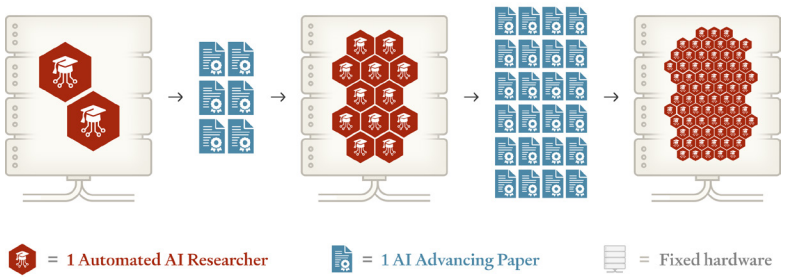
Will AI R&D Automation cause a Software Intelligence Explosion?

Empirical evidence suggests that, if AI automates AI research, feedback loops could overcome diminishing returns, significantly accelerating AI progress

Daniel Eth & Tom Davidson

March 2025



Summary

AI companies are increasingly using AI systems to accelerate AI research and development. These systems assist with tasks like writing code, analyzing research papers, and generating training data. While current systems struggle with longer and less well-defined tasks, future systems may be able to independently handle the entire AI development cycle – from formulating research questions and designing experiments, to implementing, testing, and refining new AI systems.

Some analysts have argued that such systems, which we call *AI Systems for AI R&D Automation* (ASARA), would represent a critical threshold in AI development. The hypothesis is that ASARA would trigger a runaway feedback loop: ASARA would quickly develop more advanced AI, which would itself develop even more advanced AI, resulting in extremely fast AI progress – an "intelligence explosion."

Skeptics of an intelligence explosion often focus on hardware limitations – would AI systems be able to build better computer chips fast enough to drive such rapid progress? However, there's another possibility: AI systems could become dramatically more capable just by finding software improvements that significantly boost performance on existing hardware. This could happen through improvements in neural network architectures, AI training methods, data, scaffolding around AI systems, and so on. We call this scenario a *software intelligence explosion* (SIE). This type of advancement could be especially rapid, since it wouldn't be limited by physical manufacturing constraints. Such a rapid advancement could outpace society's capacity to prepare and adapt.

In this report, we examine whether ASARA would lead to an SIE. First, we argue that shortly after ASARA is developed, it will be possible to run orders of magnitude more automated AI researchers than the current number of leading human AI researchers. As a result, the pace of AI progress will be much faster than it is today.

Second, we use a simple economic model of technological progress to analyze whether AI progress would accelerate even further. Our analysis focuses primarily on two countervailing forces. Pushing towards an SIE is the positive feedback loop from increasingly powerful AI systems performing AI R&D. On the other hand, improvements to AI software face diminishing returns from lower hanging fruit being picked first – a force that pushes against an SIE.

To calibrate our model, we turn to empirical data on (a) the rate of recent AI software progress (by drawing on evidence from multiple domains in machine learning and computer science) and (b) the growing research efforts needed to sustain this progress (proxied by the number of human researchers

in the field). We find that (a) likely outstrips (b) – i.e., AI software is improving at a rate that likely outpaces the growth rate of research effort needed to achieve these software improvements. In our model, this finding implies that the positive feedback loop of AI improving AI software is powerful enough to overcome diminishing returns to research effort, causing AI progress to accelerate further and resulting in an SIE.

If such an SIE occurs, the first AI systems capable of fully automating AI development could potentially create dramatically more advanced AI systems within months, *even with fixed computing power*.

We examine two major obstacles that could prevent an SIE: (1) the fixed amount of computing power limits how many AI experiments can be run in parallel, and (2) training each new generation of AI system could take months. While these bottlenecks will slow AI progress, we find that plausible workarounds exist which may allow for an SIE nonetheless. For example, algorithmic improvements have historically increased the efficiency of AI experiments and training runs, suggesting that training runs and experiments could be progressively sped up, enabling AI progress to continually accelerate despite these obstacles.

Finally, because such a dramatic acceleration in AI progress would exacerbate risks from AI, we discuss potential mitigations. These mitigations include monitoring for early signs of an SIE and implementing robust technical safeguards before automating AI R&D.

