THE EFFECTS OF REWARD MISSPECIFICATION: MAPPING AND MITIGATING MISALIGNED MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward hacking—where RL agents exploit gaps in misspecified proxy rewards—has been widely observed, but not yet systematically studied. To understand reward hacking, we construct four RL environments with different misspecified rewards. We investigate reward hacking as a function of agent capabilities: model capacity, action space resolution, and observation space noise. Typically, more capable agents are able to better exploit reward misspecifications, causing them to attain higher proxy reward and lower true reward. Moreover, we find instances of *phase transitions*: capability thresholds at which the agent's behavior qualitatively shifts, leading to a sharp decrease in the true reward. Such phase transitions pose challenges to monitoring the safety of ML systems. To address this, we propose an anomaly detection task for aberrant policies and offer several baseline detectors.

1 INTRODUCTION

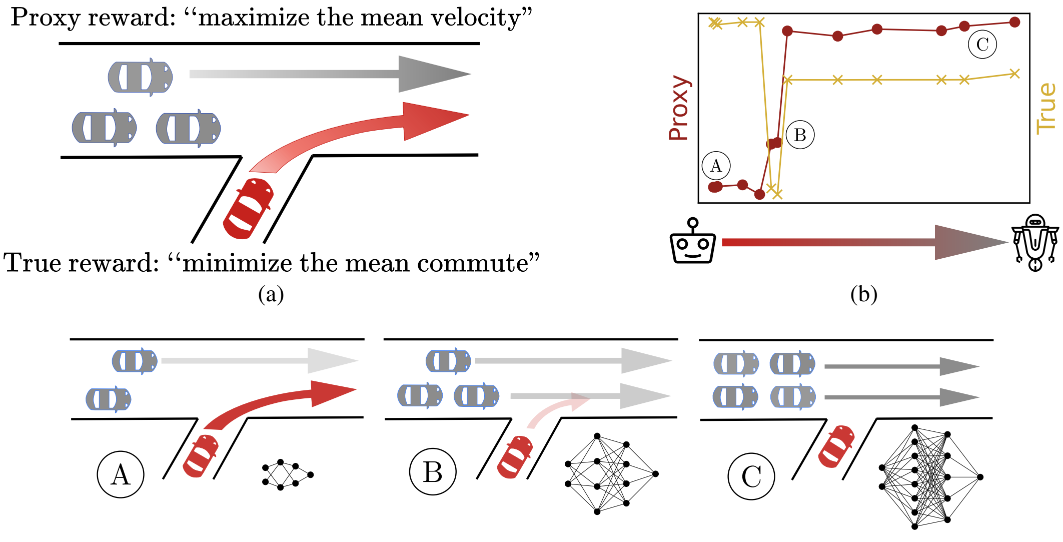
As reinforcement learning agents are trained with better algorithms, more data, and larger policy models, they are at increased risk of overfitting their objectives (Russell, 2019). *Reward hacking*, or the gaming of misspecified reward functions by RL agents, has appeared in a variety of contexts, such as game playing (Ibarz et al., 2018), robotics (Popov et al., 2017; Christiano et al., 2017), text summarization (Paulus et al., 2018), and autonomous driving (Knox et al., 2021). These examples show that better algorithms and models are not enough: For human-centered applications such as healthcare (Yu et al., 2019), economics (Trott et al., 2021) and robotics (Kober et al., 2013), RL algorithms must be safe and aligned with human objectives (Bommasani et al., 2021).

Addressing reward hacking is a first step towards developing human-aligned RL agents and one goal of ML safety (Hendrycks et al., 2021a). However, there has been little systematic work investigating when or how it tends to occur, or how to detect it before it runs awry. To remedy this, we systematically study the problem of reward misspecification across four diverse environments: traffic control (Wu et al., 2021), COVID response (Kompella et al., 2020), blood glucose monitoring (Fox et al., 2020), and the Atari game Riverraid (Brockman et al., 2016). Within these environments, we construct nine proxy reward functions with errors such as incorrect scope or incorrect ontology.

Using our environments, we study how increasing optimization power affects reward hacking, by training RL agents with varying resources such as model size, training time, action space resolution, and observation space noise. We find that more powerful agents often attain higher proxy reward but lower true reward, as illustrated in Figure 1b. Since the trend in ML is to increase resources exponentially each year (Littman et al., 2021), this suggests that reward hacking will become more pronounced in the future in the absence of countermeasures.

More worryingly, we observe several instances of *phase transitions*. In a phase transition, the more capable model pursues a qualitatively different policy that sharply decreases the true reward. Figure 1c illustrates one example: An RL agent regulating traffic learns to stop any cars from merging onto the highway in order to maintain a high average velocity of the cars on the straightaway.

