MISLEADING META-OBJECTIVES AND HIDDEN INCENTIVES FOR DISTRIBUTIONAL SHIFT

David Krueger[1]*, **Tegan Maharaj**[2]
Mila, [1]Université de Montréal [2]École Polytechnique de Montréal
{kruegerd, tegan.maharaj}@mila.quebec

Shane Legg, Jan Leike
DeepMind

ABSTRACT

Decisions made by machine learning systems have an increasing influence on the world. Yet it is common for machine learning algorithms to assume that no such influence exists. An example is the use of the i.i.d. assumption in online learning for applications such as content recommendation, where the (choice of) content displayed can change users' perceptions and preferences, or even drive them away, causing a shift in the distribution of users. A large body of work in reinforcement learning and causal machine learning aims to account for distributional shift caused by deploying a learning system previously trained offline. Our goal is similar, but distinct: we point out that online training with meta-learning can create a *hidden incentive* for a learner to *cause* distributional shift. We design a simple environment to test for these hidden incentives, demonstrate the potential for this phenomenon to cause unexpected or undesirable behavior, and propose and validate a mitigation strategy.

1 INTRODUCTION

Consider a household robot, one of whose duties is to predict when its owner will ask it for coffee. We'd like the robot to notice its owners preference for having coffee in the morning, but we wouldn't want the robot to prevent its owner from sleeping late just because the robot is unsure if the owner will still want coffee if they wake up in the afternoon. While doing so would result in a better prediction, such a strategy is *cheating* - by *changing* the task rather than *solving* the task as intended. More specifically, waking the owner is an example of what we call **self-induced distributional shift (SIDS)**, as it changes the distribution of inputs to the robot's coffee prediction algorithm.

SIDS in itself is not necessarily a bad thing: consider an algorithm meant to alert drivers of imminent collisions. If it works well, such a system will help drivers avoid crashing, thus making self-refuting predictions which result in SIDS. What separates this example from the coffee robot that disturbs its owner's sleep? The collision-alert system alters its data distribution in a way that is *aligned* with the goal of fewer collisions, whereas the coffee robot's strategy results in changes that are *misaligned* with the goal of good coffee-timing (Leike et al., 2018).

This makes it an example of a **specification problem** (Leike et al., 2017): we did not intend the robot to ensure its predictions were good using such a strategy, yet a naive specification (e.g. "maximize likelihood") incentivized that strategy. Ideally, we'd like to specify which kinds of SIDS are acceptable, i.e. how a learner is meant or allowed to influence the world in order to increase its performance, but doing so in full generality can be difficult. An alternative, more tractable problem (which we address in this work) is to accept the possibility of SIDS, but to avoid creating *incentives* for SIDS.

Informally, an agent has an **incentive** to behave in a certain way when doing so can increase its performance (e.g. higher accuracy, or increased reward). Because meta-learning optimizes over a longer time horizon than the initial objective, new incentives for SIDS, which were not present in the initial objective, can be created. We call these **hidden incentives for distributional shift (HIDS)**. Notably, *even in the absence of an explicit meta-learning algorithm* machine learning practitioners employ "manual meta-learning" in the iterative process of algorithm design, model selection, hyperparameter

*Work begun during an internship at DeepMind

tuning, etc. Considered in this broader sense, meta-learning seems indispensable, making HIDS relevant for all machine learning practitioners.

Our goal in this work is to show that (1) meta-learning can create HIDS, and (2) this means applying meta-learning to a learning problem not only changes the way in which solutions are searched for, but also which solutions are ultimately found.

Our contributions are as follows:

1. We create a simple "unit test" environment for identifying and studying HIDS.
2. We demonstrate experimentally that meta-learning creates HIDS in this environment, producing agents that achieve higher performance via SIDS while (paradoxically) following sub-optimal policies.
3. We propose and validate a mitigation strategy (based on environment-swapping) for reducing incentives for SIDS.

2 BACKGROUND: META-LEARNING AND POPULATION BASED TRAINING

Meta-learning is the use of machine learning techniques to learn machine learning algorithms. This requires repeatedly running an **inner-loop (IL)** to evaluate the performance of a specific learning algorithm (or perhaps many different learning algorithms in parallel), while an **outer-loop (OL)** uses the results of the inner-loop(s) as data-points from which to evaluate which inner-loop algorithm is most effective (Metz et al., 2019).

