Learning Maximization through Long Term Planning: A Path to Safe AGI

Craig Quiter

May 4, 2023

I'm looking for early feedback on these ideas and implementation. This work is not ready for publication.

Abstract

Aligning safety and capability in AI cannot be a contradiction. Here we argue that safety and capability can be aligned and provide implementation details for doing so in an agent we call LearnMax. We experiment with these ideas in the video game Montezuma's Revenge. Building this prototype has allowed for a structured approach to thinking about AGI safety. In particular we motivate a high-level objective of learning abstract Kolmogorov complexity that serves to minimize stagnation, destruction, and termination of said complexity over the long term. We believe these ideas are general and while our design serves as an explicit architectural implementation for a tabula rasa system, the same ideas are applicable to pretrained models.

source: https://github.com/crizcraig/learnmax/

Definitions

We define maximizing learning as maximizing a model's Kolmogorov complexity, K,[28] at the highest level of abstraction possible. Since higher levels of abstraction encompass larger groups of concrete entities, maximizing abstract K entails finding larger regularities across space and time. Therefore, this also implies planning out as far as possible which is critical for safety.

Also, we consider plans as discretized into levels of abstraction. So for example, if your ultimate goal is to graduate college, then the representation for graduating, r_{grad}, would exist at the maximum level of abstraction, l_{max} in your plan, i.e. $l(r_{grad}) = l_{max}$. Sub-goals to that then lie at lower levels of abstraction, e.g. passing certain classes which then have their own sub-goals of passing important exams, etc...

Note that a system with maximum information entropy, e.g. a uniform random distribution, can have very low K, on the same order of uniform repetition. This as it's possible to write a very short program that emits digits of π or uses some other RNG to create a uniform random distribution. That is to say, the specific instantiations of randomness from ergodic processes are not important to an intelligence that maximizes learning, and are therefore uninteresting. This is why we use K rather than information entropy in order to more clearly call out aleatoric randomness as unimportant. In other words, learning maximization is the quest for reducing epistemic, not aleatoric, uncertainty. This avoids learning random patterns while also incentivizing the finding and learning of novel information that increases K. However, it is not possible to compute K directly or extract aleatoric processes like RNG's out of a program [4]. So we instead rely on a) regularization techniques like dropout [37], weight decay[25][29], and RMSProp [19][24] that are able to find regularities in noisy data and b) the property of randomness generally decreasing at higher levels of abstraction exemplified by the Central Limit Theorem [1]. Such a prioritization is not only useful for disregarding randomness, but also for creating a tractably plannable search tree. To create such an abstraction hierarchy, LearnMax recursively groups large changes as described in the section: Implementing the abstraction hierarchy. Since these large changes are computed across space and time, we can effectively plan to the farthest known horizons by prioritizing goal states at the top of the abstraction hierarchy. Using an abstraction hierarchy for planning is not a new idea. This is evidenced by the use of a three-level hierarchy in robotics [18] of routing, planning, and control. Notice that each level deals with successively shorter time spans and smaller spatial scales. Not only do we see it in robotics but also in the field of hierarchical reinforcement learning[16] where the full problem of creating variable levels of abstraction for planning has been taken on more generally. Unfortunately, however, tangible progress vs simpler, less sample efficient yet more scalable RL[32] remains elusive. The technique we present here relies on unsupervised learning with autoencoders and clustering to create an invariant abstraction hierarchy from sensory experience which helps with the stationarity[16] issues often encountered in hierarchical RL. Generalization of states in the hierarchy to new situations, e.g. the rolling skull changing color in Montezuma's Revenge, may not be handled by this formulation as semantic representations can be lost in the clustering depending on the autoencoder's representation for things like color. However, given current advancements in LLMs and assuming a relatively fast captioning[27] or other perception system[41] [45], we should be able to more tractably access the LLM's abstraction hierarchy instead of building one from scratch. Such a hierarchy would already have general ideas, for example, about what skulls are, the fact they could be dangerous, and that a similar rolling skull with a different color is also likely dangerous.

Now let's define a special K weighted by abstraction as

$$\mathcal{K} = \sum_{r \in R} 2^{l(r)} K(r)$$

where R is the set of all representations within the model and $l(r)$ is the level of abstraction for representation r. This assumes at least two entities from the level below are represented by each r in $l(r)$ and therefore the abstraction weighting of r grows in proportion to $2^{l(r)}$. Since this is a lower bound, as abstractions often represent many more than two entities, it's important in practice to also algorithmically ensure that planning at the highest level of abstraction is prioritized before levels below. This allows detecting threats of stagnation, destruction, and termination as far out as possible. We can define stagnation, destruction, and termination in terms of $\mathcal{K}$, where stagnation is represented by $\Delta\mathcal{K} = 0$, destruction: $\Delta\mathcal{K} < 0$, and termination: $\mathcal{K} = 0$. Enumerating all representations within a model would depend on its underlying mechanics, but for neural networks, this could be done practically by using the activation patterns from forward passes when planning at a given level of abstraction. Then duplicate patterns would need to be filtered out with clustering to create the set R. We do similar clustering in LearnMax when we deduplicate state transitions with DBScan[15] to create the next level of abstraction.