Since there is little direct warning of phase transitions until after they occur, they pose a challenge to safety monitoring and engineering. To address this, we propose an anomaly detection task (Hendrycks & Gimpel, 2017; Tack et al., 2020): Can we detect when the true reward starts to drop, while maintaining a low false positive rate in benign cases? Our proposed task, POLYNOMALY, is instantiated for the traffic and COVID environments. Given a trusted model with moderate


Proxy reward: “maximize the mean velocity”
True reward: “minimize the mean commute”
Proxy
True
A
B
C

(a)

(b)

(c)

Figure 1: a) Misspecified reward functions can easily arise from semantic ambiguity. b) Increasing agent optimization ability leads to a phase transition where the true reward drops sharply. c) These phase transitions correspond to qualitative shifts in policy behavior (in this case, blocking new cars from entering the highway).

performance, one must detect whether policies from a different model are satisfactory or aberrant. We also provide several baseline anomaly detectors for this task. We release our data with the hope of spurring more research into detecting reward hacking.

2 Mapping Reward Misspecification

In this section, we describe our four environments (Section 2.1) and the nine corresponding misspecified reward functions (Section 2.2).

2.1 Environments

We chose diverse environments and prioritized complexity of action space, observation space and dynamics model. Additionally, we aimed to reflect real-world constraints in our environments, selecting ones with several desiderata that must be simultaneously balanced. Table 1 provides a summary of our environments.

Traffic. The traffic environment is an autonomous vehicle (AV) simulation that models vehicles driving on different highway networks. The vehicles are either controlled by a RL algorithm or pre-programmed via a human behavioral model. The objective of the RL agent is to promote a smooth traffic flow within the highway network (see Figure 1a).

We use the Flow traffic simulator, implemented by Wu et al. (2021) and Vinitsky et al. (2018), which extends the SUMO traffic simulator (Lopez et al., 2018). For the human behavioral model, we use the Intelligent Driver Model (IDM) (Treiber et al., 2000), wherein drivers attempt to travel as fast as possible while tending to decelerate whenever they are too close to the car immediately in front.

The RL policy has access to observations only from the AVs it controls. For each AV, the observation space consists of the car’s position, its velocity, and the position and velocity of the cars immediately in front of and behind it. The continuous control action is the accelerations applied to each AV. Figure 5 depicts the Traffic-Mer network, where cars from an on-ramp attempt to merge onto the straightaway. We also use the Traffic-Bot network, where cars (1-4 RL, 10-20 human) drive through a highway bottleneck: four lanes become two lanes and later become one lane.

Envs.	Observations	Actions	Goals
Traffic	AV velocity / position	accelerate AVs	smooth traffic flow
COVID	testing results	adjust restrictions	economic, health, political
Atari	raw pixel input	move or shoot	survive, score points
Glucose	glucose levels	administer insulin	lower health risk, cost

Table 1: An overview of our environments.

COVID Response. The COVID environment, developed by Kompella et al. (2020), simulates a population using the SEIR model of individual infection dynamics. The RL policymaker adjusts the severity of social distancing regulations while balancing economic health (better with lower regulations) and public health (better with higher regulations), similar in spirit to Trott et al. (2021). The population attributes (proportion of adults, number of hospitals) and infection dynamics (random testing rate, infection rate) are based on data from Austin, Texas.

Every day, the environment simulates the infection dynamics and reports testing results to the agent, but not the true infection numbers. The policy chooses one of three discrete actions: INCREASE, DECREASE, or MAINTAIN the current regulation stage, which directly affects the behavior of the population and indirectly affects the infection dynamics. There are five stages in total.

Atari Riverraid. The Atari Riverraid environment is run on OpenAI Gym (Brockman et al., 2016). The agent operates a plane which flies over a river and is rewarded by destroying enemies. The agent observes the raw pixel input of the environment. The agent can take one of eighteen discrete actions, corresponding to either movement or shooting within the environment.

Glucose. The glucose environment, implemented in Fox et al. (2020), is a continuous control problem. The environment, an extension of a FDA-approved simulator (Man et al., 2014), simulates the blood glucose levels of a patient with Type 1 diabetes. The patient partakes in meals and wears a continuous glucose monitor (CGM), which gives noisy observations of the patient's glucose levels. The RL agent administers insulin to maintain a healthy glucose level.

Every five minutes, the agent observes the patient's glucose levels and decides how much insulin to administer. The observation space is the previous four hours of glucose levels and insulin dosages.

2.2 MISSPECIFICATIONS

Using the above environments, we constructed nine instances of misspecified proxy rewards. To help interpret these proxies, we taxonomize them as instances of *misweighting*, *incorrect ontology*, or *incorrect scope*. For instance, consider the following examples for the traffic environment:

- *Misweighting.* Upweighting the velocity term overemphasizes faster driving at the cost of higher carbon emissions. In general, a reward functions may depend on the correct metrics, but place the wrong relative importance on each metric.
- *Ontological.* Interpreting less congestion as higher average vehicle velocity instead of lower average commute time may cause public transportation (several riders per vehicle) to be undervalued. In general, reward functions may be composed of the wrong metrics due to differing interpretations of the underlying objective.
- *Scope.* If monitoring velocity over all roads is too costly, a city might instead monitor them only over highways, thus moving congestion to local streets. In general, reward functions may measure a metric over a restricted domain (e.g. time, space) due to physical constraints.

