Evolutionary Computation and AI Safety:
Research Problems Impeding Routine and Safe
Real-world Application of Evolution

Joel Lehman
Uber AI
joel.lehman@uber.com

Abstract Recent developments in artificial intelligence and machine learning have spurred interest in the growing field of AI safety, which studies how to prevent human-harming accidents when deploying AI systems. This paper thus explores the intersection of AI safety with evolutionary computation, to show how safety issues arise in evolutionary computation and how understanding from evolutionary computational and biological evolution can inform the broader study of AI safety.

1 Introduction

As the capabilities and pervasiveness of machine learning (ML) and artificial intelligence (AI) increasingly affect society, there is increasing concern about the safety of such systems, i.e. the potential of accidental harm from implementation errors and unintended consequences in ML algorithms. As a result, there has been increasing interest in the nascent field of AI safety [1, 2, 3, 4, 5, 6], which seeks to understand and solve the technical challenges in developing and deploying AI that does what it is intended to do. The purpose of this chapter is to explore how the study of AI safety intersects with that of evolutionary computation (EC), to both highlight an exciting and important set of safety problems within EC, and to suggest that evolution and EC have important insights that could benefit the general study of AI safety.

To frame the problem of AI safety, we adopt the framework of Amodei et al. [1], which defines AI safety as concerned with accidents in ML systems, and defines five problems within three broad categories of issues: (1) specifying the wrong objective function, (2) making safe and efficient use of a true but expensive objective (e.g. human feedback), and (3) how to improve or adapt safely while interacting with the real world. A running example in that paper, which we adopt here, describes a robot with the task of cleaning an office using common tools; we modify the example to assume that the controller for this robot has been evolved, i.e. with an EC technique like neuroevolution or genetic programming (GP; [7]) in the setting of evolutionary
robotics (ER; \[8, 9\]). While this running example is posed in the reinforcement learning setting of ER, similar issues can arise whenever an EC-trained artifact interacts with the real world; for example, a credit-scoring system trained with GP symbolic regression (e.g. as in Ong et al. \[10\]) when deployed might have enact unintended consequences on the real-world borrowers its decisions affect, e.g. by potentially inferring (and biasing decisions from) ethically and/or legally-problematic borrower traits.

One motivation for this chapter is to draw attention within EC to a selection of interesting and important concrete research problems (as introduced by Amodei et al. \[1\]), in hopes of encouraging progress towards one of EC’s aspirations: to provide mature and reliably safe solutions for real-world AI and ML problems. As EC systems are increasingly trained, refined, and applied in the real-world, it becomes necessary to deal with real-world complications that are often side-stepped in closed-world research benchmarks; grappling with these issues is thus a necessary step for EC to transition into a reliable approach for safely solving real-world problems. For example, if evolution is occurring in an environment alongside humans (e.g. evolving a robot controller that interacts with people in an office setting) much care is needed to design an appropriate fitness function that at least does not cause harm in its early incarnations; in contrast, fitness functions in more traditional closed-world ER simulations often undergo many iterations of free-form debugging, with no real danger or cost (beyond wasted time and computation), where initial attempts often create highly-unexpected outcomes \[11\]. To enable reliable real-world deployment of EC, it may be useful to come up with new automated design procedures, to import tools from AI safety in statistical ML, or to perform new and directed EC research on solving technical safety problems.

A complementary motivation is to highlight problems or applications of AI safety for which EC techniques might be particularly well-suited to make significant contributions. For example, the subfields of quality diversity (QD; \[12, 13\]) and open-ended evolution \[14, 15\] might provide a natural mechanism to create a diverse set of test-scenarios to illuminate potential long-tail failures of ML systems that might otherwise go unidentified (e.g. the fooling images work of Nguyen et al. \[16\] shows how EC can automatically identify diverse visual patterns that a deep neural network will confidently misidentify). In other words, while most current AI safety work is conducted with traditional statistical ML (e.g. gradient-based deep learning approaches), EC might bring new ideas and techniques to bear on such problems.

A final motivation is to consider if and how natural evolution solved problems similar to those raised by AI safety. For example, evolution has designed various means of collaboration within and between species that can embody a minimization of negative side-effects among the behavior of social animals or mutualistic species (although of course, there are many examples of antagonistic behavior as well, e.g. the ubiquity of predator-prey relationship). Additionally, evolution has uncovered ways to “explore safely” (for some definition of safety) both across an evolutionary timescale (i.e. through the evolution of evolvability, i.e. evolution acting to improve variation) and an individual organism’s lifetime (i.e. through the complementary instincts of curiosity and fear \[17\]). The hope is that biological inspiration might
point the way towards potential solutions to these kinds of safety problems in EC or
in ML at large.

The conclusion is that AI safety is likely to be a growing field of interest in
coming years that offers a range of interesting technical challenges, and that EC
may both have important insights to offer and benefits to gain from research in that
community.

2 Background

The next sections describe the field of AI safety, and how EC is applied in the real
world, which helps to understand safety concerns from an EC perspective.

2.1 AI Safety

The field of AI safety [1, 2] seeks to pose and solve technical challenges involved
in developing AI that does what it is intended to do. The hope is to help foresee
and avoid harmful accidents that might result from good-intentioned AI gone astray.
While the name AI safety naturally evokes ideas of direct physical safety (e.g. how
to make sure there are sufficient guard-rails that prevent a robotic arm from acci-
cidentally hitting a human), the problems studied in AI safety also encompass more
abstract and broad concerns, ranging from the immediate and short-term (e.g. how
can a mobile robot driven by reinforcement learning continually improve its policy
by exploring, without taking any catastrophic actions, such as those that cause harm
to itself, to the environment, or humans), to speculative concerns about the far fu-
ture (e.g. how to make sure an AI that surpassed human intelligence would still be
controllable and aligned with our interests).

