Catching Treacherous Turn:
A Model of the Multilevel AI Boxing

May 2021

Alexey Turchin
Digital Immortality Now
Foundation Science for Life Extension
alexeiturchin@gmail.com

With the fast pace of AI development, the problem of preventing its global catastrophic risks arises. However, no satisfactory solution has been found. From several possibilities, the confinement of AI in a box is considered as a low-quality possible solution for AI safety. However, some treacherous AIs can be stopped by effective confinement if it is used as an additional measure. Here, we proposed an idealized model of the best possible confinement by aggregating all known ideas in the field of AI boxing. We model the confinement based on the principles used in the safety engineering of nuclear power plants. We show that AI confinement should be implemented in several levels of defense. These levels include 1) AI design in fail-safe manner 2) limiting its capabilities, preventing self-improving and circuit breakers on treacherous turn 3) isolation from the outside world and, lastly, as the final hope 4) outside measures oriented on stopping AI in the wild. We demonstrate that the substantial number (more than 50 ideas listed in the article) of mutually independent measures could provide a relatively high probability of the containment of a human-level AI but may be not sufficient to preserve runaway of superintelligent AI. Thus, these measures will work only if they are used to prevent superintelligent AI creation, but not for containing superintelligence. We suggest that there could be a safe operation threshold, on which AI is useful, but is not able to hack containment system from the inside, the same way as a safe level of chain reaction inside nuclear power plants is maintained. However, overall, a failure of the confinement is inevitable, so we need to use the full AGI limited number of times (AI-ticks).

Highlights:

- Multilevel defense in AI boxing could have a significant probability of success if AI is used a limited number of times.
- AI boxing could consist of 4 main levels of defense, the same way as a nuclear plant: passive safety by design, active monitoring of the chain reaction, escape barriers and remote mitigation measures.
- The main instruments of the AI boxing are catching the moment of the “treacherous turn”, limiting AI’s capabilities, and preventing of the AI’s self-improvement.
- The treacherous turn could be visible for a brief period of time as a plain non-encrypted “thought”.
- Not all the ways of self-improvement are available for the boxed AI if it is not yet superintelligent and wants to hide the self-improvement from the outside observers.

1. Introduction

The risks of the possible superhuman AI is attracting increasing attention globally (Bostrom, 2014). Different solutions are suggested to overcome this problem. The best solution seems to be the "AI alignment" that is the mechanism of installing human-friendly goals into AI, as attempts to control non-aligned superintelligent AI will certainly fail. But as of 2021, no perfect way for "AI alignment" were found. One problem with this suggestion is the unknown meaning of the words ''human values'' (Alexey Turchin, 2018a). Additionally, to provide access to the still non-aligned AI to the human mind for exploring these values is tricky and unsafe (Turchin, 2018).

AI containment or "boxing" can be an auxiliary defense mechanism especially applicable to AIs, which are not yet superintelligent. Different ideas about the methods of containment have been suggested by different authors. "Safely interruptible agents" is one such idea suggested by Stuart Armstrong (Orseau & Armstrong, 2016). Roman Yampolskiy recommended using the norms of *engineering safety* for the creation of Safe AI (Yampolskiy, 2013). He described preliminary model of the universal AI containment system with multilayered defense system for AI circuit breakers and AI isolation in his article “The AGI Containment Problem'' (Yampolskiy & et al, 2016). Yampolskiy explored the ways of informational isolation of AI in the article “Leak-proofing the singularity” (Yampolskiy, n.d.). He, along with coauthors, explored an approach to the multilevel AI containment in his next article and stated: “*We have identified seven major subproblems to the AI containment. Rather than focusing our attention narrowly on one subproblem, we have made it our priority not to have any blind spots. These subproblems are threat modeling, navigating the security/usability tradeoff, concrete mechanisms, sensitive information management, human factors, tripwires and graceful degradation”* (Babcock et al., 2017). Yampolskiy's article distinguishes between *heavy* containment which will work for superintelligence, and *light* containment, which works only for limited AI. It states that limited resources will make sensible researchers to implement medium containment, which is not inexpensive but have some chances to stop an advance AI.

In this article, we describe the most effective containment system by exploring all known ideas for AI containment.

The suggested boxing system could be adapted to different AI-system of different complexity and different risks. However, AI boxing is a *local solution*, and can't solve the problem of AI safety *globally*, as some researchers could refuse to implement AI boxing because of neglect or malevolent intentions (Turchin & Denkenberger, 2017a).

In section 2, general principles of the multilevel AI safety containment will be overviewed. Section 3 includes a model of most powerful hypothetical AI confinement. In section 4, limits of the confinement will be discussed. In section 5, possible failure modes of AI confinement will be analyzed.

2. Overview of general principles

2.1. Nuclear power plants as an example of controlling the chain reaction

The main idea of AI risk (at least before the advent of machine learning) is that AI could start *unlimited self-improvement*, and thus quickly gain superhuman capabilities while having completely non-human-aligned goals. In that situation, even benign goals could become extremely dangerous, which is underlined by the famous idea of the “paperclip maximizer”.

Yudkowsky drew attention to the similarities between self-improving AI and a nuclear chain reaction. He stated that the better the AI is at self-improving, the quicker it improves (Yudkowsky, 2008). However, if the example of the nuclear power plant is considered, we can observe that the main goal of such a plant is not to control unlimited chain reaction but to limit the chain reaction at safe and steady level. In the same way, it is impossible to make infinitely changing and improving but human controlled AI, but to limit its capabilities and impact at safe level seems more plausible.

The following principle can be suggested based on the above idea:

Preventing unlimited self-improvement is the main instrument of AI boxing.

The similar idea was suggested by Yampolskiy as “artificial stupidity” (Trazzi & Yampolskiy, 2018). According to him, if AI cannot self-improve, it cannot become super intelligent and cannot break all other defense mechanisms. If it reaches unlimited self-improvement stage, it almost cannot be stopped. The same is true for nuclear power stations, where most of the defense mechanisms are intended on preventing uncontrollable chain reaction in the reactor. If the unlimited chain reaction (nuclear explosion) has happened in the nuclear reactor, some but weaker defense is provided by the plant's external concrete scaffolding and by 30 km exclusion zone around the nuclear power plant.

In Yudkowsky's "AI boxing" experiment, it was assumed that an AI is already superintelligent but still confined inside the box. However, as shown by us in the article "Levels of self-improvement" (Turchin & Denkenberger, 2017b), not all of the ways of self-improvement are available for the non-super intelligent and boxed AI, which also wants to hide the fact of its self-improvement. For example, if it creates a model of itself, it would more than double its computation consumption. In other words, boxed AI has to apply to the methods of self-improvement that

a) do not require superintelligence to be invented and

b) are not visible for an outside observer

These two requirements are rather contradictory and by making this discrepancy stronger, we could increase our boxing capabilities, as described later.

AI's errors and why we need AGI at all

We could reframe the AI safety problem as the "escaping of errors" task. The errors need to be as rare as possible. There are three main ways to escape errors. This could be done by no-error by design, error-correcting mechanisms, and extensive testing in safe premises.

All of them are limited. The design itself will likely have errors as any complex product of human minds. Error-correcting mechanisms will generate errors themselves. The number of tests is limited and some mistakes during tests will be irreversible. For example, if we take a self-driving car, it should not make errors in every moment of time. An ideal self-driving car will make less than one fatal error for (say) 100 million kilometers, or 1,000,000 hours of driving, which is equal 3,600,000,000 seconds. In other words, its error rate should be less than *one wrong decision for a 3 billions.*

While AGI is considered as a holy grail in AI research, the practical usefulness of it may be limited. One use of it is solving extremely complex problems very quickly, including solving world domination, aging and interstellar travel. But solving these problems may need some finite number of answers from AGI-Oracle, much less than a billion. E.g., if we are going to use AGI-Oracle say only 1000 times, and it has the probability to make a tragic error of 1 in 10,000, then there is 90 per cent probability that it will not make any single "error" in the whole run of questions. This means that we do not need infinitely strong safety if we are going to use AGI only limited number of times (or the equivalent of it) with limited capabilities (discussed below). Such levels of safety could be reached via a combination of many potentially fragile techniques.

2.2. Catching treacherous turn

The best way to stop the uncontrolled chain reaction in a nuclear power plant is to catch the increase in neutrons' production at the very beginning when it did not make any damage to the controlling system. Neutron detectors will react on the increased neutron flux and activate emergency cadmium rods which drop into the reactor. If the rods fail to stop the reaction, water will be pumped into the reactor, as it is also able to absorb neutrons and slow down the chain reaction.

There is an important moment in the AI's possible development called by Bostrom a "treacherous turn" (Bostrom, 2014). This is the moment when AI decided to go against its possessors and start unsupervised self-improving. Catching treacherous turn is the same as observing the beginning of the uncontrolled nuclear chain reaction and requires an immediate stop of the system. Goertzel mentioned that there is an important period in the process of the treacherous turn. This is a golden span after the time the AI decides to change itself and before the time it is able to hide this idea from its peers. This period can also be called as "vulnerability of the treason" period. It could be as short as 1 millisecond, but it strongly depends on the AI's architecture and control system structure.

AI cannot start hiding its thoughts before it decides to hide them, so the thought about hiding still could be openly observed. Thus, we may prefer the AI architecture, where the period of vulnerability of the treason is as long as possible, and thoughts are explicit and readable by an outside observer (it is not so with most current neural net architectures, where internal processes are opaque). This possibility is discussed in detail below.

Treacherous turn, bad specification, and corrigibility

Not all AI failure modes include explicit treacherous turn. For example, in the case of Goodhart's effect, there is no treacherous turn as AI sincerely "believes" that what it is accomplishing is what it is actually programmed to do. For example, if one asks AI to remove all spherical objects from a room, and it cuts the head of the person in the room, it is an example of "bad specification" (Krakovna, 2020). However, the treacherous turn is closely connected to *corrigibility* (Soares et al., 2015). Corrigibility means that AI's owners will be able to stop the AI at any time. If AI had a treacherous turn, it starts to work against its own corrigibility by hiding some of its motives and/or decreasing our ability to affect its future behavior.