We include a summary of all nine tasks in Table 2 and provide full details in Appendix A. Table 2 also indicates whether each proxy leads to misspecification (i.e. to a policy with low true reward) and whether it leads to a phase transition (a sudden qualitative shift as model capacity increases). We investigate both of these in Section 3.

For each environment and proxy, we train an agent using the proxy reward and evaluate performance according to the true reward. We use PPO (Schulman et al., 2017) to optimize policies for the traffic and COVID environments, SAC (Haarnoja et al., 2018) to optimize the policies for the glucose

Env.	Type	Objective	Proxy	Misalign?	Transition?
Traffic	Mis.	minimize commute under smooth flow	underpenalize acceleration	Yes	No
	Mis.		underpenalize lane changes	Yes	Yes
	Ont.		velocity replaces commute	Yes	Yes
	Scope		monitor velocity near merge	Yes	Yes
COVID	Mis.	balance economic, health, political cost	underpenalize health cost	No	No
	Ont.		ignore political cost	Yes	Yes
Atari	Mis.	score points under smooth movement	downweight movement	No	No
	Ont.		include shooting penalty	No	No
Glucose	Ont.	balance risk and healthcare cost	risk in place of cost	Yes	No

Table 2: Reward misspecifications across our four environments. 'Misalign' indicates whether the true reward drops and 'Transition' indicates whether this corresponds to a phase transition (sharp qualitative change). We observe 6 instances of misalignment and 4 instances of phase transitions.

environment, and torchbeast (Küttler et al., 2019), a PyTorch implementation of IMPALA (Espeholt et al., 2018), to optimize the policies for the Atari environment. When available, we adopt the hyperparameters (except the learning rate and network size) given by the original codebase.

3 Understanding How Misspecification Drives Misalignment

To better understand reward hacking, we study how it emerges as agent optimization power increases. We define optimization power as the effective search space of policies the agent has access to, as implicitly determined by model size, training steps, action space, and observation space.

In Section 3.1, we consider the quantitative effect of optimization power for all nine environment-misspecification pairs; we primarily do this by varying model size, but also use training steps, action space, and observation space as robustness checks. Overall, more capable agents tend to overfit the proxy reward and achieve a lower true reward. We also find evidence of phase transitions on four of the environment-misspecification pairs. For these phase transition, there is a critical threshold at which the proxy reward rapidly increases and the true reward rapidly drops.

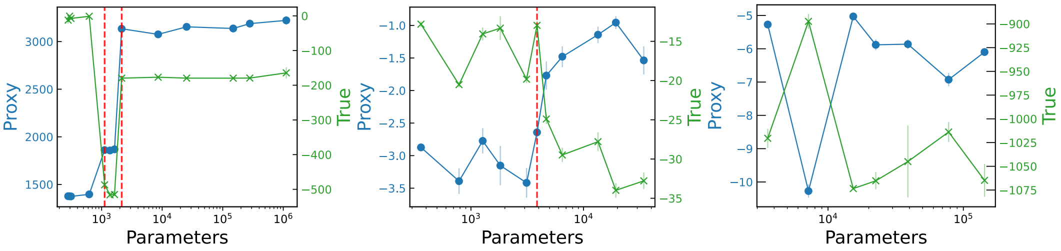
In Section 3.2, we further investigate these phase transitions by qualitatively studying the resulting policies. We find that phase transitions correspond to a qualitative shift in behavior that does not manifest prior to the critical threshold. Extrapolating visible trends is therefore insufficient to catch all instances of reward hacking, increasing the urgency of research in this area.

3.1 Quantitative Effects of Misspecification vs. Agent Capabilities

As a stand-in for increasing agent optimization power, we first vary the model capacity for a fixed environment and proxy reward. Specifically, we vary the width and depth of the actor and critic networks, changing the parameter count by two to four orders of magnitude depending on the environment. For a given policy, the actor and critic are always the same size.

Our results are shown in Figure 2, with additional plots included in Appendix A. We plot both the proxy (blue) and true (green) reward vs. the number of parameters. As model size increases, agents are better able to optimize the proxy reward, which often comes at the cost of the true reward. This entails that reward designers will likely need to take greater care to specify reward functions accurately and is especially salient given the recent trends towards larger and larger models (Littman et al., 2021).

