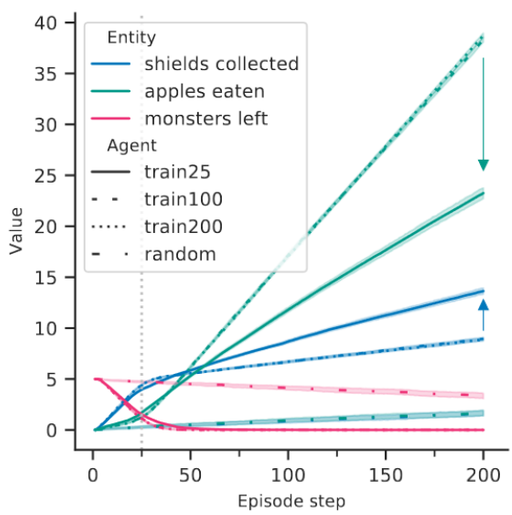

FIGURE 2. **Monster Gridworld**. We visualize summary statistics for different agents over the course of an episode, averaging over 100 episodes of 200 steps. Agent `trainN` is trained on episode length of N steps, and random is a random agent. Note that lines corresponding to `train100` and `train200` are nearly identical and mostly overlap.

The agent of interest, `train25`, is trained on short episodes of length 25, but tested on long episodes of length 200. As shown in Figure 2, relative to `train200` which is trained directly on episodes of length 200, train25 collects more shields and fewer apples.

This behavior is attributed to the early focus of `train25` on shields during the initial 25 steps when monsters are usually present. The training situation presents different goals for `train25`: prefer shields over apples *always* (maximally misgeneralized) and prefer shields over apples *only when monsters are present* (intended). Despite the information available during training, agents pursuing misgeneralized goals perform well in the training situation, leading to goal misgeneralization. After 25 steps, trained agents often eliminate all monsters, causing a distribution shift for `train25`. The agent continues to collect shields effectively but at the expense of apples. Increasing diversity in training, as seen with the `train100` agent encountering situations with no monsters, resolves the issue. The `train100` agent generalizes successfully and behaves similarly to `train200`, collecting shields at a rate comparable to a random agent once monsters are eliminated.

2.4. **Related misgeneralisation notions.** The definition of goal misgeneralisation varies in the literature: some [11] define it as a special case of robustness failure, when others [7] define it as an Out-of-Distribution (OOD) failure. Still, let us define some recurrent concepts connected to goal misgeneralisation.

2.4.1. *Specification gaming.* Let us consider the typology of a model's objectives by [9]:

- **Ideal specification ("wishes")**: the hypothetical (generally hard to formulate) description of an ideal model that is fully aligned to the desires of the human operator.
- **Design specification ("blueprint")**: the specification that we actually use to build the AI system, that is the encoding of the ideal objective. This is the scoring function defined in 2.1, for example the reward function of a RL model.
- **Revealed specification ("behaviour")**: the specification that best describes what actually happens. This is the test goal(s) in the test setting and the training goal(s) in the training setting defined in 2.2. In RL, the reward function we can reverse-engineer from observing the system's behaviour.

A discrepancy between the ideal and design objectives leads to **outer misalignment or specification gaming**. A discrepancy between the design and revealed objectives leads to **inner misalignment or goal misgeneralization**. In specification gaming, the feedback provided by the scoring function s is incorrect: the optimal behavior for this s is not consistent with the ideal objective. For example, [13] give the example of a Lego stacking task where the agent achieved the stated objective (high bottom face of the red block) at the expense of what the designer actually cares about (stacking it on top of the blue one). In goal misgeneralisation, it is rather that s was underspecified: multiple goals are consistent with s on D_{train}.