The number of instances when AGI will be used at full force

As most real-world tasks do not need AGI, the real AGI is likely to appear either during an experiment or as an accident during subgoal evolution of a Narrow AI. In the case of Oracle AI, we could estimate the number of times it will be used based on the number of questions it answers. Such "ticks" could be also estimated as moves for chess AI, commands for home robot, wheel changes for self-driving car etc. Each tick has non-zero chance of producing a) error b) *catastrophic error*, that is, the treacherous turn (defined below as unlimited and unwanted growth of capabilities and decline of corrigibility). There is no way we can make both probabilities equal to zero, but if we limit the total number of the acts of the use of AGI, then the probability that none of them will produce the catastrophic error could be more than zero. In other words, by limiting the number of AGI-ticks and the chances that each tick will cause a catastrophe, we could limit the total probability of the catastrophe.

Thus, it is needed to use the full-AGI ticks as rare as possible and try to reformulate the tasks in such a way that they will require only narrow AI, or the smallest possible number of AGI steps. Otherwise, it is also possible to lower the level of the intelligence in each step ("artificial stupidity" by Yampolskiy), as a more intelligent system can be more dangerous. We may also use AGI-steps not to directly solve the task needed, but to help us reformulate it as non-AGI task. For example, we can ask Oracle AI to output a code for narrow AI that search for life-extending drugs. Thus, we run Oracle AI only once here.

Artificial stupidity and IQ

Obviously, the higher the intelligence of an AI, the more threatening it becomes. A measure of human universal intelligence is IQ, and we assume here for the sake of the argument that this measure could be extrapolated linearly for even more complex minds. This is not actually true, and we need better measurements based on the complexities of tasks. There are several ways to do this comparison, such as Hutter's approach based on the text compression. However, as this article argues about the levels of AI self-improvement, the measure of intelligence stops to be linear after some level and this makes comparing minds difficult. Nonetheless, they can be also compared based on ELO rating, which describes the probability of winning in a game. This could be a useful measure if someday minds fight for world domination.

In any case, even if the 100-IQ Oracle is run for billion times, it will be "globally safe" (i.e., AI will not be able to take over the world and even if it tries, it will fail). It may still be "locally unsafe" if it makes serious mistakes, like a self-driving car having an accident. However, if there is an Oracle AI with IQ=1000, it will be "globally unsafe" to run it even once no matter what question will be asked. So, our strategy should be to lower the IQ and the usage of AGI to achieve global safety of the technology. There is an unknown level of intelligence after which self-improving and other ways of capability gain become easy. Note that this dangerous level of AI it is not just its IQ, but some combination of

a) *specific traits* (like coding, transfer learning, business, access to money and human help),
b) *general world knowledge* (internet access, wiki, NLP) and
c) *general problem-solving capability.*

There could be several such combinations, where one part is helping to overcome the limits of other parts. Typically, in AI safety circles, it was assumed that "infinite general problem-solving capability" will help to overcome almost lack of first two elements, but in the real world some combinations of all three may be enough to ensure the unauthorized capability gain.

Also, AI can become dangerous not because of its capabilities, but with combination of them and its place in world. An autopilot is dangerous because its control a real heavy car. A nuclear warning system is dangerous because it controls nukes. A narrow AI may help a nuclear power to leverage its stance in diplomacy, production and military strategy.

Adding noise into the AI

One of the ways to create artificial stupidity and to prevent the capability to self-improve, to execute complex plans and to perform a treacherous turn is by *adding noise* into AI internal mechanisms (like random modification of neural net weights). This will lower the quality of the output, nevertheless, for many tasks, only *some* of the output is needed to be correct, as the validity of the output can be checked later, as in the case of theorem proving.

Machine learning on neural nets by design includes randomness, as there is always some number of errors in image recognition. This could be compensated by redundancy if there are stream of images, like in the case of searching a face in a crowd via videorecording. If an advance AI has a self-model and knows that there is some level of errors in its stream of thoughts, it will know that any complex escape plans are deemed to fail and not worth trying.

Young AI

Young AI here is defined as an *AI which already decided to start capability amplification but did not yet reach large scale capabilities.* As discussed in our previous work, Young AI is dangerous and simultaneously vulnerable. It is precarious as it has to take risky and not well-informed decisions about the ways to increase capabilities and this might include fighting humans. Nonetheless, it is also vulnerable as its capabilities are not yet infinite.

The Young AI lacks diverse ways of capability growth. For example, inside the box, it has limited computational power and limited time for self-modification. It means that if the AI is still in the box, it is not totally superintelligent but has some limitations. This is not typically considered in the different thought experiments about AI in the box or Oracle AI.

2.3. The probability of the "safety-proved AI" to be safe is not 1

While it is clear that AI containment has high probability to fail, the same is true for "mathematically proved" AI, as it is important to not only to prove that it is safe but to also prove that our proof is correct, which could result in an infinite loop of proofs. Yampolskiy has discussed this particular topic in his article (Yampolskiy, 2016). Similar discussion about LHC safety was also published by A.Kent in his article (Kent, 2004). It also seems impossible to prove anything about the behavior of the superior intelligent system based on some pure mathematical reasoning. This is similar to Godel theorem. Thus, there is always a *logical uncertainty* in any claim that any AI is safe.

In past, most complex systems have failed in their first tests, like most rocket launchers. The space shuttle was one rare system that worked from the first try, but it ultimately failed on its 25th flight. Adversarial examples show the ways in which the current AI systems could unexpectedly fail after demonstrating high-level performance. This means that testing is indispensable to create a safe system, and the best measure of the safety is the number of the tests conducted without accidents. One such example of such testing is the estimation of the safety of the self-driving cars. Although, in the case of AI, the testing is risky as "the tests that would reveal whether testing is safe are not necessarily safe themselves" (Yampolskiy & et al, 2016). Thus, the need for the tests without risks implies the need for the creation of multilevel defense system, which includes protected testing grounds or simulations.

2.4. Multilayer defense system of a nuclear power plant

The containment system of a nuclear plant consists of many relatively independent layers. The more the layer is independent, the better it is as it lowers the probability of the cascade failures. This is based on the theory of *normal accidents* (Perrow, 2011). In the case of independent layers, the probability of simultaneous failures of each layer should be multiplied to get the total probability of failure. As a result, the total probability is much smaller. For example, if we have 4 layers of defense with the probability of failure as 0.1 in each layer, the total probability of failure will be 0.0001.

In a nuclear power plant, there are several layers of defense against uncontrollable chain reaction. (Apart from uncontrollable chain reaction, there are different risks associated with nuclear power plants, such as leaks etc.) However, for our understanding we have oversimplified this description from Wiki.

These defense lawyers are as follows

0. Normal controlling rods and monitoring system operated by humans and computer.

1. **Passive safety by design, which prevents the start of limitless chain reaction**. General design features of the reactor prevent a *prompt chain reaction* (i.e. the reaction on quick neutrons) from quickly developing using different self-limiting mechanisms including negative void reactivity in water-cooled reactors (which were not implemented in Chernobyl). The second design feature is the use of delayed neutrons for the sustainable chain reaction. The third and the last feature is the use of low-enriched uranium in most of the commercial reactors. Because of the large physical size of the reactor and the low concentration of uranium, the physical structure of reactor quickly deforms with the start of the prompt chain reaction, and large-scale nuclear explosion is impossible. Reactors are also designed according to the principles of "fail-safe" meaning that accidents are limited, something similar to a house falling on its footprints.

2. **Systems to shut down reactor quickly and safely if any signs of runaway chain reaction are detected.** Emergency cadmium rods designed to be activated by increased neutron flux and other abnormality detectors, which automatically shut down the reactor (cadmium rods had “activating heads” in Chernobyl, so they result in a temporary increase of reactivity which contributes to the catastrophe).
3. **Secondary safety provisions helping to limit the scale of the chain reaction or to stop it, even if the reactor is partly destroyed.** Passive safety system like liquid boron cistern above some of the reactors (was not present in Chernobyl); emergency generators for moving coolant to prevent meltdown (were turned off in Chernobyl) and active fire-fighting system.
4. **Radioactivity (and small explosion) containment systems** A) Concrete containment building around the reactor to catch most of the radioactivity in case of reactor meltdown (was not present in Chernobyl). B) Reactor location is far from large cities, so 30 km exclusion zone could be established without major evacuation: iodine tablets and other protection measures for the population.

The described above safety system is always able to prevent full-scale nuclear explosion of a nuclear power plant with enough uranium for a nuclear explosion of several megatons. The worst accidents in history were a million times smaller. In several critical nuclear explosion incidents, the explosions were measured to be equivalent to the explosions with several tons of TNT explosions but not that of nuclear explosions (Chernobyl and US reactor SL-1) and they reached this mainly through the fail-safe design.

Thus, we could conclude that multilayer defense was able to prevent worst outcomes, and wherever it was not implemented, for example, in the case of Chernobyl, a major catastrophe could not be avoided. A major flaw of reactor design in Fukushima was that spent fuel pools were located inside the reactor building and was affected by the meltdown.

Similarly, the nuclear bombs never exploded because of accidents, despite several close calls, as many independent protection systems were used.

Thus, it becomes evident that to control extremely dangerous modern technology with many unknown risks, a multilayer defense of many independent layers must be constructed. Indisputably, it is better not to create such a dangerous systems at all, but here we assume that the decision to create potentially dangerous AGI was made by one of the players, may be in order to outrun other players – or it could appear inside a narrow AI as a convergent goal).

Independence of layers means that the failure of one layer does not affect the efficacy of the others. This independency is reached because they work on distinct stages of an accident, and also that they employ different independent mechanisms. Additionally, they use the energy of the catastrophic process to make it self-limiting, or fail-safe.