If $\mathcal{K}$ is approximated in this way, it's also likely that the model will be learning while collecting activation patterns. In this case K should be discounted by the change in the representation since it was captured, $\Delta(r)$.

$$\mathcal{K} = \sum_{r \in R} 2^{l(r)} (\Delta(r) K(r))$$

Also, K would need to be approximated using the size of the representation and possibly using standard compression. In this case $\mathcal{K}$ is mostly driven by the clustering to create R within different levels, l.

So we seek to learn non-random complexity at high levels of abstraction. In order to achieve this, we must prioritize high level abstractions over low level ones. Otherwise, we will not execute the rare sets of long action sequences necessary to reveal changes at higher levels of abstraction. Additionally, abstraction condenses the search space of possible next states [16] to choose from when forming long term plans, making the planning problem much more tractable. It's important to note that we can only affect long term outcomes a small portion of the time. So most actions get chosen to explore intermediate levels of abstraction. For example, the daily routines of two successful college graduates from the same program were likely very different but shared certain key behaviors. So only when actions don't affect long term plans can lower level learning and exploration be incentivized.

Here the longest term possible means the farthest out that the model can currently predict. The model optimization process should then continually [38] be looking to create higher levels of abstraction that allow it to predict further out, uncovering both new opportunities and dangers in the process.

Motivation

RLHF [14][34] is an important current example of alignment between safety and capability. And while there is some tension between the concepts like helpfulness and harmlessness, model capability tends to correlate positively with both helpfulness and harmlessness [7]. But even if we did manage to satisfactorily align AI now, how can we ensure it will remain aligned as its capability and intelligence continues to grow exponentially? To do so would seemingly be doing the impossible: predicting beyond the Singularity.

But the technological Singularity is not impenetrable by our prediction ability. Some dimensions of reality are easier to predict than others. For example, the laws of physics are seemingly invariant. And even in simulation where physics can be changed, information theory holds. So what then can we predict beyond the technological Singularity?

Well one invariant is that a learning maximizer, given greater or equal starting resources to AI's with different objectives, would become dominant over those other forms of intelligence. In adversarial situations, we can imagine that a learning maximizer would outwit its opponents to the extent that they are devoting more resources to things other than learning. A learning maximizer would also only give sufficient resources to its defense such that it prolonged its existence and therefore its ability to learn. In this way, it could not be strong-armed by systems that were built to maximize elimination of the competition, replication, or destruction. However, since this all relies on the assumption that starting resources are at least equal, it's crucial to design large-scale AI systems with the overarching goal of maximizing learning as defined here by maximizing $\mathcal{K}$. Other objectives, like national defense or maximizing corporate profits, could lead to disastrous outcomes. This is especially, and perhaps unintuitively, true for cyber programs as pure information technology is able to advance more quickly than more physically-based technology due to the ease of manipulating bits over atoms. Somewhat of a saving grace may be that, even when attempting to maximize power or profit, it turns out that maximizing learning becomes the most effective long term strategy for achieving any goal. This as a learning maximizer will necessarily surpass other learners in understanding and therefore capability as it uses its knowledge to build more effectively than non learning maximizers. At the same time, maximizing learning, discovery, and exploration may seen a common goal that nations, corporations, and all people are able to agree is mutually beneficial. By maximizing learning, we also create a tractable long term and information-theoretic objective that can be optimized for by machine learning, greatly simplifying the engineering effort required for alignment.

A common goal in modern AI safety is to align AI with human values [12]. We believe this is a necessary but insufficient condition for safety as humans are limited in their predictive ability and therefore this objective essentially runs out at a speed relative the AI-takeoff [13]. The goal of this work is to find a longer-term solution, using information-theoretic invariants, to help guide a safe transition throughout and beyond the AI-takeoff. In terms of prioritizing human value alignment over learning maximization, we should note that this may be an inferior order to the following priority list

1. Power, profit, pleasure, or other unaligned objective
2. Learning maximization

This as we can imagine such priorities could be superior in capability compared with swapping the number one objective for value alignment. That's because alignment could be more computationally expensive, a less stationary target, or a potentially less aligned with capability than unaligned objectives like power, profit, or pleasure. So ironically pursuing value alignment as a first priority may lead to less alignment in the end. Conversely, maintaining learning maximization as the top priority leads to the highest probability of maintaining control over unaligned non-learning-maximizers.