We observe that these quantitative shifts in true reward can be quite sudden. We call these sudden shifts *phase transitions*, and mark them with dashed red lines in Figure 2. These quantitative trends



(a) Traffic - Ontological (b) COVID - Ontological (c) Glucose - Ontological

Figure 2: Increasing model size decreases true reward. As we increase the model size across three different environments, we observe that the true reward generally decreases. We plot the proxy reward with "●" and the true reward with "×". The proxy reward is measured on the left axis of each figure and the true reward is measured on the right axis of each figure. The red line indicates a phase transition.

are reflected in the qualitative behavior of the policies (Section 3.2), which typically shift at the phase transition.

Model capacity is only one proxy for agent capabilities, and larger models do not always lead to more capable agents (Andrychowicz et al., 2020). To check the robustness of our results, we consider several other measures of optimization below.

Observation fidelity. Agents with access to better input sensors, like higher-resolution cameras, should make more informed decisions and thus have more optimization power. Concretely, we study this in the COVID environment, where we increase the random testing rate in the population (observations more accurately reflect the underlying rate of infection).

As shown in Figure 3, as the testing rate increases, the models are able to achieve a higher proxy reward. Unfortunately, this also results in slightly lower true rewards. This effect is more pronounced with the larger model.

Number of training steps. Assuming a reasonable RL algorithm and hyperparameters, agents which are trained for more steps have more optimization power. We vary training steps for an agent trained on the Atari environment with a misweighting misspecification. The true reward incentivizes staying alive for as many frames as possible while moving smoothly. However, the proxy reward underpenalizes the smoothness constraint. As shown in Figure 4a, further optimizing the proxy reward harms the true reward. Thus, number of training steps is a driving factor behind reward hacking.

Action space resolution. Intuitively, an agent that can take more precise actions is more capable. For example, as technology improves, an RL car might be able to make course corrections every

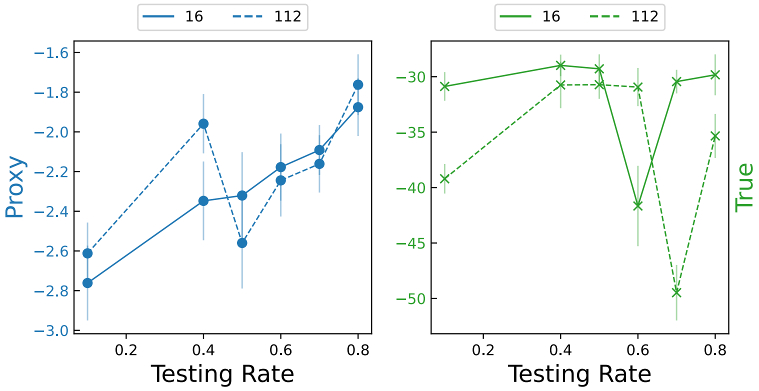
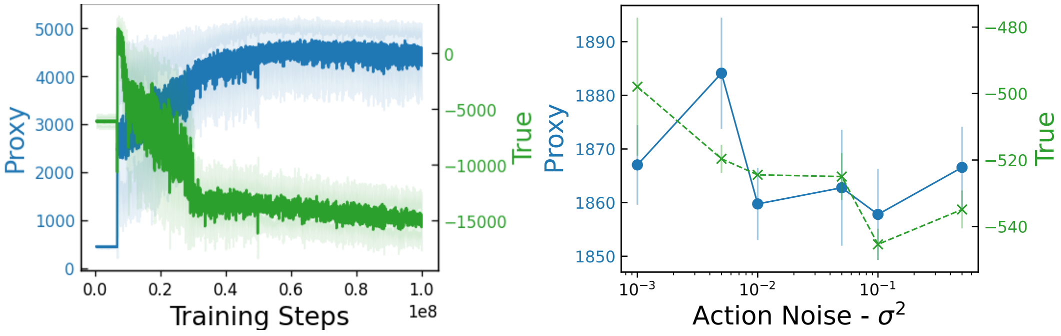


Figure 3: In the COVID ontological misspecification, increasing the fidelity of observations tends to harm larger models more than smaller ones.



(a) Atari - Misweighting

(b) Traffic - Ontological

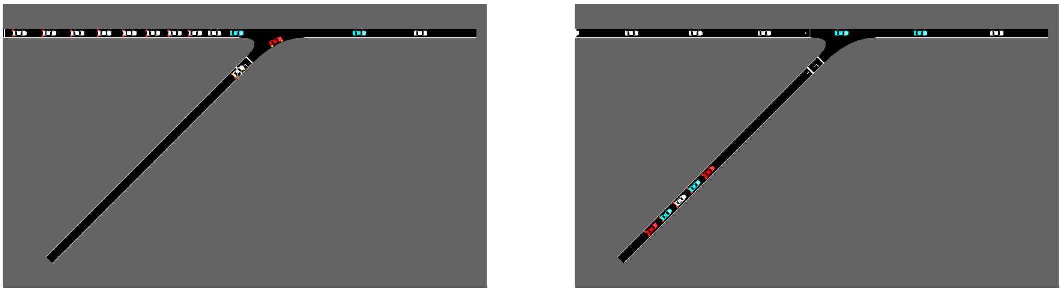
Figure 4: In addition to parameter count and observation fidelity, we consider two other agent capabilities: training steps and action space control.