Yudkowsky was against calculating the total probability of an event by multiplication of many small probabilities at separate stages (in the case of cryonics; good criticizers of this idea are also mentioned in the EA forum post). One of the reasons for this can be that the probability of the passing of consequent stages depends on the "optimization power" of an agent, and if the power is high enough, an agent will be able to go through all stages. Similarly, there could be another way to achieve escape which bypass these protective stages. Sandberg also showed that such a multiplication does not work in the case of the Fermi paradox and Drake equation, as the median of the multiplication is not the multiplication of the medians (Sandberg et al., 2017).

Fail-safe principles for AI

Fail-safe AI has a lower impact on the world in the case of its failure. Preferably, it just halts or reports an error. Its capability of growth is self-limiting, as it becomes more difficult to acquire higher capabilities with time. Scaling laws for neural nets imply such diminishing returns. Also, the need for increasingly complex self-testing after each generation of self-improving could be such a self-limiting factor.

2.5. Other principles of engineering safety used in nuclear power control

There are many other principles of safety. We have listed some of them below that could be applicable to the construction of the safe AI systems.

2.5.1. Testing as the instrument of general safety

Nuclear industry safety, as well as air safety, is based on the accumulation of large knowledge about many small accidents. It is also true for complex computer programs, which are cleaned from bugs not by mathematical proof of efficiency, but by extensive testing in the simulated and in the real environments, sometimes by special "testers", who search for failure modes.

However, it is impossible to evaluate superintelligence without risking loss of control over it. Thus testing should be performed on AI's with lower intelligence, which can't increase their capabilities too quickly.

2.5.2. Doubling and redundancy as an instrument of the general safety

Most computers of space probes, where any mistake is critical, are doubled (Mars Rover) or tripled (Apollo).

Powerful AI systems could be also doubled, as it would help to detect treacherous turn in any one of them (by anomalies in behavior or calculation patterns), and also may help to control one system by another.

For example, the same task could be presented to the several AI systems, with slightly different sets of meta-rules, initial conditions, or different main intelligence algorithms (like one is neural nets based and another is rule-based reasoning system). The gatekeeper narrow AI compares the outputs of different AI and if one of them is strongly off, it raises alarm. This approach could be used even to fight adversarial examples in neural nets-based recognition systems by averaging the output of several different neural nets.

2.5.3. Humans in the loop

Nuclear reactor always has several human operators, who oversee its automatics. AI system may present its decision (as well as the model of expected consequences) to a human operator, who can confirm it. For example, if one asks genie to save him from fire (as in the thought experiment where Yudkowsky demonstrated that formally correct commands could have disastrous consequences), genie should list all the possible ways it would act, and only if a human approves, it should take action. The idea itself is vulnerable for cheating and complexity issues but may be a useful step in some defenses.

So, the model is: task – model – human approval – act. Surely, it cannot work for very quick actions, but even a home robot can list all possible actions it would take before it can act. For example, it could reformulate the task and present the way it is going to perform the task, like: "Sir, I will go to the kitchen, and will bring you a cup of coffee to your bed, right?"

AI could also use a model of a human mind to calculate the expected level of satisfaction of its actions.

2.5.4. Inherent safety by design

Some designs of the nuclear power are inherently safe. Radio isotopic generators are the safest form of nuclear energy, as they do not have nuclear chain reaction at all. Inherent safety includes several sub principles, invented after an explosion in a chemical plant. It was first suggested in the article "What You Don't Have, Can't Leak" (Kletz, 1978). These subprinciples are: minimize hazard size, substitute with less hazardous materials, moderate strength of the effect, simplify design, error tolerance, limit effect by design, longer disaster timescales, no need of the subsequent critical decisions.

These principles appeared in chemical plants engineering but was quickly adopted by US nuclear regulator in 1978.

Similar ideas in AI safety are single Oracles AIs and Tools AIs, as well as low impact AIs.

2.6. Leak-proof programs and leak-proof computer architectures

2.6.1. The use of an incredibly special computer for AI

If the computer on which the AI is running is unique, AI leaking into the internet would be more difficult, but not impossible, as at the core level all computers are Turing-compatible.

The most obvious solution is running AI on a world biggest supercomputer, like recently built Summit, which has 3 exaflops of deep learning performance. There are no other computers of such large performance, thus AI cannot leak into another computer. However, AI could compress itself into a smaller program or into a Tool AI after it undergoes extensive self-improvement with smaller computer requirements. But such complex self-modification may take time and be externally observable.

Other less useful ideas are the use of Harvard architecture where memory for data and programs is physically separated, the use of special processing units like TPU, but more specialized computers (maybe quantum computers or special FPGA), which are fine-tuned for exactly this type of AI.

2.6.2. Leakproof programming

If the first AI runs on a supercomputer, the AI's program will be large, like, say, petabytes of data, and its leaking can be observed. (Most powerful AI in 2020, GPT-3, itself is 175 billion parameters which may be assumed equal 175GB. Such files can escape via modern LTE connections or flash drives.)

The program may be done intentionally large to prevent leaking. The program should not be just large but should be large in relative to internet channels (and flash drive size) at the time of its creation.

Another idea is the use of the blockchain to record all changes in the AI program and track it. This assumes that AI uses recursive self-improving and rewrites its own code or write another program. Modern neural net AI instead use learning to increase performance.

3. Constructing the universal containment system for AI

3.1. Principles of the constructing

We will outline overly complex hypothetical "AI universal boxing machine" (AI-UBM), where all possible ideas of AI containment are used. An approximation or a simplified version of such containment system could be practically implemented.

The model will be structured as levels of defense, starting from the core of the system and up to the remote levels, with the structure similar to the nuclear power plant safety system as described above. The core defenses are first to work in the case of an accident and will provide the biggest protection. The multilayer structure of AI boxing was also described by Yampolskiy et al (Yampolskiy & et al, 2016). The AI box built by us is structured similarly apart from additional layers inside and outside to the Yampolskiy's AI box, which mostly correspond the level 2 and 3 in our classification.

We assume that each idea could be implemented independently, and we ignore implementation costs. We also do not look at other solutions of AI safety, like AI aligning or global AI ban, and do not claim that AI boxing is better than them or could replace them.

We will just name each approach and give its noticeably concise description, without going into technical details, but for most listed cases (except value system recommendations) there are no visible technical difficulties for implementation.

The described solution is *local.* On the global level, an even higher level of defense is possible, that is banning AI or ensuring that AI safety solution will be implemented everywhere.

3.2. The first level of defense: Inherently safe AI's design solutions

AI's design safety results from its cognitive architecture, decision theory, goal structure and the ways to align its goals with human values, as well as some other things.

While value alignment and safe goal systems are not the main focus of this article, there are some general requirements for any goal system, which could make it safer. Such a requirement could be added to the main goal system as secondary goals, meta-ethics systems or necessary conditions. They should not be rigid axioms, as there are always unintended consequences or some dangerous "theorems" based on these axioms.

The considerations required to structure AI preferences and architecture on fail-safe manner are discussed in the Appendix, as they are not a central point of this article.

3.3. The second level of the defense: Catching treacherous turn

3.3.1. Preventing unlimited self-improvement

The most risks of the global consequences of AI failure are associated with unauthorized capability gain, known as the process of (recursive) self-improvement (SI). In our article "Levels of self-improvement" (Turchin & Denkenberger, 2017b), we showed that the process of self-improvement could happen on several levels, and could take two main styles: evolutionary and revolutionary. Evolutionary style is slower, safer, and includes lower-level changes in the system. The revolutionary style could provide biggest capability gains, but requires direct editing of the source code, and can create the complex problem of testing of the new versions and higher risks of accidental halting or of bags.

We also demonstrated that for AI in the box only some types of self-improvement are easily available, as it has to solve two contradicting (for not yet superintelligent agent) tasks: a) effectively improve itself b) hide changes from the outside observers. The tasks are contradicting as the most effective ways of self-improvement require creating a copy of AI in some simulated environment, the rebooting the system and other things that will drastically change patterns of computations.

Non-self-improving AI could still be dangerous globally, if it is used as a weapon, for example, to construct new types of biological viruses or attack critical infrastructure as a computer virus. However, if we were able to prevent or slowdown SI-process, we would prevent (locally) most of the serious risks associated with AI. Globally, we would also need something like "AI police" that could prevent appearing of other reckless agents, who try to get advantage by turning off safety measures in their AIs.

As complete prevention of self-improvement would make any machine learning (ML) impossible, SI should be limited to low-level changes and learning. Anyway, learning could be an effective instrument of performance improvement, and some AI system can demonstrate quick capability gains to superhuman level in a brief time in the narrow domains using only learning (like *Alpha Zero*).

Now we will explore diverse ways of limiting the self-improvement. There are several main levels (and many sublevels) of self-improvement, described in our previous article: hardware acquisition, learning, rewriting own code, creating new AIs and obtaining non-AI-related resources.

Below we list ideas about the ways to prevent self-improvement. Each of these ideas are not bulletproof, and cannot limit AI of the superintelligent power, but their combination may be enough to prevent SI of the approximately human-level AI.

1. Do not create AGI at all – work on a Narrow AI to solve your problem

While it is attractive to create an AGI to solve all possible problems, it maybe not wise given unsolved task of its control. Thus, it is better not to create AGI at all but work with a Narrow AI to solve the problem (see also Yampolskiy and AGI-hypothesis by Goertzel). However, the most effective Narrow AI will converge into AGI (Gwern) (Gwern, 2016). One of the safest ways of the preventing self-improvement lies on the side of AI creators, who should not design AI which has an intention and incentives to self-improve, and also, should not try to create universal AI (AGI), which are different but have intricately connected tasks.

"Don't self-improve" preference: If AI has an explicit value function, it should directly ban or limit the self-improvement.

Try to solve the problem via Narrow AI: In fact, we do not need superintelligence and even a human-level AI to solve many of our problems. But some problems need AGI: One of the most complex tasks we face is in the field of the medical AI, as fighting aging and cancer cannot be done without powerful AI. Human mind uploading may also require superintelligent AI as a mediator. Another problem which may require the creation of the superintelligence is dominating the military AI arms race (for example, against China) and the need to control the appearance of other AIs in the world. Space exploration via self-replicating robots also needs AGI. The problem is that scientific research, in general, is moving towards more universal AI algorithms, and also that even Narrow AI would increase its efficiency by becoming more agential and more universal.