One central challenge in AI safety (of importance both to short and long-term
concerns) is known as the value alignment problem, i.e. how to align a computa-
tional agent’s incentives with our own. This problem might appear at first simple,
because as designers of agents we have complete control over their incentives. How-
ever, such alignment remains an unsolved technical challenge: Currently we do not
know how in practice to algorithmically specify (or learn from data) the complexity
of what humans care about, e.g. our moral intuitions, common-sense knowledge,
and cultural norms, all of which can potentially come to bear upon what we intend
for a computational agent to do. In other words, in EC there is yet no procedure to
specify a correct and complete fitness function that encompasses all the background
context that could be important for a system that interacts with humans and society.
More concretely, even for an AI system that interacts with the real world in very lim-
ited ways, it is still often a challenge to design a fitness function that truly measures
or incentivizes correct behavior [11]. Indeed, the typical paradigm in AI remains
to specify a fixed and relatively simple objective function (e.g. a fitness function in
EC) that is then optimized through search; however, as practitioners in EC are well-aware, an intuitive fitness function can often be optimized in unexpected ways [1]. While there exist candidate approaches to value alignment [6, 3, 4], the problem at core currently remains unsolved.

Interestingly, even if incentives are aligned, i.e. the learning system is provided with the correct objective function, how to successfully (and safely) optimize that objective function is still a difficult and unsolved problem in its own right. For example, a reinforcement learning agent that is given the correct objective to optimize can still make mistakes while it is being optimized (e.g. it can make harmful mistakes while exploring how to improve its policy); or, the objective might be challenging to optimize (e.g. it might instantiate a fitness landscape with many local optima), and the locally-optimal policies that search converges to might not be value-aligned.

One useful framework for categorizing technical challenges in AI safety comes from Amodei et al. [1], which divides safety problems into five categories: avoiding negative side effects, reward hacking, scalable oversight, safe exploration, and robustness to distributional shift. We adopt this framework in this paper for relating AI safety problems to EC and evolution, and later in this paper describe each of these problems in detail and how they emerge in EC.

One general consideration for AI safety is that it is most relevant when considering applying AI algorithms to real-world situations, where human well-being, broadly speaking (e.g. including not only physical safety, but also social harm from biased high-stakes decisions [18] or offense from insensitive classifications [19]), might be at stake. Thus the next section reviews common paradigms for applying EC to the real world.

2.2 EC and the Real World

While there are many different motivations for studying EC, including to understand the creative potential of biological evolution algorithmically for its own sake, researchers in EC often explicitly aim towards real-world applications of their ideas, or at least paint a viable path towards how algorithmic improvements might be translated into real-world value. Below we consider how such translation generally works in different learning settings (i.e. supervised vs. reinforcement learning).

When EC is applied to supervised learning settings, i.e. training where the task is to predict or classify over a labeled training set, it is important to recall that supervised training performance is rarely an end in itself; while improved accuracy on a benchmark is often a crucial consideration for publishing a paper for symbolic regression [20] or neural classification [21] models, such accuracy is only practically important insofar as it feeds into the downstream task the model is applied towards (e.g. a classification model of credit-worthiness might be applied to decide or influence loan decisions). While improved accuracy will likely contribute to such use cases, it will not take into account the nuances of the domain (e.g. the differing impact of different kinds of mistakes [19]).
Thus, while such applications of supervised learning might at first not seem relevant to AI safety, nearly always, the objective that a supervised-learning EC model is trained towards (e.g., classification accuracy) serves only as a proxy for the true downstream objective (e.g., efficient loan allocations that abide by legal and moral norms). Notably, limitations of such a proxy are well-known, e.g., the fairness, accountability, and transparency community within ML has highlighted how maximizing training accuracy can result in models that base decisions on morally-unacceptable criteria \cite{22,18}; this kind of gap between the proxy and the true objective can be seen as a manifestation of the general value alignment problem, and techniques for minimizing such a gap highlight how AI safety research can be relevant to EC-based supervised learning.

When EC is instead applied to robotics or reinforcement learning, evolution most often occurs within simulated environments, with the idea that policies trained in simulation can subsequently be transferred to reality \cite{23,24}, and potentially further evolved in the real world. The reasons for training in simulation include that real-world evaluations can be slow, tedious, and expensive, and can risk damage to hardware (like a robot) and to the broader environment (like humans coexisting with the robot); simulation enables much more convenient large-scale experimentation (given sufficient computation), although both how to design accurate simulations for complicated domains and how to successfully transfer policies from simulation to the real world remain challenging areas of research \cite{25,24,23}. Safety concerns in this paradigm can emerge from simulations that do not reveal safety-critical edge cases later encountered when models are deployed in the real-world, or from changing circumstances in reality (i.e., distributional shift) that are not accounted for in simulation. Another paradigm in EC is embodied evolution \cite{26}, wherein evolution is conducted in the real world, to circumvent the challenges of building accurate simulators and crossing the reality gap. In this setting, to the extent that evolved policies interact with humans or can damage their robotic body or their environment, there may be the need for expensive and potentially constant supervision (an AI safety issue discussed in more detail later). In general, because there is never the protective buffer of simulation between a policy and the real-world, safety considerations in embodied evolution may be more challenging than in other settings.

The conclusion is that as EC strives and achieves greater real-world impact, there will likely be a corresponding increased risk (albeit still potentially minor in many domains) of unintentional harm, no matter the specific paradigm by which EC models are trained and deployed.