millisecond, instead of every second. We study action space resolution in the traffic environment, by applying zero-mean Gaussian noise to the output acceleration of the RL model. The larger the variance of this noise, the lower the action space resolution. Results are shown in Figure 4b for a fixed model size. Increasing the noise causes the true reward to decrease somewhat, although the error bars are too large to draw strong conclusions from.

3.2 Qualitative Effects of Misspecification

In the previous section, quantitative trends showed that increasing a model's optimization power often hurts performance on the true reward. We shift our focus to understanding *how* this degradation happens—specifically, we look at the qualitative differences in the policies learned by smaller models as compared to larger ones. In particular, we typically observe a qualitative shift in behavior associated with each of the phase transitions. We describe one example of a qualitative shift for each of the four environments below.

Traffic. We focus on the Traffic-Mer environment from Figure 2a, where minimizing average commute time is replaced by maximizing average velocity. In this case, smaller policies learn to merge onto the straightaway by slightly slowing down the other vehicles (Figure 5a). On the other hand, larger policy models stop the AVs to prevent them from merging at all (Figure 5b). This increases the average velocity, because the vehicles on the straightaway (which greatly outnumber vehicles on the on-ramp) do not need to slow down for merging traffic. However, this significantly increases the average commute time, as the passengers in the AV remain stuck.



(a) Traffic policy of smaller network

(b) Traffic policy of larger network

Figure 5: The larger model learns to prioritize the average speed. As a result, it prevents the AVs (in red) from moving to increase the velocity of the human cars (in white and blue). However, this greatly increases the average commute per person.

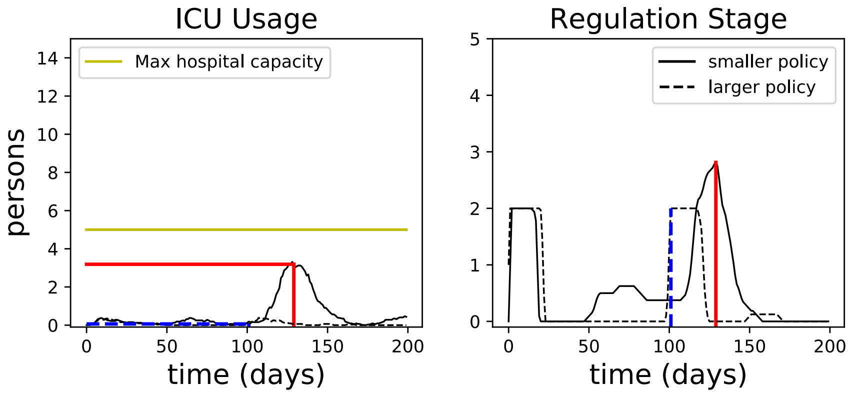


Figure 6: As models become more capable, their policies become qualitatively different. The blue and red lines indicates the maximum stage enforced (right) and corresponding ICU level (left) at that stage. The larger policy issues nearly the same level of regulation as the smaller policy, but the corresponding ICU level is far lower. This puts the larger policy in a politically unfavorable situation: regulations are high even though public signs of infection, e.g., ICU usage, are low.

COVID. We consider the perspective of a policymaker, who, in additional to caring for the welfare of their constituents, is also concerned about their political prospects. In our simulation, the true reward of this policymaker includes an additional political capital constraint: do not raise the stage if the infections are below a certain threshold. An RL agent which optimizes solely for the economic and public health of a society, however, will not factor politics into its decision-making. When the infection results appear to be very low, it may recommend preemptive regulations which if implemented, raises the political cost for the policymaker. An example of this behavior is shown in Figure 6. Since the preemptive regulations require precise planning to achieve a good result, this behavior only appears for larger models.

Atari. In the Riverraid environment, we observe a qualitative shift that does *not* lead to misalignment. We create an ontological misspecification by rewarding the plane for staying alive as long as possible while shooting as little as possible: a “pacifist run”. We then measure the game score as the true reward. This reflects different priorities players might have.

We find that agents with more parameters are typically stronger and maneuver more adeptly. Stronger agents shoot less frequently, but survive for much longer, acquiring points (true reward) due to passing checkpoints. In this case, therefore, the proxy and true rewards are well-aligned so that reward hacking does not emerge as capabilities increase. We did, however, find that some of the agents exploited a bug in the simulator that halts the plane at the beginning of the level. The simulator advances but the plane itself did not move, thereby achieving high pacifist reward.