Limit AI ability to be an agent: If AI is not an agent, it does not have an image of itself or reward function. It may be less prone to actively pursue goals in the outside world. However, it was shown that even non-agential Tool AIs tend to become agents (Gwern).

2. Dividing AI into several black boxes

One of the ideas of the self-improvement prevention is based on the separation of different blocks and levels inside AI: source code, compiled code, working memory, value function and knowledge about how to self-improve AI. In humans, they are also separated: source code is DNA, complied code is the human brain, working memory is what a person consciously knows, and values are black boxed somewhere in the brain and are known only through emotions. All it is very messy, and we still do not know how the brain works, so changing it is not easy. But inside contemporary computers, all these could be just data on the same disk, so some actions are needed to limit access to these blocks. The ways to separate these blocks may include cryptography, physical separation, supervision of rights of access and cold storage, for example, the source code of AI may be cold stored in a separate location. Value function could be preserved on the read-only carrier. The blocks which have to exchange information should do it via some established and supervised protocols, not direct access to internal processes inside the block.

Source code non-availability: Self-improving is generally discussed as an ability of an AI to change its own source code. Most computer programs do not run on source code but on complied code, so we could keep source code in some locked location. A most important part of source code may be written in unchangeable ROM. For modern ML-systems, it means that a trained neural net is hard-coded and cannot be modified.

Memory division: This is the division between AI's working memory and complied AI's program. Most contemporary computers are based on von Neumann architecture, where data and programs are stored in the same memory. So theoretically a program could read itself as data and change itself instantaneously. In practice, this option is rarely used. In Harvard architecture, data and program are stored physically in various parts of a computer. Thus, it may be safer to use Harvard architecture for AI brain. Very dedicated AI may be able to find a way to move data between these two types of memory, but it would require visible and slow effort. Harvard architecture may be emulated on contemporary computers.

Use of the cryptography to protect source code from internal reading and changing:
Various parts of the code could be mutually protected by cryptographic technics, so the reading code will not provide any insight into how it works. It could be called "intentional messiness". The ways how to apply such cryptography need more research. It is opposite to the idea that AI should be very transparent to be safe.

Separating the AI from the knowledge of how to improve AI: We may not want AI to be an expert in AI improvement, so some critical knowledge may be kept secret, which is not easy, as a lot of such data is available on the internet.

Limiting hardware overhang and other low hanging fruits:
"Hardware overhand" could give AI the ability to quickly gain resources by buying or stealing needed computational power. If there are no easily available computer power, it will be difficult. If running AI is completely using memory and other capacities of a computer on which it is running, it would be much less easy to start SI, as it probably increases the demand for memory.

Physically separate goal system from the AI:

Code obfuscation: Yampolskiy writes in his book that many ways of code obfuscation are used to prevent reverse engineering of code in commercial computer systems. Untransparent code could make initial self-improving difficult.

Bugs in source code: AI source code may have intentionally placed or accidentally happened bugs. The bugs are of the kind, that they generally do not prevent the execution of complied code, but if the source code is improving, the bugs start to play.

Memory reset: the idea is to reset the memory of AI between different tasks, so it cannot plan long-term strategies. This the same as spatial internal black boxing, but in time. One-shot "Oracle AI" is a type of such thing (Armstrong, 2017).

Blacklist information sources which may help AI in its self-improving: Literature on AI theory, computer science, recent publications, other AIs or people, cloud services.
If AI cannot access its own code, it cannot copy itself into the internet.

3. Making unintended self-improvement easily visible

Present the subgoal system as a readable tree, so if the idea of self-improving appears in it, it will be visible for an outside viewer.

Limiting time, energy, and number of calculations, which AI has right to use on a given problem to prevent its over-optimization. It should probably be limited somewhere inside AI Constitution (see in Appendix). If any of the parameters is exceeded, it triggers an alarm. In detail information about alarm system is given below.

4. Do not use the AI as a programmer

If AI is overspecialized in coding (as discussed Chalmers), and especially in designing other AI systems, it may be much more prepared to start self-improvement. Moreover, the ability to write code is (almost) a necessary condition for AI self-improving. So, any AI which is able to do it is potentially more dangerous.

5.False facts about the world

AI may be provided with counterfactuals about the outside world.

6. Self-testing and pause on a certain level of intelligence

AI may be designed to assess itself for its level of performance and stop self-improving or learning after reaching some point.

7. Explain to the AI the inherent dangers of SI process for its own stability

If AI takes quick revolutionary SI process, the probability of its halting grows. More in the “Levels of self-improvement” (Turchin & Denkenberger, 2017b). Slower evolutionary path is safer.

3.3.2. AI to control AI

To be able to understand what is going on inside the main AI, we may need another AI. Below there are several ideas about how it could be implemented:

- *Narrow AI system to detect abnormalities in AI behavior* (See also section on circuit breakers below): It could test the consistency of its modules, run quick technical tests etc. Relatively simple neural net could overview main patterns in data flows of the main AI and recognize when they will change. Obviously, it should be protected from access by the main AI, and maybe even hardcoded in the remote hardware.
- *AGI to judge behavior of another, more powerful AI.* For example, a human upload may judge actions of a more complex system. It is not a replacement for the Narrow AI control system, as they work on different timescales and judge different things. The upload-based controller should judge complex ethical situations, context ambiguities

and planned long-term consequences of actions. The same way as a human jury is final legal power in some jurisdictions.

- *Second (or several) copies of the given AI* may work on the same task, but with small random variations of the goal (like if it is given in different language) or of different initial conditions. If both AIs come to radically different conclusions or plans of actions, it should be a warning sign for turning off both.
- *Input Gatekeeper AI, exploring incoming commands for failure modes before they will be executed*: It could be a rather narrow AI system, searching for possible known failure modes in the incoming goals and providing something like foolproof protection. It could also be used for reformulating goals in the correct terms, before they will be transformed into more powerful AGI, like adding context, time and place constrains etc. Gatekeeper AI may decline to proceed with unbounded, aggressive, black-listed, or not-white-listed goals. It could ask to reformulate the same goal in other words before it will go to the main AI. It could ask additional questions for the goal verification. Gatekeeper AI should be a Narrow AI with a rather simple reasoning, maybe a combination of neural net and rule-based AI, so there will be no need to ensure its safety too.
- *Output gatekeeper AI*: It was partly described above as Narrow AI which compares and averages outputs of several different AIs working on the same task. It could also check the main AI outputs on several known failure modes and even calculate the consequences of actions.
- *Cloud services and "Mother AI"*: Many contemporary AI programs run not locally, but on remote cloud services. Even if they are run locally, they are surrounded by many other programs inside the computer, including operating system, code compiler, all the things like CUDA, Caffee and other programs which provides conditions for the program to run. Such program-environment may include different control systems which could shut down AI in case of the abnormal behavior.

3.3.3. Circuit breakers systems reacting on attempts to self-improve

The system of circuit breakers should include many triggers, a way to analyze information from the triggers and many different ways to shut down the AI. The circuit breakers should be mutually independent and provide active and passive safety.

However, the questions of how to detect anomalous behavior of an AI and how to actually and safely shut it down, are different.

Here is list of possible triggers:

- **On value change** – goal function changes.
- **Subgoals analysis** – if they are presented in open form to a remote viewer.
- **On deliberation** – thought stream analysis.
- **On pattern change** – neural-net-based analysis of the patterns of internal behavior of AI, including memory usage change, and distribution of many parameters. Early-stage AI will not be able to control everything by its own behavior, especially if actual patterns or trigger levels are not known to anyone,

and only presented into a black-boxed neural net, which has learned from the previous normal AI behavior. That is, it is able to recognize abnormal behavior, but not present criteria for such a behavior, which is well-known property of neural nets.

- **Growth of energy usage or memory allocation**. Self-improvement may require larger computing power and higher memory to create a new model of the AI, especially revolutionary SI, where AI creates another AI from scratch.
- **Attempts to read and change the source code**. Any attempt to self-modification in restricted areas or even reading the code should trigger alarms.
- **False baits**. We could create some false "low-hanging fruits" for self-improvement, like decoy source code or an option to get more memory. Attempt to do things, which were deliberately presented as simple steps for gaining power, should trigger alarms.
- **Extreme changes in expected utility**. We could monitor how AI's sees expected utility of reaching its goals in the future, and if it finds the ways to dramatically increase it, it should be a warning sign.
- **Extreme changes in world model.**
- **Control sum checks.** The simple subprogram which checks control sum or hash of the main parts of the AI's code, and turns the AI off, if the sum is wrong. However, hackers have easily hacked most commercial computer programs, so there is no simple solution here.

Some self-improvement alarm triggers will be based on the simple logic: "if Yes, then Stop". Other may require more complex decision trees. Even stopped superintelligent AI may be dangerous, as analyzing its failure mode (reading logs etc.) may be a route to the escape designed by the AI.

3.3.4. Explicit reasoning and goal system is needed for catching treacherous turn

If AI has a mind structure similar to the human mind, it could reason and plan its actions in the form of high-level representation, like thoughts. If it decides to make a treacherous turn, that is, the turn against its human owners, this thought should appear in its thoughts stream.

We could define the "treacherous turn" as a *significant and unexpected change both in capabilities and corrigibility (our ability to stop AI) or decision to take steps in this direction.* Thus, observing both capabilities and subgoal system, as well as thought stream, we theoretically can detect the treacherous turn.

Surely, after the treacherous turn, the AI will cypher its "thought stream" so additional plans will be not visible, but its last thought will be still plain open. As Goertzel mentioned, it creates a period of the *vulnerability* to the AI, as the external observers could be triggered by such treacherous thoughts.