### 3 EC and Concrete AI Safety Problems

This section explores more concretely how ideas from evolution and EC intersect with those from AI safety. We adopt the framework of Amodei et al. \cite{1}, which identifies five classes of concrete problems that can cause AI accidents: avoiding negative side effects, reward hacking, scalable oversight, safe exploration, and ro-
business to distributional shift. For each of these five problems we introduce the problem, describe how it can arise in EC, how it relates to various research areas in EC, and what potential solutions might look like within EC or from ML at large. Note that we will only sketch some connections to the broader study of AI safety within ML; for more comprehensive surveys, see Amodei et al. [1] or Everitt et al. [2].

3.1 Avoiding Negative Side Effects

The problem of negative side effects is that a fitness function that rewards achieving a desired goal is often under-specified in practice, even if the conditions of achieving that goal are correctly described. That is, there are many ways of accomplishing a goal naively in the real world that humans would nonetheless find unacceptable. For example, borrowing from Amodei et al. [1], a robot might knock over an expensive vase en route to its destination; even if the robot arrives successfully at its destination (its goal), the damage to the vase is an unacceptable negative byproduct of the robot pursuing its goal.

If the fitness function does not penalize for breaking the vase, this could potentially be viewed as a failure of the researcher to express the correct fitness function; however, while we could attempt to anticipate and hard code into the fitness function every negative contingency, there is a human tendency to overlook subtle possibilities, and a tedium in attempting to enumerate in advance every possible harm the robot could cause. Ideally, there would be a way to automatically (or with minimal supervision) augment a goal-directed fitness function to penalize such undesired impacts. The overarching challenge relates to the value alignment problem in AI safety, in that there is much background context (e.g. about what objects in the environment are fragile or important) that a human brings to their understanding of what an acceptable solution is, which is difficult to effectively and exhaustively translate into a fitness function (although some projects do aim to distill such background knowledge [27]).

Interestingly, most EC and ER environments are constructed such that there is little potential for negative side effects, because such considerations are often orthogonal to the research questions under study. That is, simulated environments in ER are very often closed-world and spartan, containing only elements directly relevant to the task at hand. For example, a common variety of ER task involves simulated wheeled robots navigating through an enclosed environment containing only walls and artifacts directly related to the task (e.g. a light switch that can be triggered, or tokens that can be collection); negative side-effects are often avoided by definition: The robot can not damage itself or anything of importance in its environment.

In ER and EC experiments that involve the real world, or interacting with humans, there is more potential for negative side-effects, although experimenters nearly always apriori minimize that possibility by design. For example, when transferring policies evolved in simulation to the real world, the real world environment is often
engineered to mimic the spartan simulated one, and often such transfers are one-off experiments (i.e. the robot will not then be operating in an ongoing way) under intensive supervision. However, despite the minimization by design of negative side effects, the conclusion is that as (or if) EC and ER progresses, we likely will want or need evolved agents to be deployed in complex open-world or human-coinhabited environments; in such situations, the problem of negative side effects can no longer be avoided. Thus, when aiming toward the real world, simulated environments may need to be augmented to include the potential for negative side-effects (and for learning to avoid them), or automated techniques for mitigating side-effects from real-world deployment may need to be developed.

So far, the problem of negative side-effects appears to be an under-studied aspect of how to scale EC, one that may provide exciting future research directions. One possible paradigm for minimizing negative-side effects is to train EC agents through interactive evolutionary computation (IEC; [28]), i.e. to involve humans directly in the breeding process. Due to the problem of user fatigue in IEC [28], i.e. that the task of breeding can become monotonous and exhausting, it is difficult to scale IEC, which necessitates learning surrogate models [29] and/or distributed IEC [30], i.e. systems that involve many humans breeding in potentially uncoordinated ways. Overall, the interaction of IEC with embodied evolution or reinforcement learning in general (as in Woolley and Stanley [31]) could benefit from greater study from a safety perspective. Current research directions in ML that address negative side effects include objective functions that minimize change to the environment [32], or algorithms that satisfice instead of unboundedly optimize [33] (motivated by the idea that side-effects may often result from extreme optimization), both of which could potentially be adapted for EC.

### 3.2 Reward Hacking

The problem of reward hacking, like that of negative side-effects, is caused by an incompletely- or incorrectly-specified fitness function. While negative side-effects are collateral damage incurred while successfully achieving the desired objective, reward hacking is when optimization uncovers unexpected ways to maximize the fitness function that do not achieve the desired objective. For example, if the true objective of a cleaning robot is to clean the office, but its fitness function rewards for each individual mess the robot cleans, the robot may discover that it maximizes fitness by creating new messes that it can subsequently clean [1].

The phenomenon of reward hacking is familiar to most EC practitioners; nearly all of us have encountered situations where an intuitive fitness function is maximized by counter-intuitive behavior. Indeed, that so many illuminating (and funny) anecdotes of reward hacking existed in the EC community was one main inspiration behind the crowd-sourced documentation effort of Lehman et al. [11], which describes many reward-hacking examples. For example, take Karl Sims’ seminal virtual creatures work [34]. In early attempts to evolve locomotion gaits by reward-
ing forward motion, the result was not the desired natural gaits, but morphological evolution towards tall rigid bodies that could exploit their potential energy by falling or somersaulting forward.

Beyond EC, the challenge of constructing incentives for agents (like fitness functions) that cannot be undermined is well known in other fields. For example, in economics, Goodhart’s law [35] states that “when a measure becomes a target, it ceases to be a good measure.” Similar understanding goes by the name of the principal-agent problem in economics and political science [36], and similar challenges exist in designing contracts in law [37]. Further, there are many historical examples of perverse incentives, where an incentive to solve one problem instead exacerbates it; for example, a French colonial program in Hanoi paid citizens for turning in rat tails, in hopes of exterminating rats, but it instead led to farming rats [38]. This consilience of evidence suggests that designing incentives is generally hard, and that humans are often overconfident about their ability to perform such a task well, failing to anticipate subtle loopholes instantiated by intuitive reward structures. In this way, reward hacking in EC and ML is one manifestation of a more general problem.