Key Points

- Even if hardware were held constant upon the creation of AI systems capable of fully automating AI R&D, software progress alone could plausibly enable faster and faster AI advancements, yielding a "software intelligence explosion."
- If a software intelligence explosion were to occur, it could lead to incredibly fast AI progress, necessitating the development and implementation of strong policy and technical guardrails in advance.

Introduction

Over the past few years, AI systems have increasingly been used by AI researchers to help conduct further AI R&D. Recent evidence suggests cutting-edge AI systems can now exceed human expert performance on some AI R&D tasks when given short (2-hour) time windows, though humans demonstrate better performance with increasing time. Many researchers expect that in the coming years or decades, further advancements will lead to AI systems capable of fully automating all tasks involved in AI R&D. Systems of this sort, which we'll refer to as *AI Systems for AI R&D Automation*, or *ASARA*, can be thought of as being able to substitute for any remote R&D workers at companies advancing the state of the art for AI.

Some researchers argue the emergence of ASARA would trigger a feedback loop in which ASARA systems performing AI R&D lead to more capable ASARA systems, which in turn conduct even better AI R&D, and so on, culminating in an "intelligence explosion" – a period of very rapid and accelerating AI progress which results in significantly superhuman AI.[1] Others, however, are skeptical that progress in AI would become extremely fast after the creation of ASARA, because they expect progress will become bottlenecked on physical processes in hardware R&D or hardware production.

In this report, we examine the possibility of an intelligence explosion occurring with a constant amount of computer hardware, with the explosive feedback loop sustained simply through software progress. If this software-only feedback loop similarly led to accelerating AI progress after the creation of ASARA, then we would have what we'll call a *software intelligence explosion*, or *SIE*.[2] While sufficiently advanced AI, if managed well, would offer tremendous benefits to society, an SIE in particular would be worrying, for a couple reasons:

1. If an SIE is possible, then hardware constraints are irrelevant for AI becoming incredibly powerful very quickly. Extremely dangerous AI capabilities may emerge very suddenly, before the world is ready to handle them. Technical guardrails and governance mechanisms that were initially adequate for the level of technology could quickly become insufficient. Note that this analysis does NOT depend on any discontinuous jumps in AI capabilities, but instead is based on "business as usual" improvements in AI software feeding back into itself.

1 Ideas of this sort date back to at least mathematician I. J. Good, who in a 1965 paper predicted that after humanity invents an "ultraintelligent machine," there would then "unquestionably be an 'intelligence explosion,' and the intelligence of man would be left far behind." As another example, Ray Kurzweil's popular 2005 book The Singularity is Near argued at length that the world would see an intelligence explosion within decades.

2 The idea of an SIE was previously discussed in Tom Davidson's report "What a Compute-Centric Framework Says About Takeoff Speeds." Note that report uses the term "software-only singularity" for the same concept.

2. Most policy proposals for AI governance are based on using computer hardware as the focus of regulation. For instance, proposals include tracking stocks and flows of hardware to monitor potential AI capabilities available to different actors, and requiring safety evaluations for AI systems trained above certain computational thresholds. These proposals are a valuable start and would offer a useful foundation for further policies. But in an SIE, they may need to be extended. Rapid software progress would quickly make computing-power-centric metrics outdated for the leading AI developers. Tracking AI capabilities in an SIE would require ongoing monitoring of companies' R&D progress and evaluation of their most powerful internally deployed systems.

In the rest of this piece, we'll explain the dynamics that may lead to an SIE, as well as reasons for and against thinking an SIE will occur. We'll end with a few preliminary policy suggestions for how society could manage the risks associated with an SIE. But first, a few caveats:

- In reality, hardware likely won't be held constant upon the creation of ASARA. Therefore, once we reach ASARA, an intelligence explosion may be even more likely than what's implied by the below discussion.
- Prediction is hard, especially about the future. Relatedly, we make many simplifying assumptions, such as assuming certain variables will grow continuously when in reality they should grow in small-yet-incremental steps. We may or may not turn out to be correct in broad strokes, but we will definitely be wrong in all sorts of small and subtle ways.
- Prediction gets even harder when there are major changes to the world, and the existence of ASARA (and of powerful AI systems on the path to ASARA) may impact the world in many unexpected ways, possibly throwing a wrench in any particular prediction.
- Except where indicated otherwise, the discussion below looks at what will happen "by default" – i.e., if business-as-usual improvements in AI continue, with AI companies independently pursuing their perceived local, short-term self-interests, and if social factors don't slow down the pace of AI progress. But it's possible that powerful stakeholders will instead coordinate to avoid undesirable outcomes.[3] Indeed, one of our main motivations for writing this piece is to alert people about the path we're potentially headed down, with the hope that we can change course if needed.

Let's now turn our attention to the dynamics that may lead to an SIE.

3 Alternatively, poorly coordinated actions may shift us away from pursuing this course for other reasons.

Where AI progress comes from

In order to predict how AI progress will change after the creation of ASARA, we first need to understand where AI progress originates. As a general rule, AI progress comes from researchers doing one of two things:

1. **Using more computational power (and more data).** Simply using more computational power with the same algorithms and similar types of data can lead to better AI systems.[4] GPT-3, for instance, is basically just a scaled-up version of GPT-2,[5] yet this increased computational power enables GPT-3 to not only engage in coherent dialogue, but also write functional computer code, translate between languages, and create poetry; GPT-2, meanwhile, largely just babbles as if discombobulated.[6] Notably, increasing the computational power of cutting-edge AI systems doesn't just lead to improved performance on the same tasks, but can also lead to the emergence of new capabilities.

   Researchers have two methods for increasing the amount of computing power for AI systems: AI developers can spend more money for more computational power, and hardware researchers can create better forms of hardware that provide more computational power for the same cost.

2. **Developing better AI "software."** This category includes basically everything other than raw computer hardware. Most obviously, it includes overarching ideas for new AI paradigms or techniques, such as the general principle behind deep learning (train very deep neural networks with large amounts of data) or the more specific techniques used for training large

4 While this paragraph focuses on how more computational power enables better AI systems within the current AI paradigm of deep learning, it's notable that in previous AI paradigms, more computational power also enabled better AI systems. For instance, in 1997, the chess-playing AI system Deep Blue bested world chess champion Garry Kasparov, using a technique known as "tree search." In tree search, the AI system plots out potential moves and countermoves, and this technique is improved by increased computing power, as more computing power lets the system consider deeper sequences of moves, effectively "thinking for longer."

5 Note that the description of GPT-3 as a scaled-up version of GPT-2 is somewhat of a simplification. First, there are various tweaks done to GPT-3 that were not done to GPT-2. Second, while GPT-3 was trained on the same general *kind* of data as GPT-2, it was trained on much more of this data; that is, GPT-3 was scaled up not only in terms of computing power but also in terms of training data. And third, the comparison here is between the "base models" of GPT-3 and GPT-2, which were both trained simply to imitate text; this is in contrast to any version of GPT found on ChatGPT, which involves a system that has been modified from the base model to perform the role of a question-answering chatbot.

6 As another example of increased computing power yielding improved AI abilities, OpenAI's recent system o1, which specializes in reasoning, performs better when enabled to utilize more computational resources to solve problems (thereby engaging in reasoning for longer).

language models, also known as LLMs (train an AI system to predict internet text, word by word,[7] with the goal of learning to imitate human-like speech).

AI "software" also includes many specific details of AI systems, such as: the "architecture" or structure of the AI system, the algorithm used to train the system to begin with, techniques used to procure high-quality data, efficiency tweaks, hyperparameters, and so on. In the context of LLMs, software would further include techniques for prompting and fine-tuning the LLM to give desirable output, as well as added "scaffolding" around an LLM which can allow the system to solve problems the LLM would be unable to solve on its own (such as by chaining together multiple LLMs into an AI agent or enabling an AI to use tools such as a web browser).