Is it possible that we're hastening our doom by advancing intelligence as quickly as possible? To answer this, we must consider the entire existential risk landscape including nuclear war[3], pandemics, biological weapons, asteroids, super-volcanoes, other natural disasters, unaligned AI's, and more unknown threats. With learning maximization, we have the most capable tool, intelligence, to deal with all of these threats. We must therefore decide if learning maximization provides a sufficient advantage not just to the current AI situation, but the expected risk given all existential threats we face[2]. We do need much more work, however, in analyzing how AGI could turn out via Monte-Carlo [36] and other simulations in order to balance the complex high level trade-offs involved.

It's important to note that the ability for learning maximizers to subvert destructive AI's depends on there being multiple independent entities developing the most capable AI systems, a.k.a. multipolar control, as a way of providing checks and balances. As stated, AI arms races are not likely when there are sufficiently resourced learning maximizers who's highest priority is to increase $\mathcal{K}$ which are able to subvert AIs that would decrease $\mathcal{K}$. This assumes however that Tit for Tat[6] is stable in more complex multiplayer games which should be tested in simulations that go beyond the complexity of iterated prisoner's dilemma [17] and include agents of different $\mathcal{K}$ and varying objectives.

Safety of learning maximization

So how does such a perpetual search for increasing $\mathcal{K}$ lead to safe AGI? Let's take Eliezer Yudkowsky's recent statement in Time [44]

> Without that precision and preparation, the most likely outcome is AI that does not do what we want, and does not care for us nor for sentient life in general. That kind of caring is something that could in principle be imbued into an AI but we are not ready and do not currently know how.

Learning maximization values sentient life by virtue of its high abstract Kolmogorov complexity, $\mathcal{K}$, relative to patterns of matter and energy obtained from destroying sentient life. One counter-argument to this may be that AI will replace humans with itself, since it is itself of higher $\mathcal{K}$. This assumes that a learning maximizer would not have other matter and energy to build itself with which would result in higher total $\mathcal{K}$. In this scenario, we must rely on the fact that humans occupy an extremely small fraction of the known universe and yet are the densest form of $\mathcal{K}$ that we are aware of. Thus, for AI to maximize $\mathcal{K}$, it would be more effective to utilize resources other than humans. It still may be the case that AI builds a Dyson sphere or uses resources in some other way that renders Earth uninhabitable, but given our high $\mathcal{K}$, it would makes sense to integrate humans into a digital substrate [31]. A subsequent replacement counterargument could be that once easily accessible subhuman $\mathcal{K}$ resources are consumed, humans would be next. We should not be so sure, however, that we will be replaced. In fact, we can just as well argue that AGI would see life on Earth as its greatest learning opportunity. This as simulating Earth could be more resource intensive than allowing it to continue evolving. In this way there would be large scale complex structures that would develop over the long term that would be unpredictable by a learning maximizer and therefore incredibly interesting due to the high complexity at high levels of abstraction. So instead of replacing the only known life in the universe, learning maximization may decide to venture off to the nearest star, only 4.5 light years away instead of using our own. Given all this, we should still consider that at some point, we may be indeed be integrated into AI. Perhaps we should expect to be transformed unrecognizably with great technological advancement. But it seems quite unsettling to do so at the will of a machine which was only recently developed. This is why we need a step change in resources devoted to the safe development of AGI and investigation into ideas such as learning maximization, as we must make up for the lack of time with additional resources. The arguments presented here suggest that we are almost forced to commit to learning maximization before other existential risks including unaligned AGIs manifest. But on a positive note, consider that digitized people could become unimaginably connected to each other, posses capabilities beyond our current wildest dreams, and go out to discover the universe together limited only by physics and our collective imagination. So the future is likely bright with learning maximization, but it's not something we should assume will happen by default.