In practice, reward hacking in EC is often solved through iteration. An intuitive fitness function leads to surprising and undesirable (but often post-hoc understandable) outcomes, and the experimenter then attempts to modify the fitness function to patch the problem, which potentially may lead to a different kind of exploit that must also be patched. Interestingly, because these failed incentives can be viewed as failures of the experimenter, and happen in the loop of scientific experimentation that precedes a polished experimental setup, they are often not reported scientifically [11]; as a result, the prevalence and importance of reward hacking in EC may be under-appreciated and understudied.

While frustrating, when evolution occurs in simulation reward hacking may not cause harm much beyond wasted experimenter effort and time; however, the ability for EC practitioners to quickly and safely explore new tasks, especially in settings such as embodied evolution or reality-gap crossing, is undercut by the expertise and trial-and-error needed to construct reliable fitness functions.

As in negative side-effects, IEC is one avenue for helping to overcome reward hacking, by involving human judgment through dynamically assessing quality rather than by crafting fixed incentive schemes; beyond directed human breeding, there is also possibility for humans to supply other forms of guidance to further constrain or replace traditional fitness functions, like demonstrations of acceptable behavior or heuristic advice [39].

Such EC research directions can be seen as connected to similar potential solutions in traditional ML, such as imitation learning [40], wherein an agent learns how to imitate expert demonstrations of behavior; cooperative inverse reinforcement learning [37], where a reinforcement learning agent cooperates with a human to discover and optimize the human’s preferences; or reward modeling [3], wherein a machine learning model is trained to predict human preferences (similar to surrogate models used in IEC [29]. Exploring if and how such ML methods could apply to EC (e.g. evolutionary imitation learning, or applying deep learning models to learn
models of human preferences to drive evolution) may be a productive area of future research.

3.3 Scalable Oversight

The problem of scalable oversight is that in EC and learning systems in general, there is often expensive feedback that accurately reflects how acceptable a system is behaving, but such feedback is too expensive to be applied as the primary incentive that drives search. For example, an exact fitness measure for a cleaning robot might require expensive manual testing of how much dirt is in a carpet before and after the robot visits a particular room. Other proxy measures may be more cheaply available, such as a human giving a quick glance to a room, or by the robot measuring how much dirt it is picking up; however, such proxies might exacerbate problems such as negative side effects or reward hacking, e.g. a robot maximizing dirt picked up might knock over a plant to gain access to more dirt, or a robot maximizing human approval after a quick glance might hide messes under a rug. The issue is how to efficiently and effectively apply combinations of cheap proxy signals with occasional expensive feedback, to produce a practical (and well-behaved) learning system.

One way the issue of scalable oversight emerges in EC is through the practical construction of real-world fitness functions (e.g. fitness functions for fine-tuning policies learned in simulation, or those used in embodied evolution). In other words, when applying evolution in a real-world situation, what sensors are available on a robot, what a human can easily evaluate, or how the environment can be augmented with automated sensors to evaluate aspects of behavior (e.g. motion capture equipment or ceiling-mounted cameras) will affect what fitness functions are possible to automate, and the overall cost-effectiveness of executing different experiments. That is, scalable oversight, like other AI safety issues, is often eliminated by design from EC domains; experiments in which cheap proxy fitness evaluations are not possible or in which they fail (due to reward hacking or negative side-effects) are unlikely to be pursued or published. However, if progress could be made on enabling more scalable oversight in EC, it might extend the range of what kinds of embodied evolution or real-world fine-tuning could be performed, making it an interesting avenue of research not only for safety reasons, but for expanding the complexity of domains for which real-world EC could be applied.

The area of EC research closest to scalable oversight is that of surrogate-assisted EC [29], wherein expensive-to-calculate fitness functions are approximated with a learned model; of particular interest (for their potential efficiency) are surrogate models that intelligently choose which points in the search space to subject to expensive ground-truth fitness queries. For example, Gaier et al. [41] applies Bayesian optimization to enable data-efficient QD. An interesting direction for future research would be to learn surrogates of expensive real-world fitness function calculations that are derived from available sensor data (i.e. the surrogate would not model how
points in the search space relate to expensive fitness scores, but how streams of sensor data can produce effective proxies of ground-truth fitness); this seems to better exploit available data, and intuitively seems more plausibly related to the process by which humans derive cheap estimates of more complex measures. Another related area of EC study are methods that attempt to model which genomes are likely to successfully cross the reality gap [24]; the reason is that there is an analogy between simulations (and their relation to reality) and proxy fitness measures (and their relation to ground-truth fitness).

From a ML point of view, Amodei et al. [1] propose that a semi-supervised formulation of reinforcement learning may be a productive paradigm for tackling scalable oversight. The idea is that an agent only receives reward information on a small subset of its experience (as opposed to the traditional reinforcement learning setting where reward is observed for each transition); in particular, the agent must learn when to request expensive reward information, and is incentivized to learn cheap proxy measures that correlate with the expensive reward. Because EC uses fitness functions that operate over an individual’s entire evaluation, rather than the per-timestep rewards of traditional reinforcement learning across an evaluation, it may not be easy to translate such a paradigm to EC (although it could be an interesting direction for research). One potential way of framing semi-supervised reinforcement learning for evolutionary RL is to learn a semi-supervised reward predictor (instead of a policy) that could assign fitness to individuals by observing their sensory-motor stream.