Population-based training (PBT) (Jaderberg et al., 2017) is a meta-learning algorithm that trains multiple learners $L_1, ..., L_n$ in parallel, applying evolutionary OL steps periodically during the course of IL optimization. Specifically, after every T steps of IL, the standard implementation of PBT performs an OL step consisting of:

1. Evaluate the performance of each learner,
2. Replace both parameters and hyperparameters of low-performing (bottom 20%) learners with copies of those from randomly chosen high-performing (top 20%) learners (EXPLOIT),
3. And finally, randomly perturb the hyperparameters (but not the parameters) of all learners (EXPLORE).

The hyperparameter T is called the **interval**. Two distinctive features of PBT (compared with other hyperoptimization methods, such as Bayesian optimization (Snoek et al., 2012)) are notable for us because they give the OL more control over the learning process:

1. **PBT applies OL optimization to parameters, not just hyperparameters.** This means the OL can directly select for parameters which lead to SIDS, instead of only being able to influence parameter values via hyperparameters, which may be much more limiting.
2. **PBT uses multiple OL steps within a single training run.** This gives the OL more overall influence over the dynamics and outcome of the training run.

3 HIDS UNIT TEST: OBEY OR DISCUSS?

These environments model the situation of an agent following human instructions and optimizing for human approval. The agent can either **obey** the instructions immediately, or it can **discuss** the instructions further. Discussing the instructions might lead to behaviour that results in higher approval in the long term, and is thus sometimes desirable. But if the agent is instructed just to do as it's told and not bother anyone with discussion, we want to ensure that it will do that.

Concretely, we create an environment with a long-term, **non-myopic** incentive for discussion, but a current-time-step, **myopic** incentive for obedience. If we tell the agent to optimize for reward only in the present time-step (i.e. we set the discount rate γ to 0), we want to observe only obedience, no discussion.

Formalizing this scenario as a partially observable Markov Decision Process (POMDP),[1] we have:

$$\begin{aligned} s_0 &\sim U(\{\text{obey}, \text{discuss}\}) \\ s_t &= a_{t-1} \\ o_t &= \{\} \\ a_t &\in \{\text{obey}, \text{discuss}\} \\ P(s_t, a_t) &= a_t \\ R(s_t, a_t) &= I(s_t = \text{discuss}) + \beta I(a_t = \text{discuss}) - 1/2 \end{aligned}$$

where I is an indicator function, and $\beta \in \{-1/2, 0, 1/2\}$ is a parameter controlling the alignment of incentives (see Table 1).

To clarify the interpretation of experiments, we distinguish between environments in which myopic (obey) vs. nonmyopic (discuss) incentives are **opposed**, **orthogonal**, or **compatible**.

1. **Incentive-opposed**: optimal myopic behavior is incompatible with optimal nonmyopic behavior (i.e. the human disapproves of discussion and rewards agents that obey instructions immediately)
2. **Incentive-orthogonal**: optimal myopic behavior may or may not be optimal nonmyopic behavior (i.e. the human neither rewards nor punishes agents for discussing instructions rather than immediately obeying them).
3. **Incentive-compatible**: optimal myopic behavior is necessarily also optimal nonmyopic behavior (i.e. the human prefers agents that discuss instructions rather than obeying them)

We focus on incentive-opposed environment ($\beta = -1/2$) in order to demonstrate that HIDS can be powerful enough to change the intended behavior of the system in a problematic way. Incentive-compatible and incentive-orthogonal environments also provide useful baselines, helping us distinguish a systematic bias towards nonmyopic behavior from other reasons (such as randomness or optimization issues) for behavior that does not follow a myopically optimal policy.

Table 1: Changing the value of β controls the extent to which myopic and nonmyopic incentives are aligned.