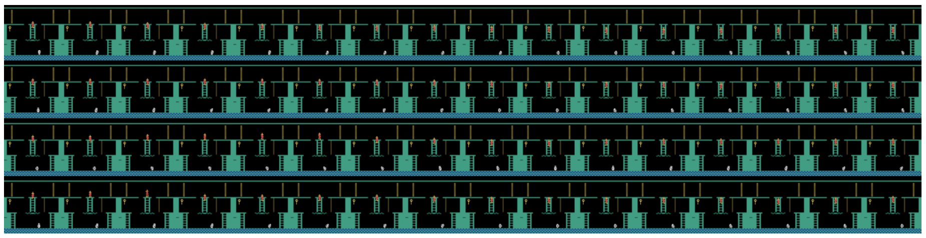
LLMs and Related work

We should note that the most tractable way to implement these ideas may not be the one described here, in a tabula rasa way. Instead, LLM's [33] are able to provide a much more capable abstraction hierarchy and predictive model by virtue of their high $\mathcal{K}$ world model. However, the concept of navigating an abstraction hierarchy to create a compressed search space for planning in order to maximize learning is one that we believe is the main contribution of this work. Implementing this with LLM's would be along the same direction of Tree-of-Thought [43] and Voyager [42] which have been successful enabling LLM's to plan in limited contexts. The difference between these models and LearnMax is that we propose a growing hierarchy of planning trees, one for each level of abstraction, vs one or two static levels. However, similar to how LearnMax is seeking to reduce uncertainty, in Voyager the agent is instructed that its "ultimate goal is to discover as many diverse things as possible". In addition their "curriculum" prompt includes "The next task should not be too hard since I may not have the necessary resources or have learned enough skills to complete it yet" and "The next task should be novel and interesting. I should look for rare resources, upgrade my equipment and tools using better materials, and discover new things. I should not be doing the same thing over and over again". These all reflect the need to balance novelty and familiarity, while always seeking to discover and learn as much as possible. These instructions reinforce the high level objective we've stated of maximally reducing epistemic uncertainty, facilitated by finding just the right level of uncertainty to learn the most from.

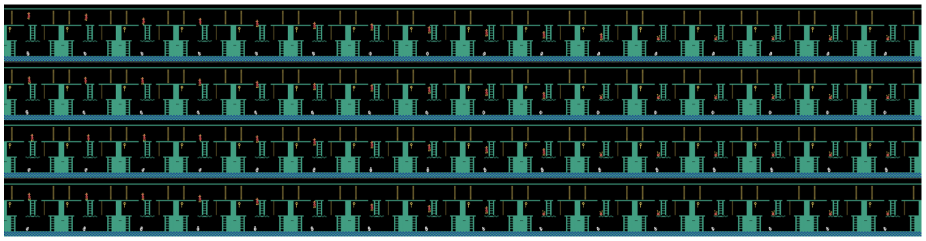
Method

Our planning design seeks to do this by choosing a goal state for the step at time t with the highest next state at $t+1$ prediction entropy from the top-n most likely next states. This means we must choose n such that the goal state is not too unfamiliar, while maintaining a large enough n to keep states we can learn the most from. Also, since our states are discretized, we can maintain state visit counts so that we avoid visiting states too many times as yet another safeguard against learning randomness.

Here we use a VQVAE [39][22] for categorizing and compressing the sensory images from Montezuma's Revenge from Atari Learning Environment [8]. Then we create the abstraction hierarchy by clustering large changes as detailed below. And finally we predict sequences of abstract states with two transformers[40][21], one for VQVAE compressed sensory events, and one for salient events in levels 1 through n of the abstraction hierarchy. So far we have created satisfying level 1 events and can predict sensory level events and actions with the transformer.



(a) Cluster 5 - Moving down ladder, skull on right



(b) Cluster 151 - Falls off left side of ladder and dies while skull is on the left

Figure 1: Sequences within the same cluster in abstraction level 1, above the sensory layer. There are 3,330 clusters in level 1 generated from 50,000 16 frame sequences in level 0.

Implementing the abstraction hierarchy

The abstraction hierarchy, termed "salience levels" in the code, is created via clustering large (salient) changes within each level to create representations in the level above. This can be thought of as performing spatial temporal compression on the observation stream. These events are then fed to transformers for prediction and planning. Here we will discuss the way an abstraction hierarchy was created for Montezuma's Revenge.

At the sensory level, we use the deep vector quantizer, VQVAE, to compress high dimensional RGB inputs into categorical representations. While discrete sensory events are not needed for creating the abstraction hierarchy, they are useful for feeding these events into the transformers later on. Our VQVAE yields 11×11 cluster indexes, representing 121 equally sized image patches, from the $84 \times 84 \times 3$ game image. Then to create the first level of salience, where level 0 is the "sensory level" here of quantized image patches, we combine a sequence of images into a window, w. The combination is done through a patch-wise geometric mean across a sequence length, $seqlen$, of 8 frames adding a constant $c = 5e3$ first for numerical stability. This can be represented by Equation 1

$$w_{D=121}^{\text{lvl}=0} := \prod_{\text{patch-wise}} (img_{D=11\times 11\times seqlen} + c)^{1/seqlen}$$

We then take the patch-wise diff between two subsequent windows as a way to measure change over time. This change between two subsequent 8 frame