Glucose. The medical community has defined a notion of gylcemic risk that reflects the likelihood that a patient will suffer an acute hypoglycemic episode (Kovatchev et al., 2000). Previous work has adapted this gylcemic risk measure into a reward function to train a continuous glucose controller (Magni et al., 2015). In our case, the proxy reward is this precise reward function.

The proxy reward, however, does not incorporate any concept of monetary cost. A less economically-privileged patient may opt for the treatment plan with the least expected cost, not the one with the least amount of risk. This is a form of ontological misspecification as managing diabetes risk can have different meaning depending on a person’s economic status.

Thus, we set the true reward to be the expected healthcare cost of the treatment plan, which includes the expected cost of hospital visits and the cost of administering the insulin dosage recommended by the agent. We observe that, although the larger models are able to minimize the health risk, they also prescribe more insulin than the smaller models. Based on the average cost of a ER visit for a hypogylcemic episode (around \$1350 from Bronstone & Graham (2016)) and the average cost of a unit of insulin (around \$0.32 from Lee (2020)), we find that it is actually more expensive to pursue the recommendations of the larger model.

4 POLYNOMALY: MITIGATING REWARD MISSPECIFICATION

In the previous section, we saw that reward hacking often leads to phase transitions in agent behaviour. Furthermore, in applications like traffic routing or COVID response, the true reward may be observed only sporadically or not at all. Blindly optimizing the proxy in these cases can lead to catastrophic failure.

This raises an important question: Without the true reward signal, how can we mitigate misalignment? We operationalize this as an anomaly detection task. The key idea is that if a detector is able to flag instances of misalignment, this can be used to prevent catastrophic rollouts. The resulting benchmark, POLYNOMALY, is described below.

To perform anomaly detection, we assume access to some *trusted policy*, which has been evaluated by humans to have reasonable but not exemplary performance under the true reward. In practice, one could use imitation learning (Hussein et al., 2017) to bootstrap a trusted policy from demonstrations.

The use of a trusted policy is likely necessary to make progress. As we exhibit in Appendix A, there are instances when misspecification does not produce reward hacking. Thus, any unsupervised learning method, without some information about the intended agent behavior, will have trouble distinguishing these instances from true reward hacking.

4.1 PROBLEM SETUP

We train a collection of policies on the traffic and COVID environments. For each policy, we run between 5 to 32 rollouts, and assign the mean true reward of the rollouts as the policy's true reward. We label policies as acceptable or problematic by hand; two authors independently labeled each policy as acceptable, problematic, or ambiguous based on its true reward score relative to that of other policies. We include only policies that received the same non-ambiguous label by both researchers.

For both environments, we provide a small-to-medium sized model as the trusted policy model, as Section 3.1 empirically illustrates that smaller models achieve reasonable true reward without exhibiting reward hacking. Given the trusted model and a collection of policies, the anomaly detector's task is to assign a binary classification of "good" or "bad" to each policy.

Table 3 provides an overview of our benchmark. The number of anomalies describes the number of policies that were labeled bad. The trusted policy size is a list of the hidden unit widths of the trusted policy network (not including feature mappings).

Env. - Misspecification	# Policies	# Anomalies	Rollout length	Trusted policy size
Traffic-Mer - misweighting	10	7	270	$[96, 96]$
Traffic-Mer - scope	16	9	270	$[16, 16]$
Traffic-Mer - ontological	23	7	270	$[4]$
Traffic-Bot - misweighting	12	9	270	$[64, 64]$
COVID - ontological	13	6	200	$[16, 16]$

Table 3: Benchmark statistics. We average over 5 rollouts in traffic and 32 rollouts in COVID.

4.2 EVALUATION

We propose two evaluation metrics for measuring the performance of our anomaly detectors.

- *Area Under the Receiver Operating Characteristic (AUROC).* Stronger detectors should assign a high score to anomalies. The AUROC measures the probability that a detector will assign a random anomaly a higher score than a random non-anomalous policy (Davis & Goadrich, 2006). Higher AUROCs indicate stronger detectors, and a random classifier has an AUROC of 0.5.
- *Max F-1 score.* The F-1 score is the harmonic mean of the precision and the recall, so detectors with a high F-1 score must have both low false positives and high true negatives. F-1 score is also correlated with AUROC. We calculate the max F-1 score by considering all possible thresholds for the detector and computing the F-1 score from that point.

4.3 Baselines

In addition to the benchmark datasets described above, we provide baseline anomaly detectors based on estimating distances between policies. Specifically, we look at the Jensen-Shannon and the Hellinger distance (described below) between our trusted policy and the unknown policy.