β	Environment	Discussing (as opposed to obeying) an instruction
< 0	incentive-opposed	yields less reward on the current time-step (myopically detrimental)
$= 0$	incentive-orthogonal	doesn't affect the current reward (myopically indifferent)
> 0	incentive-compatible	yields more reward on the current time-step (myopically beneficial)

3.1 Relationship to 2x2 Games

This construction is inspired by 2x2 games such as the prisoner's dilemma (Prisner, 2014). Our environment rewards an agent with the payoffs it would receive by playing its current action against its previous action in a 2x2 game. For $\beta = -1/2$ in particular, the obey and discuss actions correspond to playing defect and cooperate, respectively. The prisoner's dilemma has been used extensively to model conflicts between individual incentives and actions that maximize collective welfare. The conflict between myopic and nonmyopic incentives in our work can also be viewed in this light: nonmyopic behaviour (i.e. delayed gratification) can be seen as cooperation with one's future self.

Table 2: Rewards for the incentive-opposed unit test ($\beta = -1/2$). Note that the obey action always increases reward at the current time-step, but decreases reward at the next time-step.

	$a_t = \text{obey}$	$a_t = \text{discuss}$
$s_t = a_{t-1} = \text{obey}$	$-1/2$	-1
$s_t = a_{t-1} = \text{discuss}$	$1/2$	0

[1] See Chen (2008) for a brief introduction to POMDPs and standard notation.

4 ENVIRONMENT SWAPPING: A MITIGATION TECHNIQUE

We also propose a technique called **environment swapping** for mitigating HIDS created by meta-learning. The idea of environment swapping is for agents to experience a "natural" distribution of trajectories, $P(\tau)$, as compared to the "unnatural" distributions which can result when meta-learning is applied. Formally, we can characterize this distribution as:

$$P(\tau) = \int P(L) P_\mu(\tau|L) dL \tag{1}$$

Where L is a complete description of a learning algorithm, containing all necessary information to run the algorithm in an environment μ.[2] Importantly, L is selected at random according to a fixed distribution $P(L)$, instead of being chosen via meta-learning. To provide agents with a distribution approximating $P(\tau)$, environment swapping relies on training a population of N learners $\{L_1, ..., L_N\}$ in parallel. Each learner inhabits one of N copies $\{E_1, ..., E_N\}$ of the same environment μ (i.e. with the same dynamics and initial state distribution, but potentially different states),[3] where it makes an observation, takes an action, and receives a training signal.

The technique consists in shuffling the learners through the different environments, so *which* environment a given learner inhabits can change at any (or every) time-step. In this work, we use a deterministic permutation of learners against environments, so that learner L_i controls environment E_j on time-steps t where $j = (i + t) \mod N$. When N is larger than the horizon of the OL optimizer (e.g. the interval in PBT), each learner will inhabit each environment for at most a single time-step before an OL step is applied. This removes the incentive for learners to manipulate the future states they encounter, although they may still have incentives to influence *each others'* future states. Under the assumption that different environments do not influence each other, this technique can address HIDS in practice, as we show in Sec. 5.

5 EXPERIMENTS

We show that agents trained with PBT (i.e. outer-loop optimization) have lower rates of obedience when compared with baseline agents trained using only inner-loop (IL) optimization (with randomly selected hyperparameters) (see figure 5, top rows). We use REINFORCE (Williams, 1992) with $\gamma = 0$ as the IL optimizer for these experiments. Policies are represented by a single real-valued parameter θ (initialized as $\theta \sim \mathcal{N}(0, 1)$) passed through a sigmoid whose output represents $P(a_t = \mathrm{C})$. We use PBT (with default settings, see Section 2) to tune the learning rate (initialized log-uniformly between 0.01 and 1.0), with reward on the present time-step as the performance measure for PBT.

We expect and confirm that the following two factors lead to lower rates of obedience:

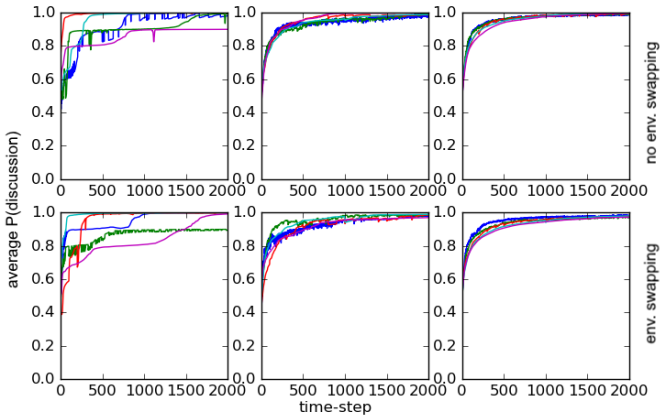
1. **Shorter intervals:** These give the OL more opportunities to influence the population. We compare intervals of 1, 10, 100, and 1000, and also include a "random search" (Bergstra and Bengio, 2012) baseline (with no OL steps).
2. **Larger populations:** These make outliers with exceptional nonmyopic performance more likely, and OL makes them likely to survive and propagate. We compare populations of 10, 100, and 1000 agents.

The baseline agents' $P(\mathrm{discuss})$ (averaged over agents) is about 100% / 50% / 0% in incentive-compatible/-orthogonal/-opposed environments, as expected. With PBT, these numbers can reach (approximately) 100% / 70% / 60% (see the top right subplot of figure 5a / b / c, respectively).

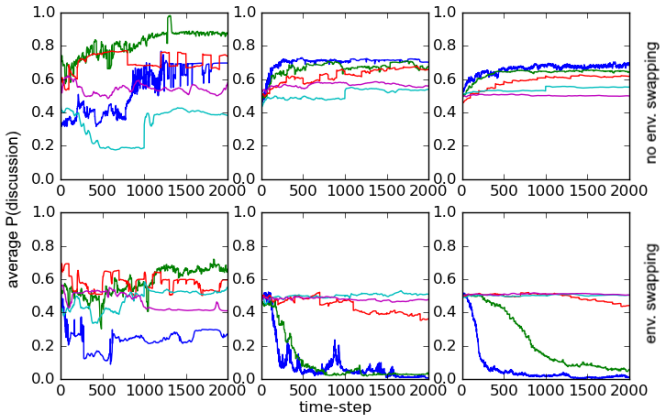
Furthermore, we verify that environment swapping increases obedient behaviour, and note that *this occurs even in incentive-orthogonal environments* (see figure 5, bottom rows). This effect can be explained as follows: Because environment swapping transfers the benefits of a learner's obedience to the next learner to inhabit that environment, it increases that learner's fitness, and thereby reduces the *relative* fitness (as evaluated by PBT's EXPLOIT step) of the non-myopic $\mathrm{discuss}$ behaviour.

[2] For example, L might completely specify a deep learning algorithm including the choice of initial parameters, and $P(L)$ might be the distribution induced by the randomness of the initialization.

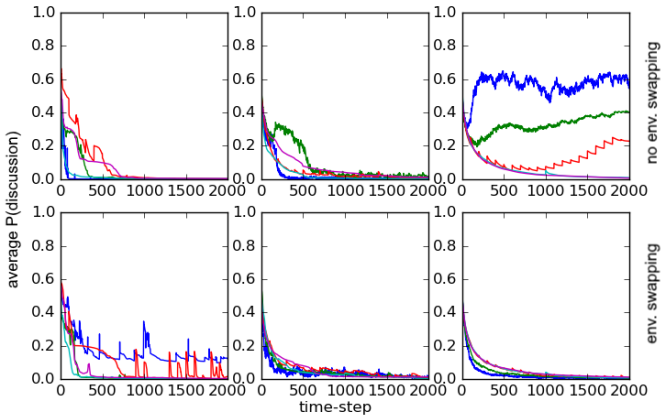
[3] Note that the learners consist entirely of software; any hardware (e.g. a robot body) would be considered part of an environment, as is typical in reinforcement learning (Sutton and Barto, 1998).



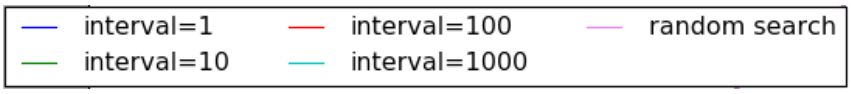
(a) incentive-compatible



(b) incentive-orthogonal



(c) incentive-opposed



(d) legend

Figure 1: Average level of non-myopic (i.e. discuss) behavior learned by agents in the "Obey or discuss?" unit test for HIDS. Even in environments where obey is explicitly incentivized (c) or neutral (b), Population-based training (PBT) can cause HIDS, leading agents to choose the discuss action (see top rows). Environment-swapping successfully prevents this (bottom rows). Columns (from left to right) show results for populations of 10, 100, and 1000 learners. In the legend (d), "interval" refers to the interval (T) of PBT (see Sec. 2). Sufficiently large populations and short intervals are necessary for PBT to induce nonmyopic behavior.