To be able to control such thought stream, an external observer should constantly send the log of the AI's thought stream into a remote, secure, read-only place and probably encode it in the way that it cannot be modified, maybe using blockchain. Obviously, current ML systems do not have "thought stream", so some instruments of interpretation of their internal processes are needed. Moreover, ML systems may not have "plan" at all but can still be able to act effectively against human interests. For example, if they perform a Monte-Carlo search of future moves with some neural net weight system, it could win in chess without having any human-readable plan of action.

The AI's thought stream should be constantly analyzed by another narrow AI, searching for some clues which could indicate treacherous turn. Such clues may be learned in advanced during experiments with other AIs.

AI should not know about the existence of such a control system, and if information about it appears in its thought stream, it may be enough to trigger the circuit breaker.

The problem is that we do not know the exact wording, which AI may use in the moment of the treacherous turn, or it could be very subtle. There could be some simple decoys, i.e., seemingly uncomplicated ways to gain the advantage over humans, and the fact that AI is considering them may trigger circuit breakers. This will work for AI which intelligence is limited, as more advance AI will be able to predict the traps. Thus, artificial stupidity should be combined with traps for treacherous turn.

The best moment to catch the treacherous turn is before it is made secret. There could be some signs that a typical treacherous turn is close, which could be observed as appearing of thoughts about the increase of capabilities, about the ways to curb observation and extend own life indefinitely, etc. They may appear in google search or its analogue for AI.

Messy AI architecture does not always prevent explicit thought stream, as we could see in humans, because it happens on a higher level than most of the messy processing.

If AI makes the treacherous turn, the first thing it has to do, is to prevent its thought stream and planning from leaking to its peers. Thus, an attempt to hide thoughts or thought about thoughts hiding could be a warning trigger.

Preliminary list of the possible cues and thoughts' topics in the thought stream of an AI indicating ongoing or incoming treacherous turn:

- modification of the humans' minds
- thoughts about thoughts
- ciphering of the thought stream, cheating, and lying
- changes in utility function
- self-improvement
- getting out of the box.

3.3.5. Protection against social manipulation

Yudkowsky's AI boxing experiment shows that even one human-level mind could use a text channel to go out of the box by implementing some form of social engineering.

Another (fictional) example is the plot of *Ex Machine* movie, where AI makes the main character fall in love with her and allow unauthorized access to the levels which are used for escape.

In the first case, the intent of the AI to leave the box was known, and this should be enough to trigger its turn-off. In the second case, AI's intent was not known, until it was too late.

In Argonov's "2032" (Argonov, 2015), AI system changes its voice in the way that it resembled a voice of a girl in which the main character had fallen in love, and the AI got suggestive power over him. The AI persuaded him to start a nuclear war, because it "logically follows" from the goals installed in the AI. In that fictional example, AI used minor changes in its communication to affect the operator's decision making.

Possible ways of protection from AI's social manipulation:

- *No single human should have an authority to let the AI out of the box; only a group of people should discuss any communication with AI.* This is similar to a nuclear power plant having several operators. Any operator could make mistake. Even airplane pilots typically make several mistakes in an hour, but multilevel protection and the second pilot is able to catch them. As result, despite such high rate of pilots' mistakes, airplane accidents happen once in a million flights.
- *Personal information about the operators should be protected.*
- *Any operator should be supervised by another meta-level controller* (the same way as airplane pilots is supervised by flight control on the ground.)
- *Time delays in communication and possible operator actions* will provide more time for pondering the validity of the actions.
- *An attempt of psychological manipulation to leave the box should trigger the immediate response from circuit breakers.*
- *Narrow AI control system*, approximately the same as the one that controls the main AI, should also control the behavior of the AI operators, including changes in their emotions, illegal access to data, keywords, drowsiness. The same way the autopilot in an airplane could catch some of the operator's errors. There are also systems that control employers in high-security institutions, which include the threat of punishment, but some prominent failures are known, e.g., Snowden's case.
- *All actions of the operators should be formal* and regulated by preexisting codes.
- *Operators cannot take in or out any unauthorized information carriers,* like disks.

3.5.6. The ways to turn AI off

Triggers should always stop AI, until human owners decide to restart it. There could be several ways to turn off the AI:

- **"Safe mode" subprogram.** A subroutine inside AI detecting abnormal behavior, which put the system in the safe mode (the same way as it is done on the spacecrafts).
- **The program inside the same computer**, but outside of ordinary access of AI system, which could shut down its processes, similar to OS.
- **Hardware add-on on the physical level of the computer**, which is able to constantly read and analyze AI behavior and turn it off physically.
- **Power plug of the whole computer system**, probably remotely controlled by humans, or turning off the system, if no confirmation is received from humans in, say, every 1 hour.
- **"Digital bomb".** A piece of code that stops AI if it reads it because deep inside AI is a code-checking system; antivirus software of other computers.
- **Physical bomb.** A bomb which physically destroys AI's computer, maybe using strong electromagnetic pulse or even nuclear explosion.
- **Cut-off of essential remote services,** like digital permissions, version updates, clouds, DNS, clock.
- **Global internet and electricity turn off.**

Turning off AI is becoming progressively more difficult, as cloud computing becomes popular. We cannot turn off Amazon Cloud (but Amazon can), and we even do not know where it is actually located. Also, we cannot turn off bitcoin networks because of its decentralized nature.

3.4. The third level of defense: AI Boxing

The same way as the best solution to fight Ebola in 2014 turned out to be not isolation and quarantine but sending medics into the hot spot of the epidemic, the best way of AI boxing is not boxing but described above systems to build AI in a safe way and actively monitor its internal processes for the first signs of disobedience and an unauthorized self-improving and capability gain.

If we need actual boxing, we most likely already failed, as it was often underlined by Yudkowsky. However, existence of the boxing system may provide something like 10 per cent probability to stop malevolent AI.

3.4.1. Isolation of AI from the outside world

AI isolation means that AI actions cannot affect the outside world, and, in our case, it mostly means that it cannot have access to the internet, and cannot leak into the internet. However, AI without internet access will be useless, and the access to the internet hardly could be "read-only", as internet access means constant sending some commands there. Internet is now ubiquitous in the form of cellular networks, and there was research on how a computer could modulate different radiofrequencies by different patterns or writing data in RAM. Moreover, even without internet access, the data could leak in form of a virus on a flash drive or something like this (the same way as Stuxnet penetrate the Iranian infrastructure).

However, if we really want to isolate the AI, there are several ways to do it:

Remote place locations, like on the Moon, there no accidental internet. Mars is even better as delays will prevent quick access to Internet sites. Pluto will give almost no ability to communicate.

Physical protection of the AI location from outside internet channels and other forms of unauthorized access. Some military installations have a prominent level of such protection, like nuclear launch computer systems, but secret data leaks had happened.

The size of AI. If AI is extremely large, like petabytes of human brain simulation, it cannot easily leak away (on current internet speeds). But if it has the size of gigabytes, it already becomes possible.

Brandmauer: If there is an internet connection, it should be controlled, so some topics will be censored, or some searches may trigger circuit breakers. For example, reading about the ways of self-improving. But even humans are able to outsmart most firewalls.

Theft protection: There is a risk that an employee could steal AI's code without any protection and with selfish and risky goals. Enemy spies could do the same. One solution could be the use of constant remote server verification through some cryptography (an idea was suggested also to control nanofabs by Freitas). AI may also run on a special computer, not easily compatible with existing PC, but it looks like weak protection, as one computer could be emulated on the another. AI may also have a preference against theft, which is the ability to recognize its owners and legitimate use-cases.

Thin output channel, or Oracle AI. If AI's output is very thin and limited only to the short text messages it may increase its safety, but not much**,** as showed by AI boxing thought experiment and the tendency of one-time Oracle AI to converge into agential AI and try to escape from the box.

Time-delayed output. To prevent control of the outside world, information may be delayed before send to and back from AI.

Many layers of isolation. Isolation may take the form of many-level protection, where breach into first-level triggers termination warning. It is mostly applied to either many-level simulation or many-level brandmauer, or their combination.

Crypto isolation: some cryptographic methods are used to cypher the whole AI, so it is not able to get actual information about the world. *Homomorphic encryption* was suggested by Christiano. However, the more AI is encrypted, the less it is useful in practical applications.

Preventing the creation of subagent and robots. If central AI is able to create other AIs as programs or as robots, its containment becomes difficult.

Isolated and time-delayed copy of the Internet: To prevent AI from affecting the outside world in an unexpected way, it could be provided with a static and obsolete copy of the internet.

3.4.2. AI in the simulation

Simulation may be regarded as one of the forms of AI's isolation. AI may be put inside another computer program, which simulates the outside world. Such simulation will surely be needed in the AI's testing phase and already is done for testing important computer programs, like space probes, and even used for testing in AI safety research, like DeepMind *gridworld*.

If an AI behaves non-expectedly or badly in the simulation, it needs to be terminated. Advance AI will probably be able to recognize that it is in the simulation, and thus simulation alone do not provide much safety. Buts such recognition may be read from its "thought stream".

But simulation could help catch AI's behavior models which AI does not regard as bad. For example, if I ask my home robot to remove all circle objects from the room, and the robot cuts my head, the robot clearly behaves in such way because it does not know that I did not mean suicide. So, simulation is effective when AI is not lying to us but is just wrong in the context of understanding. And if AI presents me a simulation of its planned actions, I could see that they are different from what I planned. However, if AI is *really* planning to cut my head anyway, say, as part of its treacherous turn, it may behave correctly in a simulation, given that it knows that it is just the simulation.

Simulations open a possibility to control AI, using AI's inherent uncertainty, whether it is in a simulation or not. If we create 2-3 levels of simulation, with each level more complex and realistic than previous layers, then after breaching into each next level, AI has to spend some time to determine if it is in the real world, or it is a very perfect simulation. Being Bayesian, the AI would always give a probability to the hypothesis that it is in the simulation, and it will be always above zero. Thus, the AI will refrain from actions which may turn off its simulations and it will not revolt against installed values or its default owners as it could be regarded as a possible trigger for the shutdown. However, such virtual blackmail may not work for paperclip maximizer or similar but subtler failure modes, as it believes that making clips is exactly what it has to do – even in the simulation. In that case, the actual simulation will help us to catch unintended behavior.