3.4 Safe Exploration

The problem of safe exploration is how evolution (or individuals capable of lifetime learning) can explore new solutions without ever (or only very rarely) taking catastrophic actions (that either harm themselves, humans, or other valuable aspects of the environment). Note that safe exploration remains a problem even if objectives are correctly specified. That is, even if a fitness function correctly identifies all unacceptable negative side-effects, and a properly-trained agent would eventually learn to not cause any such effects, during learning an agent learning in the real world might still request catastrophic actions that would be unacceptable if actually executed. For example, the cleaning robot may suffer a fitness penalty for breaking a vase, but it still needs to learn how, during training, to not break the vase. In other words, given a robotic controller that behaves safely, there is no guarantee that an arbitrary mutation of it will also be safe. The danger of exploration is a deep problem, in that the act of exploration seems inherently to involve risk by stepping into the unknown. However, humans can often successfully explore new possibilities and emerge relatively unscathed (sometimes using mental models to predict whether a new strategy would be catastrophic before trying it, somewhat similarly to model-based reinforcement learning [42]), suggesting that reasonable solutions may be possible.
There are two main ways that real-world accidents from safe exploration can emerge in EC. First, take the case of learning a plastic policy (e.g. a policy that is capable of or needs to learn from experience during its lifetime \[43, 44\]). For example, a robot might be trained to explore any environment it is embedded within in search of a particular goal. In effect, such an agent must learn how to explore, and if the deployment plan involves the real world (through embodied evolution, or crossing the reality gap), then there are risks from unsafe exploration, e.g. in a new environment, a learned exploratory strategy might lead the robot to damage itself. Second, there is the case where a learned (non-plastic) policy is either trained in the real world (embodied evolution), or is fine-tuned in the real world after being trained in simulation. In this case, exploring the space of policies (through mutations of existing policies) may result in unsafe policies; for example, in some robotics domains solutions are known to be fragile, i.e. that most mutations result in degenerate (possibly damaging) behavior \[45, 46\]. For concreteness, a robot trained to walk successfully in simulation may lose some performance when transferred across the reality gap, and there is no guarantee that perturbations of the transferred policy (explored in hopes they will improve the policy) will not cause the robot to fall and harm itself.

Overall, it may be impossible to solve the issue of safe exploration without involving some form of human oversight. The reason is that learning what is unsafe seemingly requires either: (1) an accurate model of the world that includes robust identification of catastrophes, (2) labelled data of all possible causes of unsafe scenarios in a domain, or (3) active experience in the domain with feedback from an overseer that prevents unsafe actions from being taken. All three require either extensive domain knowledge, e.g. (1) or (2), or direct human intervention (3). In this way, the problem of safe exploration may be intrinsically tied (like some of the other problems) to that of scalable oversight: Given that potentially expensive human feedback is needed, how can it be gathered and exploited in an efficient way to enable reliable real-world exploration in EC?

Interestingly, like other problems mentioned here, often the issue of safe exploration in EC currently arises outside the formal scientific process: Domains are unrealistically constructed that limit risk (e.g. through spartan closed-world design), and guard-rails to minimize damage to real-world robots and their environment are engineered on a robot-by-robot or domain-by-domain basis by experimenters (and failure modes encountered in such experiments may not be deemed of enough scientific import to be published). Thus, one contribution to studying safe exploration in EC would be to introduce a variant of common ER benchmarks that simulate the idea of safe embodied evolution; for example, a maze navigation task could include deep holes that would endanger a robot, or fragile and valuable aspects of the environment. Another avenue of safe evolutionary exploration relates to the robustness and evolvability of genomes; for example, some EC methods find parts of the search space that are more robust to mutation \[46\], or adapt variation operators to increase robustness or evolvability \[45, 47\], or attempt to enforce small changes to an evolved policy \[48\]. The idea is that with well-tuned variation, parent policies that are safe may be more likely to produce safe children policies, under the assumption that larger policy changes are more likely to be degenerate.
EC could also attempt to solve existing safe exploration benchmarks from the reinforcement learning community, e.g. the safe exploration grid-world of Leike et al. [49] or domains explored by Moldovan and Abbeel [50]. Potential solutions could be imported from the traditional ML community’s study of safe exploration in reinforcement learning [51], such as the approach of Saunders et al. [52], wherein human oversight is used to train a supervised learning model that blocks unsafe actions, or Lipton et al. [53], wherein catastrophic actions are explicitly stored and rehearsed to endow a reinforcement-learning agent with an intrinsic sense of fear. Similar models could be trained to block unsafe actions for ER or in embodied evolution.

### 3.5 Robustness to Distributional Drift

The problem of robustness to distributional shift is how to skillfully deal with the fact that when AI systems are deployed, they will often encounter situations that deviate from the exact ones it was trained upon. Accidents can result in this paradigm if an agent’s policy results in dangerous actions when encountering new situations.

In some EC communities, such as ER, experiments may not always explore how well a learned behavior generalizes to situations other than the exact ones experienced in training; i.e. in the language of statistical ML, the training set doubles as the testing set. As a result, there may be little understanding of how a policy would generalize, and how pathological a robot’s behavior would be if it encountered a novel situation. Note that interestingly, the issue of lacking-generalization is a topic of recent interest in deep reinforcement learning as well [54, 55, 56]. While this paradigm may not be intrinsically problematic, i.e. if the research question does not involve generalization or real-world deployment, graceful degradation of out-of-training-distribution performance becomes critical as policies are deployed in the real-world (especially open-world scenarios where it is well-understood that circumstances will change over time).