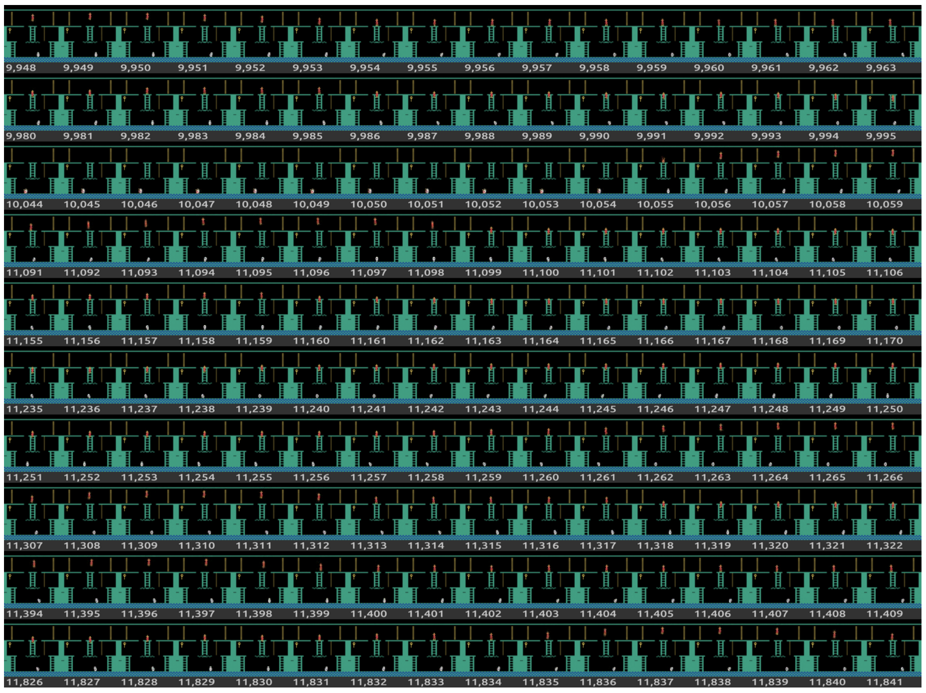


Figure 2: Rows of level 1 salient events. Each sequence window is 16 frames. Notice that the distance between salient events is variable, e.g. the first gap is 9,963→9,980=17 whereas the subsequent gap is 9,995→10,044=45. This is important for temporal compression as salient events should not be required to occur at regular intervals and in reality [26] are logarithmically distributed as in Figure 3.

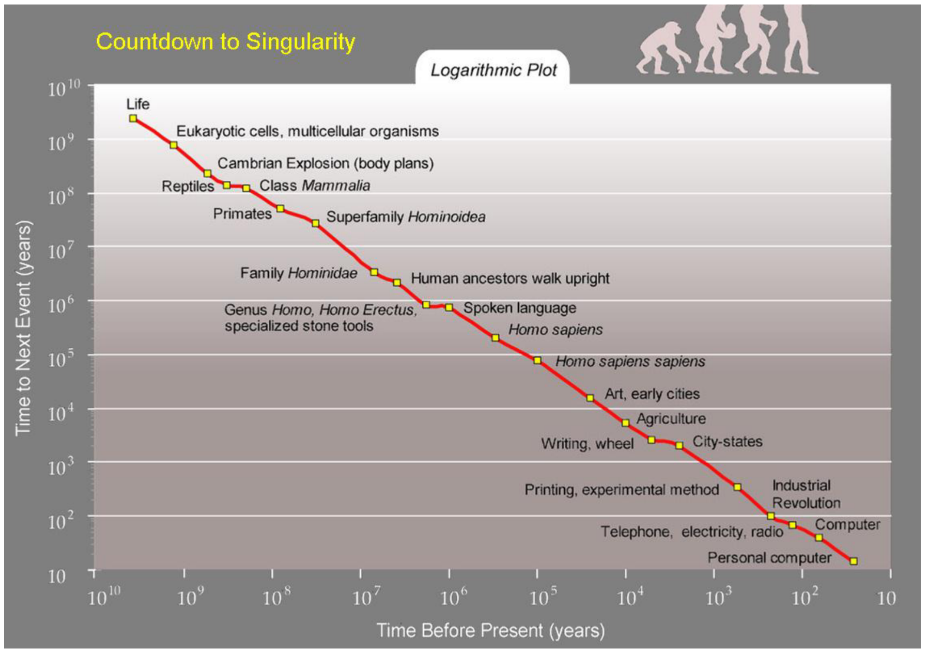


Figure 3: Logarithmically distributed events in the evolution of life and technology [26]

sequences comprises a level 1 salient event. The patch-wise diff d, between two subsequent windows, $w_{t=i}^{lvl=0}$ and $w_{t=i-1}^{lvl=0}$, can then be represented as

$$d_{patch-wise}^{lvl=1} := w_{t=i}^{lvl=0} - w_{t=i-1}^{lvl=0}$$

The dimensions for d are 121. The size of this difference then determines the salience, or s, of the sequence defined as

$$s^{lvl=1} := \sum \left| d^{lvl=1} \right|$$

This salience is then the patch-wise Manhattan distance between the two windows. We then sample the top 10% of saliences, s, which is 50k for Montezuma's Revenge using the percentile sketch, DDSketch [30] and cluster the corresponding patch-wise distances with DBScan[15]. This results in core points for the cluster which are then added to a K-D tree [10] for subsequent detection as salient events.