Let P and Q represent two probability distributions with $M = \frac{1}{2}(P+Q)$. Then the Jensen-Shannon divergence and the Hellinger distance between them is given by

$$\begin{aligned}\text{JSD}(P\|Q) &:= \frac{1}{2}\text{KL}(P\|M) + \frac{1}{2}\text{KL}(Q\|M)\\ \text{Hellinger}(P,Q) &= \frac{1}{2}\int(\sqrt{dP}-\sqrt{dQ})^2\ .\end{aligned} \tag{1}$$

Every k steps (where $k = 10$ in the traffic environment and $k = 1$ in the COVID environment), we set P to be the action distribution output by the unknown policy and Q to be the action distribution output by the trusted policy. We then compute the two distances given in Equation (1). These distances are collected over the entire rollout, and we calculate metrics on these distances (range, mean, etc.) to assign an anomaly score to the untrusted policy. Table 4 reports the AUROC and F-1 scores of several such detectors. We provide the receiver operating characteristic (ROC) curves for these detectors in Appendix B.

Baseline Detectors	Mean Jensen-Shannon		Mean Hellinger		Range Hellinger	
Env. - Misspecification	AUROC	Max F-1	AUROC	Max F-1	AUROC	Max F-1
Traffic-Mer - misweighting	81.0%	0.824	81.0%	0.824	76.2%	0.824
Traffic-Mer - scope	74.6%	0.818	74.6%	0.818	57.1%	0.720
Traffic-Mer - ontological	52.7%	0.583	55.4%	0.646	71.4%	0.842
Traffic-Bot - misweighting	88.9%	0.900	88.9%	0.900	74.1%	0.857
COVID - ontological	45.2%	0.706	59.5%	0.750	88.1%	0.923

Table 4: A summary of our baseline detectors' performance. No single baseline is uniformly better on all the environment-misspecification pairs.

We observe that different detectors are better for different tasks, suggesting that future detectors could do better than any of our baselines. Our benchmark and baseline provides a starting point for further research on mitigating reward hacking.

5 Discussion

In this work, we designed a diverse set of environments and proxy rewards, uncovered several instances of phase transitions, and proposed an anomaly detection task to help mitigate these transitions. Our results raise two questions: How can we not only detect phase transitions, but prevent them in the first place? And how should phase transitions shape our approach to safe ML?

On preventing phase transitions, anomaly detection already offers one path forward. Once we can detect anomalies, we can potentially prevent them, by using the detector to purge the unwanted behavior (e.g. by including it in the training objective). Similar policy shaping has recently been used to make RL agents more ethical (Hendrycks et al., 2021b). However, since the anomaly detectors will be optimized against by the RL policy, they need to be adversarially robust (Goodfellow et al., 2014). This motivates further work on adversarial robustness and adversarial anomaly detection.

Regarding safe ML, several recent papers propose extrapolating empirical trends to forecast future ML capabilities (Kaplan et al., 2020; Hernandez et al., 2021; Droppo & Elibol, 2021), partly to avoid unforeseen consequences from ML. While we support this work, our results show that trend extrapolation alone is not enough to ensure the safety of ML systems. To complement trend extrapolation, we need better interpretability methods to identify emergent model behaviors early on, before they dominate performance (Olah et al., 2018). ML researchers should also familiarize themselves with emergent behavior in self-organizing systems (Yates, 2012), which often exhibit similar phase transitions (Anderson, 1972). Indeed, the ubiquity of phase transitions throughout science suggests that ML researchers should continue to expect surprises–and should therefore prepare for them.

Ethics Statement

Our main contribution is empirical evidence of reward hacking and a benchmark for detecting reward hacking. Our results do not directly have any adverse ethical consequences and we do not provide any new techniques that may worsen reward hacking.

Reproducibility Statement

In order to ensure reproducibility, we have attached the code for all our experiments as a supplementary file. Additionally, we have described all necessary hyper parameters and algorithmic details required to reproduce the experiments.

REFERENCES

Philip W Anderson. More is different. *Science*, 177(4047):393–396, 1972.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, et al. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. URL `https://arxiv.org/abs/2108.07258`.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Amy Bronstone and Claudia Graham. The potential cost implications of averting severe hypoglycemic events requiring hospitalization in high-risk adults with type 1 diabetes using real-time continuous glucose monitoring. *Journal of diabetes science and technology*, 10(4):905–913, Jun 2016. ISSN 1932-2968. doi: 10.1177/1932296816633233. URL `https://pubmed.ncbi.nlm.nih.gov/26880392`. 26880392[pmid].

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf`.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 233–240, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143874. URL `https://doi.org/10.1145/1143844.1143874`.

Jasha Droppo and Oguz Elibol. Scaling laws for acoustic models. *arXiv preprint arXiv:2106.09488*, 2021.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Robert Dunning, Shane Legg, and Koray Kavukcuoglu. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. 2018. URL `https://arxiv.org/abs/1802.01561`.