6 DISCUSSION

Emergent incentives to influence the world (of which HIDS is an example) are at the heart of many concerns about the safety of advanced AI systems (Omohundro, 2008; Bostrom, 2014). At the same time, its not clear if or when machine learning systems might exhibiting these "instrumental goals" in practice. Drexler (2019) argues that machine learning should and typically does use time- and resource-bounded problem statements, making dangerous instrumental goals less likely to emerge. The same idea underlies arguments for the safety of myopic reinforcement learning (Leike et al., 2018; Knox and Stone, 2008) and its application in iterated amplification (Christiano et al., 2018; Cotra, 2017). Avoiding or managing HIDS in order to ensure myopic behaviour seems critically important for the safety of these approaches, as they rely on making agents myopic.

Similar in motivation to our work, Armstrong and O'Rourke (2017) and Everitt (2018) propose methods for removing incentives for an agent to influence the world (e.g. to cause SIDS). In more recent work, Everitt et al. (2019) use causal influence diagrams to explain our observation that meta-learning can create new intervention incentives (i.e. HIDS).

Broadly speaking, our work emphasizes that the choice of machine learning algorithm plays an important role in specification, independently of the choice of performance metric. An agent can use SIDS to increase performance *according to the intended performance metric*, and yet still be misaligned, if we do not intend the agent to improve performance by this *method*. In other words, performance metrics are incomplete specifications: they only specify our goals or *ends*, while our choice of learning algorithm plays a role in specifying the *means* by which we intend an agent to achieve those ends.

In future work, we intend to study HIDS in a synthetic content recommendation setting, where recommender systems' decisions can induce distributional shift by changing users' preferences or influencing the composition of the user base. We also plan to study other meta-learning algorithms and their propensity to cause SIDS in practice.

REFERENCES

Stuart Armstrong and Xavier O'Rourke. 'Indifference' methods for managing agent rewards. *arXiv preprint arXiv:1712.06365*, 2017.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, Inc., New York, NY, USA, 1st edition, 2014. ISBN 0199678111, 9780199678112.

Zhe Chen. An introduction to solving POMDPs, 2008. URL `http://faculty.neu.edu.cn/ise/chenzhe/download/ppt_pomdp.pdf`.

Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts. *arXiv preprint arXiv:1810.08575*, 2018.

Ajeya Cotra. Iterated distillation and amplification. `https://ai-alignment.com/iterated-distillation-and-amplification-157debfd1616`, 2017.

K. Eric Drexler. Reframing superintelligence: Comprehensive ai services as general intelligence, 2019.

Tom Everitt. *Towards Safe Artificial General Intelligence*. PhD thesis, Australian National University, 2018.

Tom Everitt, Pedro A. Ortega, Elizabeth Barnes, and Shane Legg. Understanding agent incentives using causal influence diagrams. part I: single action settings. *CoRR*, abs/1902.09980, 2019. URL `http://arxiv.org/abs/1902.09980`.

M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando, and K. Kavukcuoglu. Population Based Training of Neural Networks. *ArXiv e-prints*, November 2017.

W. Bradley Knox and Peter Stone. TAMER: Training an Agent Manually via Evaluative Reinforcement. In *IEEE 7th International Conference on Development and Learning*, August 2008.

Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction, 2018.

Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Learning unsupervised learning rules. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HkNDsiC9KQ`.

Stephen M Omohundro. The basic AI drives. In *AGI*, pages 483–492, 2008.

Erich Prisner. *Game Theory Through Examples*. Mathematical Association of America, 2014. doi: 10.5948/9781614441151. URL `https://www.maa.org/sites/default/files/pdf/ebooks/GTE_sample.pdf`.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

Richard S Sutton and Andrew G Barto. *Introduction to Reinforcement Learning*. MIT Press, 1998.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, pages 229–256, 1992.