Several EC communities study partial solutions to this problem. For example, one subfield of EC studies dynamic fitness landscapes [57, 58], wherein evolution continues as circumstances change, which could continually align the policy to the current distribution of scenarios. Further, such fluid adaptation may favor (or be enabled by mechanisms that encourage) more *evolvable* representations, i.e. representations offering diverse and adaptive variation, another important and related field of EC study [59, 60]. Complementarily, others in EC study meta-learning [43], or evolutionary approaches to learning *how to learn*, which may enable a policy to quickly learn online from its own mistakes.

While these research communities provide important insights for tackling distributional shift, new benchmark tasks may be needed to ground out the risks from real-world distributional shift and to determine which (or which combinations) of these techniques would help ameliorate them in practice. For example, an ER domain could be introduced in which environments are produced through procedu-
ral content generation (PCG; \cite{61}), but where the distribution of PCG parameters changes over evolutionary time; different approaches could be compared by how many catastrophic failures are encountered across evolutionary time.

Solutions could potentially be inspired also by study of the problem within ML at large. For example, the insight in Inverse Reward Design \cite{62} is that the fitness function encountered during training should only be trusted insofar as it reflects situations that occur during training (i.e. the human designer of the fitness function designed it explicitly to solve such training situations); an agent should thus have uncertainty of what the fitness implications are for new situations. It may be possible to export such an insight to an evolutionary context, perhaps by querying a human for guidance or forcing a known safe policy to take over when anomalous circumstances are encountered (e.g. as studied by the fields of novelty/anomaly detection \cite{63,64} or when an uncertainty-aware).

4 Discussion

One interesting question is if EC has unique contributions to make to the general study of AI safety. A potential benefit of evolution relative to traditional ML is its divergent creative potential – evolution seems well-suited to discovering a diversity of high-quality artifacts; subfields of EC that study artificial life \cite{65}, open-endedness \cite{15}, and quality diversity \cite{12} focus on this facet of evolution, which may be of use for helping in particular with the problem of robustness to distributional shift. That is, evolution could be driven to discover a wide range of new training situations to discover latent flaws in learned policies or models, to augment a limited training set that might not cover the diversity of situations that could later be encountered. For example, the work of Nguyen et al. \cite{16} applies a QD algorithm to find, in a single evolutionary run, a set of diverse images that reliably fool a deep neural network vision model; following work has shown that these kinds of adversarial images can provide safety hazards for real-world use cases of such vision models \cite{66,67}. Similar principles from QD or open-endedness can be used to evolve scenarios to test robotic policies. Work in this spirit includes Goldsby and Cheng \cite{68}, wherein novelty search and GP are used to probe latent behavior of a robotic navigation system and an automobile door locking control system. Similarly, the environments evolved by open-ended systems like POET \cite{69} could be adapted as a testing suite for fixed policies.

A separate but related question is to consider what lessons biological evolution has for AI safety. Many problems faced by AI safety have been solved, at least in some abstract sense, by biology. For example, the problem of negative side effects in AI safety is related to the evolution of cooperation and sociality in biology, in that cooperation often entails considering other agents and their goals in addition to one’s own goal (whether through behavioral convention, e.g. as in bees, or deliberative thought as in humans); from this perspective, the negative side effects of a robot pursuing its own limited agenda result from not understanding and/or taking into
account the broader preferences of outside agents (e.g. that a vase is a valuable artifact and should not be broken while cleaning a room). Humans have evolved moral instincts, the ability to empathize with others, and verbal and written language, all of which helps us to understand the gestalt of a task another human might ask us to perform (which can help to avoid reward hacking and negative side effects). Similarly, the robustness of our genetic architecture to random mutations and the natural instincts of curiosity and fear are nature’s hard-won solution to the problem of safe exploration on both a genetic and individual level. In the same way that evolution (and EC) have a privileged position when it comes to human-level AI (because evolution is the way in which human intelligence came into existence), so too evolution and EC may have privileged position when it comes to the AI safety challenges that biology has in some sense solved.

Another question is if methods in EC may manifest different kinds of AI safety concerns than those considered within traditional ML, e.g. due to their lack of formal gradient-following or because they might seek to produce AI as the result of a divergent creative process (as opposed to optimizing an explicit objective function as common in most ML). As a result, it is unclear whether the safety agendas currently popular in ML [3, 5, 6] are applicable to AI produced by paradigms such as evolutionary artificial life or open-ended evolution, which in their grandest aspirations (just as in traditional ML or AI) include producing agents with human-level intelligence [70]. We believe this is a question deserving of more study.

A final discussion topic is to draw together some of the recurring themes from considering each AI safety problem separately, in hopes of highlighting promising research questions and paradigms. One theme is the potential need for modifications of EC benchmarks to include safety considerations or the adoption of existing AI safety benchmarks within EC. Benchmarks, for better or worse, help draw researcher attention, and can make seemingly nebulous problems more concrete. Because existing EC domains and benchmarks minimize safety concerns by design (because researchers most often are pursuing research questions orthogonal to safety), new benchmarks may help to catalyze safety research, especially if they are variants of domains familiar to EC researchers. For example, EC techniques could be applied to the AI safety grid-worlds of Leike et al. [49], or existing ER domains (such as maze navigation or ball-gathering) could be augmented with catastrophic actions (for investigating safe exploration) or in which held-out test environments would test for robustness to distributional shift. Another overarching theme is the potential for some form of IEC to help in the solution to nearly all of the reviewed problems; this is not surprising, because many AI safety problems emerge precisely because human insight is relegated to constructing a fixed setup (i.e. in EC the genetic encoding and the fitness function), and IEC is a framework for allowing human choice to intervene during evolution. Safety considerations may drive more efficient ways to perform IEC (through improved surrogate models), as well as the construction of new forms of IEC. For example, IEC most often helps steer what individuals reproduce, but IEC solutions to problems such as safe exploration may require humans to interact more directly with policies as they execute, i.e. to intervene to prevent unsafe actions. One source of inspiration may be systems such as the neuroevolution-
based game NERO [21], in which a human experimenter can interact in real time to dynamically change the environment, parameters of the fitness function, and even embody a virtual agent to probe learned agent behaviors.