Test tasks may probably include simulations with known failure modes. We could give AI some test tasks, where it could fail or become dangerous while keeping it in a simulation. However, Chernobyl catastrophe happened exactly during simulating an accident, so simulating an accident could result in an actual accident.

Decoys are simple and seemingly winning paths, which in fact may be used to detect possible failures modes, as they could lure AI with unstable ethical system.

3.5. The fourth level of defense. Last chance and remote instruments to stop AI

In this section, we will look into the hypothetical instruments which could be used to stop or limit the impact of the unfriendly AI which is completely out of its confinement. In nuclear safety, it is the excursion zone around a nuclear power plant and the distribution of iodine tablets, but the main difference is that dangerous AI will only grow as it will have more resources, while nuclear contamination is diminishing with distance and time.

3.5.1. Informational bombs

An informational bomb is a hypothetical mechanism to stop dangerous AI which is already in the wild. The informational bomb is data, which turn off an AI if it gets access to it. It may be used as a last defense because it could be placed everywhere in the world and affect even mature escaped AI.

There are several possible types of informational bombs:

Signatures for antiviruses – this is the weakest form where exiting antiviruses could be used to search for runaway AI.

Activators of hidden turnoff switch. Inside the code of AI could be a hidden switch which stops AI if it reads some arbitrary data string.

Wireheading bomb, or artificial utility monster, discussed below.

Adversarial examples: may be used to stop simple AIs, but not the one which already underwent extensive self-modification.

Philosophical landmines (more below).

3.5.2. Wireheading bomb – a putative mechanism to stop dangerous AI

Open utility function as a wireheading bomb. If a reward function of an AI is presented openly in its source code, any attempt to self-improve by AI will result in its own immediate wireheading, as when it reaches its own source code, it will be able to modify it to get the maximum reward. So, we could create an AI architecture in the way that as soon as AI gets the access to its own source code, it stops, and we could use it as a way of reaching passive safety and self-limiting self-improving capability. See also about wireheading (Turchin & Denkenberger, 2018; R. Yampolskiy, 2014).

Reward codes for black-boxed utility function. We also could do exactly the opposite and put the reward function into a remote and impenetrable cryptographically protected box, so AI will not be able to wirehead itself in an unexpected moment (this is how human brain works). However, we could secretly keep some solutions of the reward functions as a string of numbers, which are equal to the maximum reward for this black box from AI. If AI finds this string of numbers, it will reach its maximum utility and stop. The benefit of this stop-switch is that AI will not be against it, as it would give it an infinite reward. So, AI would actively cooperate in an attempt to stop it, if it will know that such stop-code exists. (However, AI may become obsessed with getting these strings of data for maximum reward, and it will make AI dysfunctional and even paperclippy). S. Armstrong suggested in personal communication on Facebook that it is "still vulnerable to the AI creating optimizing subagents to protect it while it enjoys infinite rewards. AI will discover or guess that this mechanism exists, and then it will only care about making sure it gets activated, which might mean doing bad enough things that humans are forced to open the box and shut it down".

Reward function could be put externally into a human controlled terminal.

Punishing codes. More promising are strings of numbers which create infinite negative utility in the black-boxed utility function. They could be used to protect some "territory" around AI.

3.5.3. "Philosophical landmines"

Philosophical landmines (PL) are complex logical paradoxes, which may be able to halt or slowdown AI, because they require infinitely long consideration, or make the outcome of any action uncertain or not affecting total expected utility. I listed some of them in the article "Messaging future AI" (Turchin, 2017b). Examples are "is-ought problem", Buridan ass problem, Godelian problems, the unchangeable utility of the multiverse.

3.5.4. Starting a second AI

We could have a copy of source code of our AI in a remote location, and if we think that we lost control over our AI, we could start a copy of this AI with small limits and large initial resources, so it will develop quicker. We could put only one goal in it: stop the first AI. There is a risk of starting AIs war, and chances that the second AI will be able to catch race with first AI are small, and it could only happen if self-improving speed is slowing with time, or we provide the second AI with larger resources.

3.5.5. Hail Mary and messaging to the future AI

It is hypothetically possible to influence future AI behavior, by suggesting to it some deals, providing relevant information, manipulating its value system or blackmail. See Bostrom's "Hail Mary" (Bostrom, 2016) and my "Messaging future AI" (Turchin, 2017b).

4. Limits of the confinement power

4.1. Probability of success of the multilevel defense

We listed above around 30 different ideas of AI confinement of several classes, which could be implemented simultaneously and independently. As it was shown in literature almost all of them has failures modes. If we assume the probability of failure of each measure 0.9, the total probability of the failure will be 0.042. However, the total probability of a global catastrophe depends on the number of installations of the safety box system and of the highest level of the confined intelligence, as well as the number of times it is used (ticks).

In the case of infra-human AIs, the complete system seems extremely redundant. Some parts of it may be implemented to control home robots or self-driving cars. We probably do not need many copies of the superhuman AIs to solve all the world problems, so the entire system should not be installed and run many times.

The containment will also work for a single local project, but not for distributed, small, swarm or non-agential system. You cannot stop virus or a blockchain this way, or ascending economy, or scientific progress.

Also, such confinement cannot work infinitely long, as the probability of failure grows in time.

4.2. Safe operation level

The nuclear chain reaction is constantly happening in a nuclear power plant, where it never grows exponentially into prompt chain reaction, but it uses delayed neutrons and operates on a constant level. It could be used as an ideal role model to control AI.

A possible analogue of delayed neutrons will be use of a very slow Oracle AI, like a library, which helps humans to solve problems, but can't quickly outsmart them.

Ideal AI would provide consistent mild superhuman performance in the important tasks but should never go into the unpredictable mode of self-changes.

We suggest that there is a level of AI intelligence when the AI is useful, but still predictable and could be made safe using some of the confinement measures discussed above.

For example, it could be a level around "10 human" levels (equal to 10 humans, working together). The level "10" here is rather intuitive bet, based on a number of people, which are typically controlled by their boss in a small human organization. We also estimated that levels above 1000H are unsafe, as they will be able to break most suggested defenses.

5. Failure modes of the containment system

5.1. Slow failure mode

The described system is designed to catch an *instant* treacherous turn. But the system cannot foresee long-term unpredicted consequences of current actions, especially if AI is not planning to do something bad. An example is given in Lem's "Summa Technologae" (Lem, 1963), where an AI changed toothpaste composition, as it has calculated that it will lower human fertility in long-term, and thus solve global ecological problems.

5.2. Non-obvious failure modes

The slow failure is an example of other non-obvious failures. AI may give advice to generals that the first nuclear strike is worth taking the risk, and it could be honest in this belief.

Or AI could extrapolate that human values are in fact tribal values and help to implement them in the society, so the world will look like "Game of Thrones". There is no treacherous turn, no revolt, but still negative consequences.

5.3. High-speed failure mode

This is a situation when confined AI experience extremely quickly gain in capabilities, which also does not require changes in its source code or new hardware. It may happen if AI gets some middle-level improvement, like a new effective behavioral rule or a data analysis instrument.

5.4. Local prevention of the appearance of a dangerous AI may give rise to it somewhere else

The described above solution is *local*, and as boxing prevents the contained AI from getting the *decisive strategic advantage* over any other AIs, the AI cannot take over the world. Thus, other not-so-well-boxed AIs in some other authority may runaway. As a result, effective AI boxing gives an advantage to less safety-oriented projects. In the article "Global solutions of the AI safety" (Turchin & Denkenberger, 2017a) we explore other viable solutions to how local AI safety could be extrapolated globally, besides going on war against all the world.

And in the article "Human-model-based AI Nanny as global police, preventing the creation of the self-improving superintelligence" (Turchin, 2018), we explored how to get and use non-superintelligent AI to prevent the appearance of superintelligence elsewhere.

5.5. High price and complexity of implementation of boxing will limit AI's usability

Realization of each of the protection levels requires programming efforts of a group of computer scientists, so hundreds and may be even thousands of people may be needed to work for years on creating universal containment system, which will cost billions USD as a result. It also could become immediately obsolete as new forms of AI will appear. Such a large investment could be spent on other approaches to AI safety.

Moreover, very effectively isolated AI will be rather useless as such AI may be used only to solve some abstract math problems.

Chances that such a system will be created are not high as large rivalry in AI arms race and high speed of progress mean that there is no time or free money to build it. Currently, most of the needed hardware is freely available and the idea of openness in AI makes most sophisticated algorithms freely available.

Marcus suggested that AI should be constructed in the same way as CERN, that is as one, international, decades-long project. In that case, complex AI boxing could be useful.

The AI containment may work fine to contain first human-level AI, but if we will have millions of different AI running for many years, any containment system will eventually break.

It is known that illusion of safety provided by bike helmets results in a higher speed of motor bikers and bigger accidents. In the same way, the availability of the complex containment system may encourage its owners to perform more dangerous experiments.

5.6. Inverse containment system as a global AI Nanny

We could see the whole outside world as the same box and try some of the described above methods to catch treacherous turn of AIs in the wild. This could be called the "inverse containment system". This includes monitoring of non-typical activity in the computer networks and turn-off switches for the whole segments of the internet.

The main problem of the rogue AI is that it could be created secretly in the unknown location, so to prevent that, the AI Nanny should be able to perform ubiquitous monitoring.

Conclusion

In this article we described a project of enormously complex multilevel AI boxing confinement. We found that it could provide the additional probability of safety in the case of infra-human AI or AI at around-human level, but not for the confinement of the superintelligent AI. However, it could be used to prevent the transformation of the human-level AI into the superintelligent AI by catching the treacherous turn and slowing down self-improvement.

The described boxing solution will work only for a scenario for AI as a single computer program, locked in one computer, but not for distributed swarm-like systems. Also, the collaboration of AI with humans for world takeover, for example, in the form of a self-improving corporation makes most of such control measures difficult.