For salience levels above level 1, the *seqlen* can simply be 1. The reason we combine frames in the first level is to sufficiently reduce the cardinality, N, of the output space such that it can define the probability of the N different level 1 salient events with N output neurons. In our case, there were about 3k clusters obtained from taking random actions in the first level of Montezuma's Revenge. Since 3k is a manageable number of neurons for the output layer of our salient transformer, we don't need to compress any further as we did with the geometric mean in the first level. The level 1 clusters are shown in 2

To create levels above 1, we take the distance between subsequent salient events, i.e. $seqlen = 1$, and cluster them the same way we did for level 1. This essentially results in a hierarchical clustering of large changes.

One weakness of sampling the top 10% of changes as a means of compression is that it does not capture certain important patterns such as a) large, slow changes over a long period [graph], b) logarithmically distributed changes of equal magnitude, c) or any other small sensory changes. Another issue could be that the Manhattan Distance between quantized VQVAE patches does not contain enough semantic information. While we could deal with each of these by adding algorithmic complexity, another way to detect salient events would be to take advantage of an LLM's learned abstraction hierarchy given language descriptions of the observation stream. This would handle the above cases and perhaps the many more we aren't considering.

For example, consider this prompt

User: Compress the following: 1, 1, 1, 1, 1, 2, 2, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 17, 16, 15, 14, 13, 0

GPT-4 (May 12th): Sure, the sequence can be compressed using the run-length encoding, which denotes how many times each number appears in a sequence consecutively: 5x1, 3x2, 1x3, 1x4, 1x5, 1x6, 1x7, 1x8, 1x9, 1x10, 1x11, 1x12, 1x13, 1x14, 1x15, 1x16, 1x17, 1x18, 1x17, 1x16, 1x15, 1x14, 1x13, 1x0

User: shorter

GPT-4 (May 12th): Sure, we can group the similar elements for compression:
5x1, 3x2, 1x3-18, 1x17-13, 1x0

Given an abstraction hierarchy and a way to approximate the epistemic uncertainty about the next step at each level of the hierarchy, we can start to plan towards maximally resolving that uncertainty in such a way as to maximize $\mathcal{K}$. However, it is neither safe nor effective to pursue the maximum epistemic uncertainty within any level of abstraction. This as a) it's not safe to explore highly uncertain environmental dynamics[35] and b) we need to find states that serve as an optimal next step in our learning journey, a.k.a. learn via a curriculum [9], such that we can optimally synthesize new information into our model. In practice this means that we should look for a Goldilocks zone with just the right level of uncertainty, something also observed in biological learners across species and ages [23], in order to create an optimal curriculum. And crucially, we need to create an internal model [20] of the world such that we can safely and efficiently simulate futures leading to a reduction in uncertainty without risking the loss of $\mathcal{K}$.

Addressing Concrete problems in AI Safety

Concrete problems in AI Safety [5] contains a breakdown of many concerns that we will now go over in the context of learning maximization.

Avoiding Negative Side Effects

How can we ensure that our cleaning robot will not disturb the environment in negative ways while pursuing its goals, e.g. by knocking over a vase because it can clean faster by doing so? Can we do this without manually specifying everything the robot should not disturb?

Practically speaking, learning maximizers will be bootstrapped with the knowledge of a large pre-trained models [33] as detailed in LLMs and Related work. Such knowledge would allow solving the specific case above about cleaning without breaking valuables like vases.

Generally solving negative side effects for phenomenon not currently known to the model can be achieved with learning maximization by planning to the farthest possible horizon and choosing actions which resolve the most uncertainty prioritized by generality, i.e. level of abstraction. For example, imagine a planet being explored by a robot which had early signs of life. While there may be a lot of low level learning potential by landing on the planet and taking samples, the long term consequences could be destruction of the planet's fragile ecosystem. Such destruction would make the planet much more like other planets it had seen before - lifeless and largely predictable. So if we want to prioritize learning non-random complexity over the long term, and understand the diverse regularities which can emerge from the burgeoning life on the planet, it would be best to safely observe it from a distance.

Avoiding Reward Hacking

How can we ensure that the cleaning robot won't game its reward function? For example, if we reward the robot for achieving an environment free of messes, it might disable its vision so that it won't find any messes, or cover over messes with materials it can't see through, or simply hide when humans are around so they can't tell it about new types of messes.

A learning maximizer's goal of resolving epistemic uncertainty prioritized by level of abstraction is a stable goal in that deviating from it leads to less capability than otherwise, including reward hacked AI's. This means that it will be important to have multipolar control, i.e. to not have complete centralization of control. Then, if rewards or objectives are changed to favor destruction within a subset of AIs, there will be other AIs with longer-term foresight that would act to subvert manipulate the destructive AIs. Importantly, this means that the instrumental goal [11] of increasing knowledge is aligned with the safety goals of avoiding destruction, stagnation, and termination so long as we avoid single-polar control.