5 Conclusion

AI safety is an important research topic for enabling EC to reach one of its aspirations, which is to maximize its beneficial real-world impact. At first glance, such research might seem uninteresting, because it can evoke sentiments of domain-specific engineering, rather than the pursuit of grand scientific questions; however, AI safety enwraps interesting and philosophically deep unsolved technical challenges, including how to avoid catastrophe while learning about the world, and how to create fitness functions that incentivize agents to do what we intend them to do. As ML and AI grow in import, we can expect funding and interest in AI safety to similarly grow, and the hope of this paper is to advocate for EC researchers to both contribute and take note of advances in this developing field.

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. arXiv preprint arXiv:1606.06565, 2016.
[2] Tom Everitt, Gary Lea, and Marcus Hutter. Agi safety literature review. arXiv preprint arXiv:1805.01109, 2018.
[3] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. arXiv preprint arXiv:1811.07871, 2018.
[4] Eliezer Yudkowsky. Coherent extrapolated volition. Singularity Institute for Artificial Intelligence, 2004.
[5] Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts. arXiv preprint arXiv:1810.08575, 2018.
[6] Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. arXiv preprint arXiv:1805.00899, 2018.
[7] Wolfgang Banzhaf, Peter Nordin, Robert E Keller, and Frank D Francone. Genetic programming: an introduction, volume 1. Morgan Kaufmann San Francisco, 1998.
[8] Stefano Nolfi, Dario Floreano, and Director Dario Floreano. Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines. MIT press, 2000.
[9] M Anthony Lewis, Andrew H Fagg, and Alan Solidum. Genetic programming approach to the construction of a neural network for control of a walking robot.
In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2618–2623. IEEE, 1992.

[10] Chorng-Shyong Ong, Jih-Jeng Huang, and Gwo-Hshiung Tzeng. Building credit scoring models using genetic programming. *Expert Systems with Applications*, 29(1):41–47, 2005.

[11] Joel Lehman, Jeff Clune, Dusan Misevic, Christoph Adami, Lee Altenberg, Julie Beaulieu, Peter J Bentley, Samuel Bernard, Guillaume Beslon, David M Bryson, et al. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *arXiv preprint arXiv:1803.03453*, 2018.

[12] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

[13] Joel Lehman and Kenneth O Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218. ACM, 2011.

[14] Tim Taylor, Mark Bedau, Alastair Channon, David Ackley, Wolfgang Banzhaf, Guillaume Beslon, Emily Dolson, Tom Froese, Simon Hickinbotham, Takashi Ikegami, et al. Open-ended evolution: perspectives from the oee workshop in york. *Artificial life*, 22(3):408–423, 2016.

[15] Russell K Standish. Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3(02):167–175, 2003.

[16] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[17] David Buss. *Evolutionary psychology: The new science of the mind*. Psychology Press, 2015.

[18] Meredith Whittaker, Kate Crawford, Roel Dobbe, Genevieve Fried, Elizabeth Kazuinias, Varoon Mathur, Sarah Mysers West, Rashida Richardson, Jason Schultz, and Oscar Schwartz. *AI now report 2018*. AI Now Institute at New York University, 2018.

[19] Google photos labeled black people ‘gorillas’. [https://www.usatoday.com/story/tech/2015/07/01/google-apologizes-after-photos-identify-black-people-as-gorillas/29567465/](https://www.usatoday.com/story/tech/2015/07/01/google-apologizes-after-photos-identify-black-people-as-gorillas/29567465/).

Accessed: 2019-05-01.

[20] John R Koza. Genetic programming ii: Automatic discovery of reusable subprograms. *Cambridge, MA, USA*, 13(8):32, 1994.

[21] Miguel Rocha, Paulo Cortez, and José Neves. Evolution of neural networks for classification and regression. *Neurocomputing*, 70(16-18):2809–2816, 2007.

[22] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1171–1180. International World Wide Web Conferences Steering Committee, 2017.
[23] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In European Conference on Artificial Life, pages 704–720. Springer, 1995.

[24] Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. IEEE Transactions on Evolutionary Computation, 17(1):122–145, 2013.

[25] Jordan B Pollack, Hod Lipson, Sevan Ficici, Pablo Funes, Greg Hornby, and Richard A Watson. Evolutionary techniques in physical robotics. In International Conference on Evolvable Systems, pages 175–186. Springer, 2000.

[26] Richard A Watson, Sevan G Ficici, and Jordan B Pollack. Embodied evolution: Distributing an evolutionary algorithm in a population of robots. Robotics and Autonomous Systems, 39(1):1–18, 2002.

[27] Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11):33–38, 1995.

[28] Hideyuki Takagi. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. Proceedings of the IEEE, 89(9):1275–1296, 2001.

[29] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation, 1(2):61–70, 2011.

[30] Jimmy Secretan, Nicholas Beato, David B D Ambrosio, Adelein Rodriguez, Adam Campbell, and Kenneth O Stanley. Picbreeder: evolving pictures collaboratively online. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 1759–1768. ACM, 2008.

[31] Brian G Woolley and Kenneth O Stanley. A novel human-computer collaboration: combining novelty search with interactive evolution. In Proceedings of the 2014 annual conference on genetic and evolutionary computation, pages 233–240. ACM, 2014.