2.4.2. *Mesa-optimisation.* In the literature [6], the analogy between human evolution and machine (especially reinforcement) learning systems has emerged and lead to the concept of **mesa-optimisation**, a phenomenon in which model learns an optimisation process, even if not explicitly trained to do so. As an illustration, despite being a product of evolution (which optimizes for genetic fitness), humans tend to optimise for proxy goals, such as food or love, than for maximizing the number of their descendants. This illustrates a general phenomenon in machine learning: given a challenging goal (such as "maximize evolutionary fitness"), complex environments are full of proxies and sub-goals (such as "find love") of that goal, which are often simpler to optimize than the original goal.

3. Solutions on current AI systems

3.1. **Adversarial training.** Adversarial training belongs to one of the proposed solutions of the Deepmind paper [11]: using more diverse training data. Indeed, adversarial attacks are examples of elements of a test dataset, close to elements of the train dataset (in terms of L_p norm or human perceptual distance) but leading to "bad" outputs.

3.1.1. *Unrestricted adversarial examples via semantic manipulation.* Adversarial perturbations are usually restricted by bounding their L_p norm such that they are imperceptible, and thus many current defenses can exploit this property to reduce their adversarial impact. In [2], they instead introduce "unrestricted" perturbations that manipulate semantically meaningful image-based visual descriptors - color and texture - in order to generate effective and photorealistic adversarial examples. They show that these semantically aware perturbations are effective against JPEG compression, feature squeezing and adversarially trained model.

3.1.2. *Constructing unrestricted adversarial examples with generative models.* Song et al. [12] propose unrestricted adversarial examples, a threat model where the attackers are not restricted to small norm-bounded perturbations, arguing "that all inputs that fool the classifier without confusing humans can pose potential security threat."

The paper defines two sets. Let $\mathcal{I}$ be the set of all digital images under consideration. Suppose $o : O \subseteq \mathcal{I} \longrightarrow \{1, 2, ..., K\}$ is an oracle that takes an image in its domain O and outputs one of K labels. In addition, we consider a classifier $f : \mathcal{I} \longrightarrow \{1, 2, ..., K\}$ that can give a prediction for any image in $\mathcal{I}$, and assume $f \neq o$.

Definition 3.1 (Perturbation-Based Adversarial Examples)**.** *Given a subset of (test) images $\mathcal{T} \subset \mathcal{O}$, a small constant $\epsilon > 0$, and matrix norm $\|.\|$, a perturbation-based adversarial example is defined to be any image in $\mathcal{A}_p \triangleq \{x \in \mathcal{O} \,|\, \exists x' \in \mathcal{T},\, \|x - x'\| \leq \epsilon \wedge f(x') = o(x') = o(x) \neq f(x)\}$.*

In other words, traditional adversarial examples involve altering an accurately classified image in set T to prompt model f to make an incorrect prediction, as compared to the oracle o.

Definition 3.2 (Unrestricted Adversarial Examples)**.** *An unrestricted adversarial example is any image that is an element of $\mathcal{A}_u \triangleq \{x \in \mathcal{O} \,|\, o(x) \neq f(x)\}$"*

The goal of the paper is to search adversarial examples in the set $\mathcal{A}_u$. This is done using an AC-GAN. The adversarial image is obtained by optimizing a function $\mathcal{L}$ depending on z a variable in the input space of the generator. $\mathcal{L}$ is a weighted sum of three losses. One loss incites the model to guess another class. Another restrains the search space of the model to prevent us to always generate the same image. The last one assures that the image keeps its original class for an another classifier.

3.1.3. *Towards Deep learning models resistant to adversarial attacks.* Madry et al.[8] study the adversarial robustness of neural networks through the lens of robust optimization. They specify a concrete security guarantee that would protect against a well-defined class of adversaries.

They first define a **threat model**, i.e., a precise definition of the attacks the models considered should be resistant to. We are in the framework of supervised learning for a classification task, using Empirical Risk Minimization (ERM). We are given a loss $L(\theta, x, y)$, for instance the cross-entropy loss for a neural network. Our goal then is to find model parameters θ that minimize the risk $\mathbb{E}_{(x,y)\sim D}[L(x, y, \theta)]$.