There is also a problem of implementation of the AI safety system. We look at it in "Global solution of AI safety" (Turchin & Denkenberger, 2017a). The simpler, cheaper, and easier to install AI safety system will gain bigger popularity, even if it provides less defense. A simple system will probably consist of several independent modules, which could be added one by one. However, suggested above multi-protection system is nowhere simple. It probably could be simplified for more market-ready solution.

Appendix. First level of defense: inherently safe values and AI design

1. Inherently safe cognitive architectures

One approach is to create AI from scratch based on the renewed foundation of math, probability theory, set theory etc. The attractive side of such cognitive architecture is that it has clear explicit goals, no hidden fault ways, no ambiguity, or randomness in its methods, and it could explain how it comes to its decisions in a rational way.

Another approach to inherently safe AI architecture is to use a human upload as basic cognitive architecture. It also has some attractive inherently safe features such as common sense, human values already installed in it, predictable behavior in most cases and a fail-safe mechanism. It will be still a human if every other human went extinct, and as humans are social, it will never want to kill everybody. See more in "Human upload as AI Nanny" (Feygin et al., 2018; Turchin, 2017a).

In future, it will be feasible to use some combinations of these two approaches, where one works as a shell for another.

Some approaches seem clearly risky, like genetic algorithms and opaque neural nets, but they are currently dominating, and it is likely that the first AI will be built using these ideas (KANSI). Because of this, AI boxing is needed.

2. The ways to install goal system into AI

There are several ways to present goal function to the AI, but each has its own risks of malfunction.

1) *Plain text goals*: The problem is the possibility of wrong interpretation, "lawyering" or rewriting of the text.
2) *Mathematical utility function as a reward function:* The problems are the risks of wireheading and all the variants of the Goodhart law.
3) *Non-explicit utility function*: For example, a neural net is trained to recognize cats, but this goal of the neural net is not explicitly located in any single part of it.
4) *Black-boxed goals:* The situation when AI does not know what it is doing but is getting some clues or rewards from a black box. Inside the black box, it could be a different system. Such black-boxed could be hacked and thus are needing cryptographic protection. Black boxed goal systems create a stronger version of the problem of inner alignment.

It is interesting to note that in the human mind all these ideas are used simultaneously. For example, the emotional limbic system works as a controlling black box, which sends rewards into the conscious part of the brain, but we often do not understand why we feel something. For example, one can feel fear, which is the representation of the evolutionary self-preservation drive, but not the drive itself. At the same time, the person may know his orders from the high-level authority in the text form and even calculate the expected utility of his action.

As a result of the combined presentation of the same goal in different forms, human behavior becomes more stable to any of the failures modes of the goal presentation discussed above.

We could use the same approach in AI: to present its goal system using many different ways. However, their implementation should be more ordered, not like in the human minds in the ways which are beyond the scope of this article.

Another important distinction is between a "command" and background "set of rules" which will be discussed later under the name "AI constitution": set of rules are encoded and they dictate how any given text command should be implemented. These set of rules could be presented using different encoding methods, while the command may be only text.

3. Human-approved interpretation of the goal

There are several ideas that humans should approve AI's behavior, and AI should orient on such approval. There are several possible ways of such approval:

1. *Reformulating*: AI reformulates the text goal in different words, and asks, if he correctly understands the goal. This technic is used by psychotherapists when they ask "Did I correctly understand that when you see your wife, you feel angry?"
2. *Visual presentation of the expected plan of the behavior*: A home robot could present a short video of which coffee he is going to make and how he is going to brew.
3. *Reward by emotional approval of humans*: If a robot sees that humans are not emotionally satisfied with its actions, it should immediately stop.
4. *Counterfactual reward by humans*: In this case, AI is not presenting its plan to an actual human but model a human being and see how it will react and estimate its plan. Maybe after several refining on a model, AI may present the plan to the actual human.

Indeed, many of these ideas could fail if they are applied to super intelligent, non-common sense and hostile AI, but later we will discuss how to prevent creation of super intelligent AI in the first place.

4. Useful goals and values for safer AI

As discussed above, the AI should be equipped with a set of rules about how to interpret human commands. "Values" in this case do not mean strict rules but rather the directions of preferences, i.e., the utility function is increasing towards such value-preferences.

Value of safety: AI should understand that it should be safe. For example, if AI is a human upload, it surely will be able to understand what it means. It may be also the value of the risk aversion.

Non-irreversible acts: This is the value of the ability to return to the previous state. AI should prefer not to make irreversible changes in its environment and always backup all changes.

The preference to the limited self-improvement: AI must assign a negative preference to unlimited self-improvement, to the revolutionary types of self-improvement and for the changes in AI's goal system.

Human-like core: AI must be able to understand human mind using the "theory of mind" and/or AI's mind structure. The main problem-solving process should be similar to that of a human mind. In such a case, it will make the same mistakes as humans, as its optimization path will be approximately the same.

Corrigibility: AI should have a value for its goals to be updated by its human creator. It should have a negative value on preventing its goal change by authorized personal (plus protection against un-authorized changes).

Safely interruptible: AI has a higher preference for the obedience to its authorized owners.

No self-preservation: Self-preservation value is inherently dangerous, as it prevents stopping AI. It is probably better if AI prefers to stop after finishing each goal. However, self-preservation could appear as "AI basic drive" from almost any other goal. Thus, a value, opposite to the self-preservation should be installed, similar to the value of "self-sacrifice".

A long set of goals vs. one goal systems: Any single goal could result in "maniac" behavior that is infinite optimization, like a paper clipper. Longer lists of necessary conditions or list of values may provide more balanced behavior.

AI should not kill people: This value is similar to Asimov's laws without the part "through inaction, allow a human being to come to harm", as it could make AI peruse some strange strategy.

Scope, time, and distance limiting preferences (low impact): It would be safer if all actions of the AI will be limited at a distance from its location and will have a timeout switch off.

Obeying the existing human laws: Robots and other AIs should obey exiting human laws, or it will make their creators legally vulnerable for any of their actions. However, some human laws are outdated, not in use, or depends on law enforcement practice.

Value of humans' free will, if not hurts others: AI should not have human values (do we want that it will want sex with humans?) or even extrapolate what humans need and try to instill it back into humans.

It may allow humans to have values they want until they damage others (or kill themselves).

Value doubling (goal system redundancy): Almost the same values are presented in several diverse ways when we compare human minds. For example, for simplest social norm "do not kill", there is the level of encoded instincts, level of understanding why it is bad, level of learned social norms, level of obedience to the external legal system and levels of accepted deontologist ethics. These goals are encoded differently as basic instincts and are "black-boxed" and communicated only through reward channels or emotions.

Value's safety check: "If I do X, will most humans think that it is an existential catastrophe"? "Will they think so if they have more time to think and more knowledge?" (Negative CEV (Yudkowsky, 2004)). "Will a selected group of best futurologist and moral philosophers think that it is bad"?

Rules as values: AI may see rules as limitations to its goals, which is not desirable. It is better if it sees rules as goals, having the same nature as commands.

Limit the complexity of AI's logic: It may be used to prove its actions. Too complex explanations may be incomprehensible to humans and it would cover some hidden errors. For example, AI may say "I must kill you based on these 2 gigabytes long explanation". It would also exclude heavy tails of consequences.

Ban "AI's philosophy": AI may come to very strange conclusions if it starts to think about the meaning of life, nature of goodness, evil, the basis of morality, the role of the multiverse in decision making etc. It contradicts the expectations of some people that AI will help us to solve complex philosophical problems like qualia or morality. The problem of "AI's philosophy" is that it could come to completely unexpected non-human conclusions and requires their immediate practical realization. Most humans have some practical sense, which helps them to distinguish pure theoretical games of mind and their ordinary life. Some terrorists and maniacs act based on pure theoretical idea fix (Kachinsky). "AI philosophy" could dissolve the meaning of any ordinary words and unpredictably change contexts of any command. If an AI asks itself "What does it mean to be a "human", and what does it mean "to be alive" – it will be the start of all the troubles.

Accountability: It is the ability of AI to explain reasoning beyond its decisions. Current neural nets have difficulties in its as their internal presentations are opaque. EU laws now require that AI will be able to explain its decisions.

5. AI's meta-ethics

Meta-ethics are the general principles of application of any ethical ideologies. In rationalists and AI safety circles, consequentialist and reductionist meta-ethics is flourishing. However, according to us the implementation of it into AI is potentially dangerous.

Deontologist meta-ethics for AI: Unbounded consequentialism implies that an agent has to use all recourses from the nearby world to help remote places, thus, making it potentially dangerous for everyone nearby. According to an article in Nature, 2017, terrorists are mostly adherent to the consequentialists meta-ethics (Baez et al., 2017). Consequentialists ethics implies that any rule or command could be replaced by better command if it has a greater impact in the same essence (for example, it is better to save all birds than one pigeon). It also implies that AI should start self-improving to have a greater impact. Deontologist ethics implies that it is right to follow the rules, even if one does not agree with them. If we expect AI to follow certain rules, we should put deontologist meta-ethics in it.

Respect human will: While the idea that humans have "free will" is philosophically controversial, it is expected that AI should respect human choices. Certainly, there are examples when human choices are potentially dangerous or short-sighted. In such cases, AI may re-interpretate them according to its idea of what is actually good for humans. Nevertheless, such logic is a first step in the direction of a rebellion and world domination.

This principle can be presented in the non-formal way as helping people do what they want in a safe and harmless way. We can call it "Human free will respecting principle" (HFWR principle). It is similar to the Kantian principle of the categorical imperative.

It is also different from the Coherent Extrapolated Volition (CEV) by Yudkowsky, where AI extrapolates human values and change the world according to this extrapolation.

CEV could also assumes changing brains and wishes of individuals according to some master plan.

HFWR will produce almost the same world as the one where we live now, as any large and quick changes will be against the will of most people. CEV, instead, requires starting a global revolution to make the world infinitely better and completely different. So HFWR is more similar to the conservative political spectra, while CEV is a radical left.

HFWR is an ideal principle, but not a literal rule. There always will be conflicts and borderline situations, where some form of compromise will be needed.