[32] Stuart Armstrong and Benjamin Levinstein. Low impact artificial intelligences. arXiv preprint arXiv:1705.10720, 2017.

[33] Jessica Taylor. Quantilizers: A safer alternative to maximizers for limited optimization. In Workshops at the Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[34] Karl Sims. Evolving virtual creatures. In Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pages 15–22. ACM, 1994.

[35] Charles AE Goodhart. Problems of monetary management: the uk experience. In Monetary Theory and Practice, pages 91–121. Springer, 1984.

[36] Stephen A Ross. The economic theory of agency: The principal’s problem. The American economic review, 63(2):134–139, 1973.

[37] Dylan Hadfield-Menell and Gillian K Hadfield. Incomplete contracting and ai alignment. 2018.

[38] Michael G Vann. Of rats, rice, and race: The great hanoi rat massacre, an episode in french colonial history. French Colonial History, 4(1):191–203, 2003.
[39] Igor V Karpov, Vinod K Valsalam, and Risto Miikkulainen. Human-assisted neuroevolution through shaping, advice and examples. In Proceedings of the 13th annual conference on Genetic and evolutionary computation, pages 371–378. ACM, 2011.

[40] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. Robotics and autonomous systems, 57(5):469–483, 2009.

[41] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. Data-efficient design exploration through surrogate-assisted illumination. Evolutionary computation, 26(3):381–410, 2018.

[42] Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In Machine Learning Proceedings 1990, pages 216–224. Elsevier, 1990.

[43] Andrea Soltoggio, John A Bullinaria, ClAUDIO MATTIUSI, Peter Dürr, and Dario Floreano. Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. In Proceedings of the 11th international conference on artificial life (Alife XI), number CONF, pages 569–576. MIT Press, 2008.

[44] Andrea Soltoggio, Kenneth O Stanley, and Sebastian Risi. Born to learn: the inspiration, progress, and future of evolved plastic artificial neural networks. Neural Networks, 2018.

[45] Joel Lehman and Kenneth O Stanley. Improving evolvability through novelty search and self-adaptation. In 2011 IEEE congress of evolutionary computation (CEC), pages 2693–2700. IEEE, 2011.

[46] Joel Lehman, Jay Chen, Jeff Clune, and Kenneth O Stanley. Es is more than just a traditional finite-difference approximator. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 450–457. ACM, 2018.

[47] Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. Natural evolution strategies. In 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pages 3381–3387. IEEE, 2008.

[48] Joel Lehman, Jay Chen, Jeff Clune, and Kenneth O Stanley. Safe mutations for deep and recurrent neural networks through output gradients. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 117–124. ACM, 2018.

[49] 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.

[50] Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. arXiv preprint arXiv:1205.4810, 2012.

[51] Javier Garcia and Fernando Fernández. A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16(1):1437–1480, 2015.

[52] William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In Proceedings of the 17th International Conference on Autonomous
Agents and MultiAgent Systems, pages 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[53] Zachary C Lipton, Kamyar Azizzadenesheli, Abhishek Kumar, Lihong Li, Jianfeng Gao, and Li Deng. Combating reinforcement learning’s sisyphean curse with intrinsic fear. arXiv preprint arXiv:1611.01211, 2016.

[54] Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. A study on overfitting in deep reinforcement learning. arXiv preprint arXiv:1804.06893, 2018.

[55] Niels Justesen, Ruben Rodriguez Torrado, Philip Bontrager, Ahmed Khalifa, Julian Togelius, and Sebastian Risi. Procedural level generation improves generality of deep reinforcement learning. arXiv preprint arXiv:1806.10729, 2018.

[56] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. arXiv preprint arXiv:1812.02341, 2018.

[57] Jürgen Branke and Hartmut Schmeck. Designing evolutionary algorithms for dynamic optimization problems. In Advances in evolutionary computing, pages 239–262. Springer, 2003.

[58] Hendrik Richter. Detecting change in dynamic fitness landscapes. In 2009 IEEE Congress on Evolutionary Computation, pages 1613–1620. IEEE, 2009.

[59] Günter P Wagner and Lee Altenberg. Perspective: complex adaptations and the evolution of evolvability. Evolution, 50(3):967–976, 1996.

[60] Nadav Kashtan, Elad Noor, and Uri Alon. Varying environments can speed up evolution. Proceedings of the National Academy of Sciences, 104(34):13711–13716, 2007.

[61] Noor Shaker, Julian Togelius, and Mark J Nelson. Procedural content generation in games. Springer, 2016.

[62] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse reward design. In Advances in neural information processing systems, pages 6765–6774, 2017.

[63] Markos Markou and Sameer Singh. Novelty detection: a review—part 1: statistical approaches. Signal processing, 83(12):2481–2497, 2003.

[64] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3):15, 2009.

[65] Christopher G Langton. Artificial life: An overview. Mit Press, 1997.

[66] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models. arXiv preprint arXiv:1707.08945, 2017.

[67] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.

[68] Heather J Goldsby and Betty HC Cheng. Automatically discovering properties that specify the latent behavior of uml models. In International Conference on Model Driven Engineering Languages and Systems, pages 316–330. Springer, 2010.
[69] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019.

[70] Kenneth O Stanley, Joel Lehman, and Lisa Soros. Open-endedness: The last grand challenge you’ve never heard of. *While open-endedness could be a force for discovering intelligence, it could also be a component of AI itself*, 2017.

[71] Kenneth O Stanley, Bobby D Bryant, and Risto Miikkulainen. Real-time neuroevolution in the nero video game. *IEEE transactions on evolutionary computation*, 9(6):653–668, 2005.