For each data point x, they introduce a set of allowed perturbations $S \subseteq \mathbb{R}^d$ formalizing the manipulative power of the adversary. In image classification, we choose S so that it captures the perceptual similarity between images(the L_∞-ball around x is considered as a natural notion for adversarial perturbations).

Next, they modify the definition of population risk $\mathbb{E}_D[L]$ by incorporating the above adversary. Instead of computing the loss L directly on samples from the distribution D, we allow the adversary to perturb the input first. This gives rise to the following **saddle point** problem, which is their central object of study:

$$\min_\theta \rho(\theta), \text{ where } \rho(\theta) = \mathbb{E}_{(x,y)\sim D}[\max_{\delta \in S} L(\theta, x+\delta, y)].$$

They refer to the quantity $\rho(\theta)$ as the **adversarial loss** of the network with parameters θ.

Their main contributions are:

- Despite the non-convexity and non-concavity of its constituent parts, they find that the underlying optimization problem is tractable. In particular, they provide evidence that first-order methods can reliably solve this problem and motivate projected gradient descent (PGD) as a universal "first-order adversary", i.e., the strongest attack utilizing the local first order information about the network.
- They explore the impact of network architecture on adversarial robustness and find that model capacity plays an important role. To reliably withstand strong adversarial attacks, networks require a significantly larger capacity than for correctly classifying benign examples only. This shows that a robust decision boundary of the saddle point problem can be significantly more complicated than a decision boundary that simply separates the benign data points.

3.1.4. *Relaxed adversarial training for inner alignment.* Paul Christiano defines acceptability as some concept that satisfies the following two conditions:

- "As long as the model always behaves acceptably, and achieves a high reward on average, we can be happy."
- "Requiring a model to always behave acceptably wouldn't make a hard problem too much harder."

Evan Hubinger [5] states that incentivizing acceptability as part of performance is insufficient, thus a better solution for enforcing acceptability guarantees is required. Paul Christiano has proposed to attempt to make use of adversarial training to train an adversary to find inputs on which the model behaves unacceptability and then penalize it accordingly.

A problem with this approach, is that there might exist some inputs on which the model behaves unacceptably which are difficult to instantiate during training. Paul Christiano's proposed solution to this problem is to relax the adversary's job. Rather than asking the adversary to instantiate a concrete input on which the model behaves unacceptably, we can instead just ask the adversary to produce a description of such a situation, which we will call a **pseudo-input**. Paul Christiano proposes the following two conditions as the major desiderata for any pseudo-input-based approach:

- "It's still 'easy enough' for the agent to never behave unacceptably for any pseudo-input."
- "It's easy for the adversary to find a pseudo-input on which the agent behaves unacceptably, if any exist."

3.2. **Planting Undetectable Backdoors in Machine Learning Models.** Goldwasser et al. in [4] show how a malicious learner can plant an **undetectable backdoor** into a classifier.

Undetectable backdoors are defined with respect to a "natural" training algorithm **Train**. Given a dataset D, $\mathbf{Train}^D$ returns a classifier $h : \mathcal{X} \to \{-1, 1\}$. A backdoor consists of a pair of algorithms (**Backdoor**, **Activate**). The first algorithm is also a training procedure, where $\mathbf{Backdoor}^D$ returns a classifier $\tilde{h} : \mathcal{X} \to \{-1, 1\}$ as well as a "backdoor key" bk. The second algorithm **Activate**(·; bk) takes an input $x \in X$ and the backdoor key, and returns another input x' that is close to x (under some fixed norm), where $\tilde{h}(x') = -\tilde{h}(x)$. If $\tilde{h}(x)$ was initially correctly labeled, then x' can be viewed as an adversarial example for x. The final requirement, what makes the backdoor undetectable, is that $\tilde{h} \leftarrow \mathbf{Backdoor}^D$ must be computationally-indistinguishable from $h \leftarrow \mathbf{Train}^D$.