Putting together the above points, we can illustrate the main factors leading to AI progress as follows:

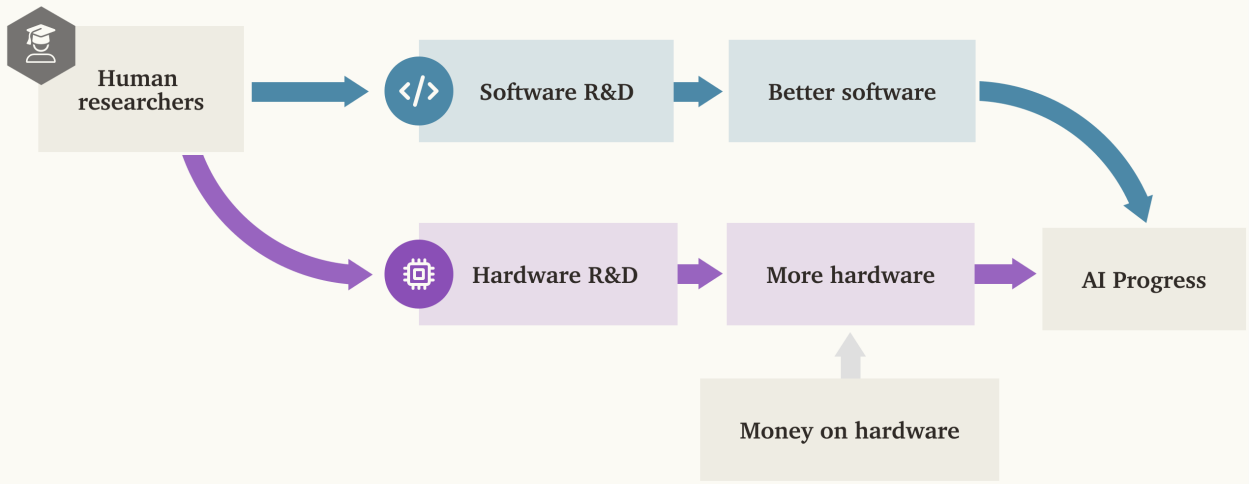


***Figure 1**: Simplified diagram showing the main factors leading to AI progress. These factors are increases in hardware (due to either more money being spent on hardware or hardware R&D leading to improved hardware) and improvements in software (due to software R&D). Today, both hardware R&D and software R&D are performed by human researchers.*

In an SIE, meanwhile, human researchers would be replaced by ASARA systems as the graph becomes a loop. Given the definition of an SIE, hardware would also be held constant; this assumption allows us to examine the potential for very rapid AI progress, unencumbered by physical bottlenecks related to hardware improvements.

7 Technically, this description of training LLMs is a simplification, for two reasons. First, while LLMs can be thought of as being trained to predict text "word by word," they are actually typically trained to predict text "token by token," where a "token" could be a word, but also could be a part of a word or another character (such as a comma or an emoji). Second, while this description applies to the "base model" of LLMs, it neglects that LLMs are often modified with further training to exhibit specific types of behavior.

The resulting scenario would look like this:

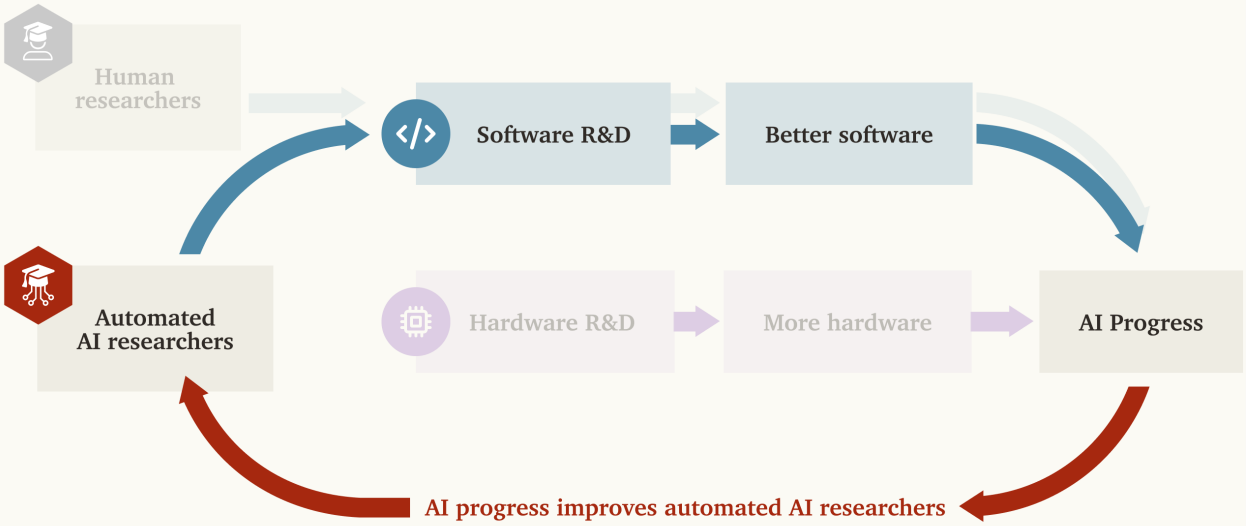


Figure 2: *Diagram similar to Figure 1, but modified in two ways for an SIE. First, the hardware path is nixed, allowing for examining a scenario where only software improves. Second, ASARA systems replace human researchers in performing R&D, yielding faster AI progress, which further enhances ASARA systems' abilities, enabling them to conduct even better and/or faster software R&D.*

But before we consider what happens after the creation of ASARA, let's turn our attention to the strength of the process whereby human researchers make AI progress through software R&D.

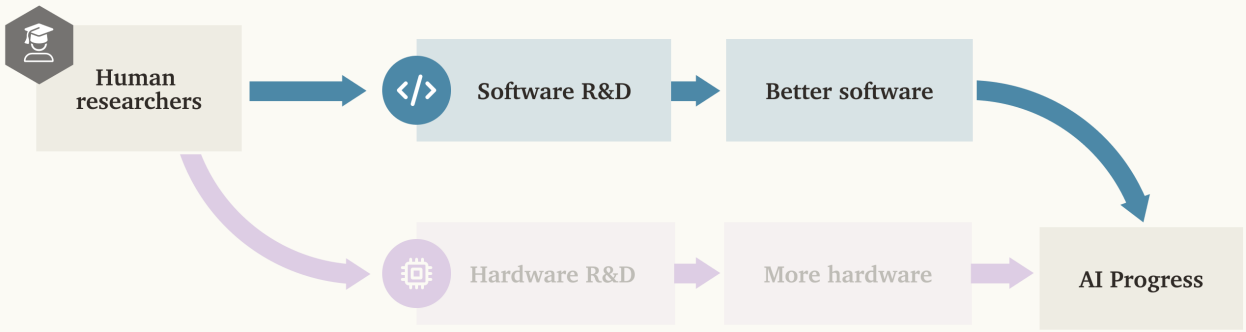


Figure 3: *Diagram representing the current path by which software R&D yields AI progress.*

Improvements in AI software are already driving fast AI progress

Historically, there's been a lot of attention on how AI systems have become bigger and have been using more computing power. But software improvements are also responsible for much AI progress.

As we saw in the above section, software progress encompasses many different avenues (from new architectures, to new and better forms of training data, new or improved "learning algorithms" that guide the AI system through training, new scaffolding or other methods of integrating the trained AI system within some broader system, etcetera). This software progress can be further broken down into "efficiency improvements" (i.e., when new AI systems do approximately the same thing as previous AI systems, but require less computing power) and "capability improvements" (i.e., when new AI systems do things that previous systems didn't do at all, or when newer systems do things more competently than previous systems).

In reality, the distinction between efficiency improvements and capability improvements is at times fuzzy. For instance, improvements in the efficiency of *training* AI systems can generally be converted into capability improvements. With greater training efficiency, you can train a larger AI system, and larger systems tend to have new capabilities and better performance. Regardless, the two types of improvements are still conceptually distinct, even if in reality they're often blurred. And there are other types of improvements that more clearly fit into one bucket or the other – such as improvements in capabilities which don't derive from improvements in efficiency.

So how fast is AI software progress now?