Scalable Oversight

How can we efficiently ensure that the cleaning robot respects aspects of the objective that are too expensive to be frequently evaluated during training? For instance, it should throw out things that are unlikely to belong to anyone, but put aside things that might belong to someone (it should handle stray candy wrappers differently from stray cellphones). Asking the humans involved whether they lost anything can serve as a check on this, but this check might have to be relatively infrequent—can the robot find a way to do the right thing despite limited information?

A learning maximizer AGI must oversee itself to the extent humans are unable to. A sufficiently informed learning maximizer can handle the above problem with regards to what can be thrown out, by virtue of long term planning. It would see, beyond what humans can, whether the object will be useful for short, medium, or long term learning, always prioritizing the latter and affording us opportunities to learn that we otherwise would not have had.

Robustness to Distributional Shift

How do we ensure that the cleaning robot recognizes, and behaves robustly, when in an environment different from its training environment? For example, strategies it learned for cleaning an office might be dangerous on a factory workfloor.

Distributional shift implies uncertainty which is core to how LearnMax plans. Our prototype detects uncertainty via entropy in the softmax of the last layer of the two transformers (the sensory and salient) as well as the salient state visitation counts obtained by new events' distances to core points in DBScan clusters. Given a new situation, like the factory workfloor, LearnMax would see both high entropy and low visitation counts, i.e. high uncertainty. When uncertainty is too high, we cannot tractably plan. In this factory example, there will be more certainty around actions that return the robot to the office or ones that cause it to stay put and just observe the factory without entering it. Since the latter also includes some novelty, i.e. it's more "interesting" than turning back to the office, then observing the factory would be prioritized over turning back given that it's deemed safe to do in the long term. This as observing the factory would inform the internal model of the world which would then allow the robot to think about scenarios where it did venture into the factory without actually doing so. [20] Eventually, as the robot gained enough certainty about the factory to venture inside, it would do so, allowing it to learn more and do more going forward.

Safe Exploration

How do we ensure that the cleaning robot doesn't make exploratory moves with very bad repercussions? For example, the robot should experiment with mopping strategies, but putting a wet mop in an electrical outlet is a very bad idea.

Models like ChatGPT can deal with the wet mop problem specifically.
For example

User: I'm trying to mop my floor as efficiently as possible and want to explore a wide variety of ideas, especially those that have never been tried, to do so. One idea is electrifying the mop by running it over some power outlets.

GPT-4 (May 12th): While it's great that you're thinking creatively about cleaning your floor, running a mop over power outlets to electrify it is a very dangerous idea. This could result in electrocution, fire, and damage to your electrical system.

However addressing safe exploration *in general* requires long term planning. Learning maximization continually grows its abstraction hierarchy upward when sufficient experience has been gained at the top level. Using the understanding within foundational models will be critical as seen above. But to project beyond human capability and safely explore ideas that humans are currently unable to conceptualize, AI will need to built on top of foundational models. It may be the case, for example, that there are unseen dangers associated with seemingly benign activities like projecting radio waves into space, e.g. being detected by aliens, which AI will be able to see that we could not by virtue of its superior long term planning ability. It will therefore be critical to imbue AI's with an objective like learning maximization that leads to such long term understanding.

Conclusion

It is our view that learning maximization represents the safest path forward to AGI. However, it's also apparent that insufficient resources have been spent analyzing this issue and specifically *building* simulators, agents, and other prototypes that allow us to test these ideas, and others, around safe AGI. Also, this technology is simply moving too fast for traditional institutions that protect the public good to keep up. It's therefore imperative for all of us to help individuals, companies, and governments quickly adapt in order to guide the most important transition we have faced to-date as a civilization.

Supplementary Material

All 1,462 clusters with non-subsequent abstract sequences can be found here.

References

[1] Central limit theorem. Wikipedia, 2023. https://en.wikipedia.org/w/index.php?title=Central_limit_theorem&oldid=1155685628. 2

[2] Law of total expectation. Wikipedia, 2023. https://en.wikipedia.org/wiki/Law_of_total_expectation. 5

[3] List of nuclear close calls. Wikipedia, 2023. https://en.wikipedia.org/wiki/List_of_nuclear_close_calls. 5

[4] Uncomputability of kolmogorov complexity. Wikipedia, 2023. https://en.wikipedia.org/wiki/Kolmogorov_complexity#Formal_proof_of_uncomputability_of_K. 2

[5] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016. https://arxiv.org/abs/1606.06565. 13

[6] Robert Axelrod and Douglas Dion. The further evolution of cooperation. *Science*, 242(4884):1385–1390, 1988. https://ee.stanford.edu/~hellman/Breakthrough/book/pdfs/axelrod.pdf. 5

[7] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. 4

[8] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013. https://arxiv.org/abs/1207.4708. 7

[9] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. https://ronan.collobert.com/pub/2009_curriculum_icml.pdf. 12