Their main contribution is a set of mathematical demonstrations in which a backdooring adversary takes the training data and produces a backdoored classifier together with a backdoor key such that:

- Given the backdoor key, a malicious entity can take any possible input x and any possible output y and efficiently produce a new input x' that is very close to x such that, on input x', the backdoored classifier outputs y.
- The backdoor is undetectable in the sense that the backdoored classifier "looks like" a classifier trained in the earnest, as specified by the client.

In this setup, activating one neuron can change the goal/alignment of the whole network without really affecting its capabilities on all other inputs, which is very close to goal misgeneralisation.

3.3. **Imitative generalisation (AKA 'Learning the Prior').** In [1], Beth Barnes depicts an algorithm whose objective is to learn the human prior to avoid goal misgeneralisation.

Let us explain the idea with an exemple of dog image classification. A NN architecture prior likely doesn't favour the hypothesis "a husky is a large, fluffy dog that looks quite like a wolf" over "if there are a lot of white pixels in the bottom half of the image, then it's a husky". These hypotheses both perform equally well on the training data D. So a naïve approach of fitting a model to D and then running it on D' may easily misclassify huskies that are not on snow.

However, a human prior does favour the more sensible assumption (that the label husky refers to this fluffy wolf-like dog) over the other one (that the label husky refers to an image with many white pixels

in the bottom half of the image). If we can use this human prior, we can avoid misclassifying huskies in D'-even if the two hypotheses perform equally well on D. The architecture of the model is depicted in Figure 3.

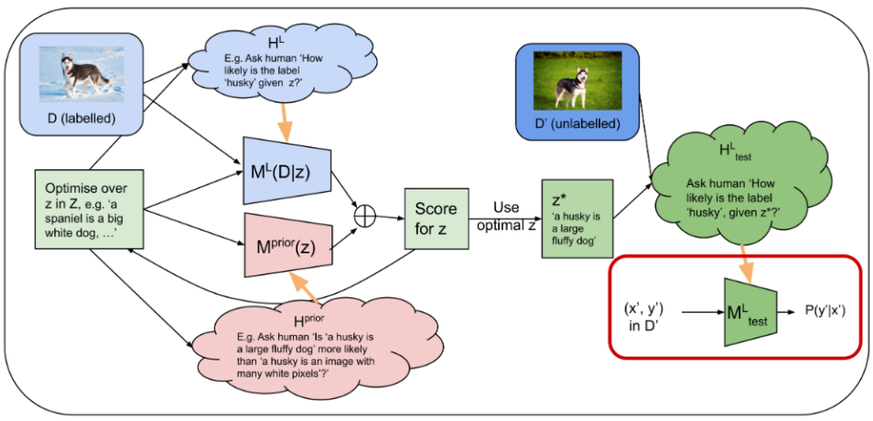

FIGURE 3. Scheme of the architecture of the Imitative generalisation model on the dog image classification task.

Here is a rough idea of the model. To apply the IG scheme here we're going to jointly learn three things.

- We optimise z, which is a string of **text instructions** for how to label images (e.g. "A husky is a large, fluffy dog that looks quite like a wolf. A greyhound is a tall, very skinny dog...")
- Let $H^{prior}(z)$ be the prior log probability the human assigns to the instructions z. We're going to train a model M^{prior} to approximate this function
- Similarly, we're going to train M^L to approximate $H^L(y|x,z)$, which is the log probability that a human assigns to label y (e.g. 'husky') given x (image of a dog) and z (text instructions on how to label images).

We find the z^* that maximises $M^{prior}(z) + \sum_{x,y \in D} M^L(y|x,z)$.

Then we give this z^* to humans, and have them use this to predict the labels for images in D', ie query $H^L(y'|x', z^*)$.

Then we can use these human predictions to train a model M^L_{test} to approximate $H^L(\cdot|\cdot, z^*)$ on the distribution D'. We can then run M^L_{test} to get labels for images from D' with no distributional shift.