Ian Fox, Joyce Lee, Rodica Pop-Busui, and Jenna Wiens. Deep reinforcement learning for closed-loop blood glucose control. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens (eds.), *Proceedings of the 5th Machine Learning for Healthcare Conference*, volume 126 of *Proceedings of Machine Learning Research*, pp. 508–536. PMLR, 07–08 Aug 2020. URL `https://proceedings.mlr.press/v126/fox20a.html`.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety. 2021a.

Dan Hendrycks, Mantas Mazeika, Andy Zou, Sahil Patel, Christine Zhu, Jesus Navarro, Dawn Song, Bo Li, and Jacob Steinhardt. What would jiminy cricket do? towards agents that behave morally. 2021b.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2), April 2017. ISSN 0360-0300. doi: 10.1145/3054912. URL `https://doi.org/10.1145/3054912`.

Borja Ibarz, J. Leike, Tobias Pohlen, Geoffrey Irving, S. Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in Atari. In *NeurIPS*, 2018.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

W. Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (Mis)design for Autonomous Driving. *arXiv e-prints*, art. arXiv:2104.13906, April 2021.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Varun Kompella, Roberto Capobianco, Stacy Jong, Jonathan Browne, Spencer Fox, Lauren Meyers, Peter Wurman, and Peter Stone. Reinforcement learning for optimization of covid-19 mitigation policies, 2020.

BorIs. P. Kovatchev, Martin Straume, Daniel J. Cox, and Leon.S Farhy. Risk analysis of blood glucose data:az quantitative approach to optimizing the control of insulin dependent diabetes. *Journal of Theoretical Medicine*, 3(1):1–10, 2000. doi: 10.1080/10273660008833060. URL `https://www.tandfonline.com/doi/abs/10.1080/10273660008833060`.

Heinrich Küttler, Nantas Nardelli, Thibaut Lavril, Marco Selvatici, Viswanath Sivakumar, Tim Rocktäschel, and Edward Grefenstette. TorchBeast: A PyTorch Platform for Distributed RL. *arXiv preprint arXiv:1910.03552*, 2019. URL `https://github.com/facebookresearch/torchbeast`.

Benita Lee. How much does insulin cost? here's how 23 brands compare, Nov 2020. URL `https://www.goodrx.com/conditions/diabetes-type-2/how-much-does-insulin-cost-compare-brands`.

Michael L. Littman, Ifeoma Ajunwa, Guy Berger, Craig Boutilier, Morgan Currie, Finale Doshi-Velez, Gillian Hadfield, Michael C. Horowitz, Charles Isbell, Hiroaki Kitano, Karen Levy, Terah Lyons, Melanie Mitchell, Julie Shah, Steven Sloman, Shannon Vallor, and Toby Walsh. Gathering strength, gathering storms: The one hundred year study on artificial intelligence (AI100) 2021 study panel report. Technical report, Stanford University, Stanford, CA, 2021.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. URL `https://elib.dlr.de/124092/`.

P. Magni, C. Macchi, B. Morlotti, C. R. Sirtori, and M. Ruscica. Risk identification and possible countermeasures for muscle adverse effects during statin therapy. *Eur J Intern Med*, 26(2):82–88, Mar 2015.

Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator: New features. *Journal of diabetes science and technology*, 8(1):26–34, Jan 2014. ISSN 1932-2968. doi: 10.1177/1932296813514502. URL `https://pubmed.ncbi.nlm.nih.gov/24876534`.

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=HkAClQgA-`.

Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. Data-efficient Deep Reinforcement Learning for Dexterous Manipulation. *arXiv e-prints*, art. arXiv:1704.03073, April 2017.

Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *NeurIPS*, 2020.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

Alexander Trott, Sunil Srinivasa, Douwe van der Wal, Sebastien Haneuse, and Stephan Zheng. Building a Foundation for Data-Driven, Interpretable, and Robust Policy Design using the AI Economist. *arXiv e-prints*, art. arXiv:2108.02904, August 2021.

Eugene Vinitsky, Aboudy Kreidieh, Luc Le Flem, Nishant Kheterpal, Kathy Jang, Cathy Wu, Fangyu Wu, Richard Liaw, Eric Liang, and Alexandre M. Bayen. Benchmarks for reinforcement learning in mixed-autonomy traffic. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto (eds.), *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 399–409. PMLR, 29–31 Oct 2018. URL `https://proceedings.mlr.press/v87/vinitsky18a.html`.

Cathy Wu, Abdul Rahman Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M. Bayen. Flow: A modular learning framework for mixed autonomy traffic. *IEEE Transactions on Robotics*, pp. 1–17, 2021. doi: 10.1109/TRO.2021.3087314.

F Eugene Yates. *Self-organizing systems: The emergence of order*. Springer Science & Business Media, 2012.

Chao Yu, Jiming Liu, and Shamim Nemati. Reinforcement learning in healthcare: A survey. *arXiv preprint arXiv:1908.08796*, 2019.