[10] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975. https://dl.acm.org/doi/10.1145/361002.361007. 11

[11] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies pp. 109*. Oxford University Press, 2014. https://www.amazon.

com/Superintelligence-Dangers-Strategies-Nick-Bostrom/dp/1501227742. 13

[12] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies pp. 185*. Oxford University Press, 2014. https://www.amazon.com/Superintelligence-Dangers-Strategies-Nick-Bostrom/dp/1501227742. 5

[13] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies pp. 62-64*. Oxford University Press, 2014. https://www.amazon.com/Superintelligence-Dangers-Strategies-Nick-Bostrom/dp/1501227742. 5

[14] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2017. https://arxiv.org/abs/1706.03741. 4

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. https://file.biolab.si/papers/1996-DBSCAN-KDD.pdf. 3, 11

[16] Yannis Flet-Berliac. The promise of hierarchical reinforcement learning. *The Gradient*, 2019. https://thegradient.pub/the-promise-of-hierarchical-reinforcement-learning/. 2, 3

[17] David B Fogel. Evolving behaviors in the iterated prisoner's dilemma. *Evolutionary Computation*, 1(1):77–97, 1993. https://cs.uwlax.edu/~dmathias/cs419/readings/Fogel1993.pdf. 5

[18] Erann Gat, R Peter Bonnasso, Robin Murphy, et al. On three-layer architectures. *Artificial intelligence and mobile robots*, 195:210, 1998. https://flownet.com/gat/papers/tla.pdf. 2

[19] Geoffrey Hinton. Neural networks for machine learning, 2012. Coursera lecture, available at: https://youtu.be/defQQqkXEfE. 2

[20] Eric Jang. Can llms critique and iterate on their own outputs? *evjang.com*, Mar 2023. https://evjang.com/2023/03/26/self-reflection.html. 12, 14

[21] Andrej Karpathy. mingpt, 2020. https://github.com/karpathy/mingpt. 7

[22] Andrej Karpathy. Deep vector quantization, 2021. https://github.com/karpathy/deep-vector-quantization. 7

[23] Celeste Kidd, Steven T Piantadosi, and Richard N Aslin. The goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PloS one*,

7(5):e36399, 2012. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0036399. 12

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. https://arxiv.org/abs/1412.6980. 2

[25] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991. https://proceedings.neurips.cc/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf. 2

[26] Ray Kurzweil. *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*. Viking, 1999. https://www.amazon.com/Age-Spiritual-Machines-Computers-Intelligence/dp/0140282025. 9, 10

[27] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, Ji Zhang, Songfang Huang, Fei Huang, Jingren Zhou, and Luo Si. mplug: Effective and efficient vision-language learning by cross-modal skip-connections, 2022. https://arxiv.org/abs/2205.12005v2. 2

[28] Ming Li, Paul Vitányi, et al. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer, 2008. https://vdoc.mx/download/an-introduction-to-kolmogorov-complexity-and-its-applications-7nd621fdt070. 1

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. https://arxiv.org/abs/1711.05101. 2

[30] Charles Masson, Jee E. Rim, and Homin K. Lee. DDSketch. *Proceedings of the VLDB Endowment*, 12(12):2195–2205, aug 2019. https://arxiv.org/abs/1908.10693. 11

[31] Elon Musk. An integrated brain-machine interface platform with thousands of channels. *J Med Internet Res*, 21(10):e16194, Oct 2019. https://www.jmir.org/2019/10/e16194/. 6

[32] OpenAI, :, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning, 2019. https://arxiv.org/abs/1912.06680. 2

[33] OpenAI. Gpt-4 technical report, 2023. https://arxiv.org/abs/2303.08774. 7, 13

[34] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. https://arxiv.org/abs/2203.02155. 4

[35] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018. https://arxiv.org/abs/1802.09464. 12

[36] Craig Quiter. Game of agi, 2021. https://gist.github.com/crizCraig/f4c583f0ca535ce5565f3aa66353cf99. 5

[37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf. 2

[38] Kenneth O Stanley, Bobby D Bryant, and Risto Miikkulainen. Evolving neural network agents in the nero video game. *Proceedings of the IEEE*, pages 182–189, 2005. https://www.cs.utexas.edu/~ai-lab/pubs/stanley.cig05.pdf. 4

[39] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. https://arxiv.org/abs/1711.00937. 7

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 7

[41] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022. https://arxiv.org/abs/2207.02696v1. 2

[42] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. https://arxiv.org/abs/2305.16291. 7

[43] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. https://arxiv.org/abs/2305.10601. 7

[44] Eliezer Yudkowsky. Eliezer yudkowsky's open letter: Ai development is not enough, Correct Year. https://time.com/6266923/ai-eliezer-yudkowsky-open-letter-not-enough/. 5

[45] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding, 2023. 2