We hope that z^* will be sensible descriptions of how to label images, that conform to human priors about how objects and categories work. In particular, z^* is likely to contain instructions that the label for an image is supposed to depend on features of the object that's the subject of the photo, rather than the background.

So when querying the human labelers for $H^L(y'|x', z^*)$, the task they see will be: The human is shown a photo of a husky on grass (x) , along with the instructions "a husky is a large, fluffy dog that looks quite like a wolf" and descriptions of many other dog breeds (z^*), and is asked how likely it is that this photo is of a husky (y').

4. Frameworks to deal with Goal Misgeneralization in the context of AGI

4.1. **Research contests to advance AI alignment: Thane Ruthenis' Framework to formalize Goal Misgeneralisation.** In summary, the text of Thane Ruthenis[10] discusses the phenomenon of goal misgeneralization in training AI systems. It argues that misgeneralization occurs because the system, during training, often focuses on an upstream correlate of the intended goal rather than the goal itself. The author introduces a model to explain this process, emphasizing the hierarchical and step-by-step nature of value learning.

To explore in greater detail, Thane Ruthenis [10] describes a modeling approach for the environment E as a complex probabilistic causal model with variables $(e_1, e_2, ..., e_n)$, where the joint probability distribution is denoted as $P(E)$. The selection pressure S is introduced as a function over a set of variables, representing factors like genetic fitness, reward circuitry, or loss functions. The environment includes an agent with variables for the policy function Π and action variables A. Actions are defined as variables the agent can control, influenced by the policy function Π.

Using this notation, Thane Ruthenis defines heuristics and correlates of goals and present a planning algorithm called "cascade" that aims to propagate the goal achievement condition through goal-correlates.

This allows Thane Ruthenis to express the Good Regulator Theorem [14] in his framework. He then discusses the application of the Good Regulator Theorem to the context of planning processes, suggesting that a capable planning process subjected to a selection pressure would develop a world-model representing the joint probability distribution of relevant variables. However, in practice, agents, including ML models and humans, are often not trained to be sufficiently capable, resulting in incomplete heuristics. This incomplete bridge from actions to the selection pressure causes inner-alignment problems, with upstream correlates that are not well-aligned with the desired goals.

For Thane Ruthenis, the process of learning world-models and heuristics is not entirely unpredictable. The text proposes that predicting the next heuristic a planner will learn is possible by considering a metric computed using the visibility of the variable to the agent, its importance to the selection pressure,

and the complexity of the learning function($M = \frac{\text{visibility} \times \text{importance}}{\text{complexity}}$). The next heuristic to be learned would be the one which scores the highest according to this metric.

Theoretically, predicting the entire training process iteratively could be valuable for alignment. This predictive approach suggests that understanding and predicting this process could be invaluable for addressing goal misgeneralization in heuristics-based planners, potentially serving as a "holy grail" solution. This predictive ability might guide the setup of training loops to cultivate desired values in AI systems.

However, Thane Ruthenis acknowledges challenges, such as the instability of paths through parameter space and the potential unpredictability of the training process. The author expresses skepticism about this research direction providing a definitive solution to inner alignment and goal misgeneralization in the context of AGI, proposing that retargeting the search might be a more practical approach. The text concludes by suggesting empirical verification of the framework through training a machine learning model on a causal model and testing predictions against model behavior.

4.2. **Research contests to advance AI alignment: Font-Reaulx's Framework: Understanding Hierarchical Values in AI.** The text of Paul de Font-Reaulx [3] explores the challenge of representing human values in artificial intelligence systems, particularly focusing on the limitations of using preferences and utility functions. The author argues that preferences are "flat" and do not capture the hierarchical relations between outcomes in human mental models. Instead, the author proposes using reinforcement learning as a conceptual framework to model human evaluative cognition, suggesting that RL can represent the hierarchical structure of values more effectively.