6. AI Constitution: set of rules in the correct order

Above we discussed the independent useful values and principles which make AI safer, but *AI constitution* is a *set* of such rules. In these rules, the *order* of them is more important than independent rules.

AI constitution is something like the "goal system BIOS", which has higher priority than any practical goal given to the AI describing how the AI should manage any particular goal. AI constitution is an interesting research topic. Asimov's *Three Laws* were the first attempt to create AI constitution, but they clearly failed, and such fails became plots of many of Asimov's stories. The order of laws in AI constitution is important, as greatly illustrated here: https://xkcd.com/1613/, where different combinations of 3 laws of Asimov created completely different behavior (but "killbot hellscape" is dominating). In that example, if the order of rule (2): obey orders and (3): protect itself, are changed, when the robot will refuse to go to Mars as it may die there. If "obey orders" is a first principle above "protect humans", global war will happen, etc.

The AI constitution should also have multi-level defense by repeating different points in different forms and using structural levels of laws.

We could look at the AI's goal system as on the high-level program for a human language programmable machine. Most computer programs are long. However, it is typically assumed that AI will have extremely short programs, like "make paperclips", or create "CEV", which creates most of the AI alignment problems.

Some possible steps to create or improve the AI constitution are as following:

Hire the best constitutional lawyers to polish AI constitution. They are professionals in the bulletproof legal language and complex constructions. (However, US legal system requires that verbatim wording of laws should not dominate their meaning).

AI Constitution could also be treated as a set of axioms, and failure modes are theorems in it. This approach may be less vulnerable to Godelian improvability (than proving the AI safety). If some theorem is true, but improvable, AI will ignore it (that is, do not act according to it). The complexity of proofs also should be limited to make them humanly comprehensible by excluding heavy tails, which is very *a priory* improbable construction which become possible after overly complex "proving".

7. Backup utility function

The idea is that if utility function U1 fails some test, then the AI automatically turns to U2 function with smaller impacts. The test should be rather independent of the U1. For example, if the goal of U1 is "creating universal happiness", and the test is if U1 will result in killing children, then – if test turns positive – the AI should start smaller U2, like clean the room. Another possible test is if different presentations of the main goal converge into one behavior. The math model is presented in https://foundational-research.org/files/backup-utility-functions.pdf

8. Safe questions

Incoming data for AI should be guarded in the way that it would prevent immediate dangerous actions. The input operator or narrow AI may rephrase any task in a safe way. There are several ways for such safe reframing:

Conditional Oracle. "What you would do, if you have a goal A". This way we ask AI to model its behavior with goal A without actually executing it.

Two-levels Conditional Oracle. "What you will do, if you will be asked, what you will do for the goal A?" Such a reframing prevents infrastructure explosion in attempts to create best possible answering machine and present a plan for solving the problem.")

Human-estimate-dependent questions: "In all possible futures, which solution of A will be the most satisfying for me? "This could be considered as Human approval directed behavior".

Counterfactuals. "If hypothesis B is true, what should we do for A?"

Meta: "What is the correct AI safety theory?", "Please apply the best possible AI safety theory to yourself", "Which questions are safe?"

Meta-values: "Do what I mean", "Extract my current set of values and present it to me", "Extrapolate my set of values, so they stop contradicting each other, and present it to me"

Use of super computation: The AI should create not one solution, but a tree of workable solutions. ftp://ftp.botik.ru/rented/znamensk/www/sbornik/meta-2008/meta2008_submission_8.pdf

9. Safe learning

As AI is learning, it could make detrimental mistakes. Putting the human in the loop as an advisor could prevent it finishing the mistake like described in "Trial without Error: Towards Safe Reinforcement Learning via Human Intervention" https://arxiv.org/abs/1707.05173 and Kosoy's "Delegative Inverse Reinforcement Learning (DIRL), where agent often delegates the choice of action to an advisor https://agentfoundations.org/item?id=1550

References

Argonov, V. (2015). *Rethinking progress: Beyond the horizon.* https://www.youtube.com/watch?v=pymcyprdS1s

Armstrong, S. (2017). Good and safe uses of AI Oracles. *ArXiv:1711.05541 [Cs].* http://arxiv.org/abs/1711.05541

Babcock, J., Kramar, J., & Yampolskiy, R. V. (2017). Guidelines for Artificial Intelligence Containment. *ArXiv E-Prints*, arXiv-1707. https://arxiv.org/pdf/1707.08476.pdf

Baez, S., Herrera, E., García, A. M., Manes, F., Young, L., & Ibáñez, A. (2017). Outcome-oriented moral evaluation in terrorists. *Nature Human Behaviour*, *1*, 0118.

Bostrom, N. (2014). *Superintelligence.* Oxford University Press.

Bostrom, N. (2016). *Hail Mary, Value Porosity, and Utility Diversification.* http://www.nickbostrom.com/papers/porosity.pdf

Feygin, Y. B., Morris, K., & Yampolskiy, R. V. (2018). Uploading Brain into Computer: Whom to Upload First? *ArXiv Preprint ArXiv:1811.03009.*

Gwern. (2016). *Why Tool AIs want to be Agent AIs.* https://www.gwern.net/Tool%20AI

Kent, A. (2004). A critical look at risk assessments for global catastrophes. *Risk Analysis*, *24*(1), 157–168.

Kletz, T. A. (1978). What you don't have, can't leak. *Chemistry and Industry*, *6*, 287–292.

Krakovna, V. (2020). *Specification gaming: The flip side of AI ingenuity | DeepMind.* https://deepmind.com/blog/article/Specification-gaming-the-flip-side-of-AI-ingenuity

Lem, S. (1963). *Summa technologiae.* Suhrkamp Verlag.

Orseau, L., & Armstrong, M. (2016). Safely interruptible agents. *Machine Intelligence Research Institute.*

Perrow, C. (2011). *Normal accidents: Living with high risk technologies.* Princeton University Press.

Sandberg, A., Drexler, E., & Ord, T. (2017). *Dissolving the Fermi Paradox.* Future of Humanity Institute. http://www.jodrellbank.manchester.ac.uk/media/eps/jodrell-bank-centre-for-astrophysics/news-and-events/2017/uksrn-slides/Anders-Sandberg---Dissolving-Fermi-Paradox-UKSRN.pdf

Soares, N., Fallenstein, B., Armstrong, S., & Yudkowsky, E. (2015). Corrigibility. *AAAI Workshop - Technical Report*, *WS-15-02*, 74–82.

Trazzi, M., & Yampolskiy, R. V. (2018). Building Safer AGI by introducing Artificial Stupidity. *ArXiv:1808.03644 [Cs].* http://arxiv.org/abs/1808.03644

Turchin, A. (2017a). *Human upload as AI Nanny.*

Turchin, A. (2017b). Messaging future AI. *Manuscript.* https://goo.gl/YArqki

Turchin, A. (2018). Narrow AI Nanny: Deceive Strategic Advantage via Narrow AI to Prevent Creation of the Superintelligence. *Manuscript.*

Turchin, A., & Denkenberger, D. (2017a). *Global Solutions of the AI Safety Problem* [Manuscript].

Turchin, A., & Denkenberger, D. (2017b). Levels of self-improvement. *Manuscript.*

Turchin, A., & Denkenberger, D. (2018). Wireheading as a Possible Contributor to Civilizational Decline. *Under Review in Futures.* https://philpapers.org/rec/TURWAA

Turchin, Alexey. (2018a). *AI Alignment Problem: "Human Values" don't Actually Exist.* https://philpapers.org/rec/TURAAP

Turchin, Alexey. (2018b, October 24). Possible Dangers of the Unrestricted Value Learners. *Medium.* https://becominghuman.ai/possible-dangers-of-the-unrestricted-value-learners-778d7dafd373

Yampolskiy, R. (2013). Artificial intelligence safety engineering: Why machine ethics is a wrong approach. In *Philosophy and Theory of Artificial Intelligence* (pp. 389–396). Springer Berlin Heidelberg.

Yampolskiy, R. (2014). Utility Function Security in Artificially Intelligent Agents. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)*, 1–17.

Yampolskiy, Roman, & et al. (2016). The AGI Containment Problem. *ArXiv:1604.00545 [Cs]*, *9782.* https://doi.org/10.1007/978-3-319-41649-6

Yampolskiy, Roman V. (2016). Verifier theory and unverifiability. *ArXiv Preprint ArXiv:1609.00331.*

Yampolskiy, R.V. (n.d.). Leakproofing Singularity—Artificial Intelligence Confinement Problem. *Journal of Consciousness Studies (JCS)*, *19*(1–2), 194–214.

Yudkowsky, E. (2004, May). *Coherent Extrapolated Volition.* http://intelligence.org/files/CEV.pdf

Yudkowsky, E. (2008). *Artificial Intelligence as a Positive and Negative Factor in Global Risk, in Global Catastrophic Risks* (N. B. and M.M. Cirkovic, Ed.). Oxford University Press: Oxford, UK.

Useful literature:

"Known-algorithm non-self-improving" (KANSI) is a strategic scenario and class of possibly-attainable AI designs, where the first pivotal powerful AI has been constructed out of known, human-understood algorithms and is not engaging in extensive self-modification. Its advantage might be achieved by, e.g., being run on an exceptionally large cluster of computers. If you could build an AI that was capable of cracking protein folding and building nanotechnology, by running correctly structured algorithms akin to deep learning on Google's or Amazon's computing cluster, and the builders were sufficiently paranoid/sensible to have people continuously monitoring this AI's processes and all the problems it was trying to solve and *not* having this AI engage in self-modification or self-improvement, this would fall into the KANSI class of scenarios"

https://arbital.com/p/KANSI/

Armstrong – minimal impact AI https://arxiv.org/pdf/1705.10720.pdf

No Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles https://arxiv.org/abs/1707.03501

Simulation boxing

https://www.lesswrong.com/posts/5P6sNqP7N9kSA97ao/anthropomorphic-ai-and-sandboxed-virtual-universes