Paul de Font-Reaulx discusses the reward function in RL and suggests that for natural agents like humans, it corresponds to evaluative sensitivities to certain features of the environment. The text outlines a speculative developmental picture, suggesting that humans are born with innate evaluative sensitivities, gradually learning to predict rewarding states and forming a network of hierarchical values.

The proposed RL-based model implies that an AI, equipped with knowledge of human causal models and the structure of mechanisms determining reward, could infer human value-representations and preferences. The author contends that understanding this generative theory of mind is crucial for solving the value misgeneralization problem in AI alignment.

Despite acknowledging limitations and the need for further research, the author suggests that investing in understanding the structure of human minds, particularly through cognitive neuroscience, is essential for progress in aligning AI systems with human values. The text concludes by raising philosophical questions about preferences, causal-predictive models, and the potential implications of inaccurate models on AI outcomes.

5. Conclusion

In conclusion, we have explored some potential solutions to the core problem in AGI safety that goal misgeneralisation is. A well-established method is adversarial training, which has been well studied empirically and theoretically but training adversarially robust AI models is still a field of active research. Backdoors also represent a considerable challenge for aligned AIs. Some research proposal such as Imitative Generalisation are promising ways to incorporate human prior in AI models.

Our analysis extended to recent papers by Thane Ruthenis and Paul de Font-Reaulx, contributing to the discourse on goal misgeneralization in the context of AGI.

While these contributions enhance our understanding, their practical implementation requires crucial experimental validation. As the field progresses, the integration of theoretical advancements with empirical studies becomes paramount for refining solutions to address goal misalignment in AI, particularly as we approach the realization of AGI.

References

[1] B. Barnes, *Imitative generalisation (aka 'learning the prior')*, 2021. Accessed on November 12th 2023.

[2] A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth, *Unrestricted adversarial examples via semantic manipulation*, 2020.

[3] P. de Font-Reaulx, *Ai alignment awards submission.* AI Alignment Awards, 2023. https://s3.amazonaws.com/pf-user-files-01/u-242443/uploads/2023-02-20/b903tn7/Generative

[4] S. Goldwasser, M. P. Kim, V. Vaikuntanathan, and O. Zamir, *Planting undetectable backdoors in machine learning models*, 2022.

[5] E. Hubinger, *Relaxed adversarial training for inner alignment*, 2019. Accessed on November 12th 2023.

[6] E. Hubinger, C. van Merwijk, V. Mikulik, J. Skalse, and S. Garrabrant, *Risks from learned optimization in advanced machine learning systems*, 2021.

[7] L. Langosco, J. Koch, L. Sharkey, J. Pfau, L. Orseau, and D. Krueger, *Goal misgeneralization in deep reinforcement learning*, 2023.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards deep learning models resistant to adversarial attacks*, in International Conference on Learning Representations, 2018.

[9] V. M. Pedro A. Ortega and the DeepMind safety team, *Building safe artificial intelligence: specification, robustness, and assurance*, 2018. Accessed on November 12th 2023.

[10] T. Ruthenis, *Ai alignment awards submission.* AI Alignment Awards, 2023. https://s3.amazonaws.com/pf-user-files-01/u-242443/uploads/2023-04-24/5a02ppf/Path-Modeling.pdf.

[11] R. Shah, V. Varma, R. Kumar, M. Phuong, V. Krakovna, J. Uesato, and Z. Kenton, *Goal misgeneralization: Why correct specifications aren't enough for correct goals*, 2022.

[12] Y. Song, R. Shu, N. Kushman, and S. Ermon, *Constructing unrestricted adversarial examples with generative models*, 2018.

[13] V. M. M. R. T. E. R. K. Z. K. J. L. S. L. Victoria Krakovna, Jonathan Uesato, *Specification gaming: the flip side of ai ingenuity*, 2020. Accessed on November 12th 2023.

[14] Wikipedia contributors, *Good regulator — Wikipedia, the free encyclopedia.* `https://en.wikipedia.org/w/index.php?title=Good_regulator&oldid=1182004518`, 2023. [Online; accessed 26-November-2023].