That's a really tough question to answer, because AI software progress is really hard to measure. It's particularly difficult to measure capability improvements; for instance, ChatGPT was created by modifying GPT-3.5 to engage in informative dialogue and putting this modified version within an intuitive user interface. How are we supposed to measure the amount of progress represented by the fact that ChatGPT is more likely to productively converse with the user than the original GPT-3.5, or the amount represented by the good user interface?

While capability improvements are particularly difficult to quantify, we can better measure efficiency improvements. One way to measure efficiency improvements is to look at the amount of computing power needed for an AI system to exhibit a particular level of performance, and consider how much more computing power was previously needed for AI systems to reach the same level of performance.

By tracking the change over time, we can chart how efficiency has improved over time, at least for the capability that's investigated. And by considering several lines of evidence simultaneously, we can hopefully get a decent measure for the speed of AI efficiency progress in general. Being concrete, let's consider:

- **Image recognition.** OpenAI has found that, between 2012 and 2017, state-of-the-art image recognition algorithms became much more efficient, requiring 1/18th as much computing power to run in order to achieve consistent results. This growth rate corresponds to the runtime efficiency doubling every **15 months** on average – i.e., a 15-month "efficiency doubling time." Similarly, they found that, between 2012 and 2019, the amount of computing power needed to *train* these state-of-the-art image recognition systems (to the same level of performance) fell by 44x, corresponding to a training efficiency doubling time of **16 months**.

  As another data point, the research group Epoch has estimated that, from 2012 – 2022, training efficiency of image recognition algorithms had a shorter doubling time of only **9 months**.

- **Language translation and game playing.** OpenAI has found even faster progress in the efficiency of training AI systems for language translation and game playing. For language translation, based on two analyses, they calculated an efficiency doubling time of **4 months** and **6 months**, and for game playing, they found an efficiency doubling time of **4 months** for Go and **25 *days*** for Dota. Note, however, that these analyses were rougher and covered shorter periods of time than their image recognition analysis.

  A separate analysis on changes in the data efficiency of training AI systems to a specific level of performance on Atari games found an efficiency doubling time of **between 10 months and 18 months**, depending on the specified level of performance. Though note that this analysis was concerned with a separate type of efficiency than what we're focused on.[8]

- **Large language models.** Analysis from Epoch estimates that, from 2012 to 2023, training efficiency for language models has doubled approximately every **8 months** (though with high uncertainty – their 95% confidence interval for the doubling time was 5 months to 14

8 Specifically, we're focused on *computational* efficiency – when AI systems require less computational resources to perform a task. This study instead focused on *data* efficiency – when AI systems require less pieces of data in training. While these two types of efficiency would not be expected to be a perfect match, they would be expected to be related, with greater data efficiency leading to greater computational efficiency. In the training phase, computational resources must be spent on training the system on each piece of training data, so reducing the needed amount of training data (such as by increasing the data efficiency) would reduce the amount of computational resources needed for training.

months). Efficiency improvements in *running* these LLMs (instead of for training them) would be expected to grow at a roughly similar rate.[9]

The analyses so far just look at improvements for unmodified "base models" and therefore neglect efficiency benefits from improvements in "post-training enhancements" like fine-tuning, prompting, and scaffolding. These neglected benefits from post-training enhancements can be substantial. A separate analysis finds that individual innovations in post-training enhancements for LLMs **often give >5x efficiency improvements** in particular domains (and occasionally give ~100x efficiency improvements). In other words, AI models that incorporate a given innovation can often outperform models trained with 5x the computational resources but without the innovation.

And a separate informal analysis finds that for LLMs of equivalent performance, the cost efficiency of running the LLM (i.e., amount of tokens read or generated per dollar) has doubled around every **3.6 months** since November 2021. (Though note that cost efficiency doesn't just take into account software improvements, but also decreases in hardware costs and in profit margins; with that said, software improvements are probably responsible for the great majority of the cost efficiency improvements.)

- **Algorithms writ large (not just AI).** An analysis of algorithms writ large (not just AI) provides an outside perspective, which we can use to both check how surprised we should be about the above results, and how quickly we might expect AI software to progress if we have a future paradigm shift to a very different type of AI.

  That analysis found that different classes of algorithms have seen very different rates of efficiency improvements over the previous several decades – when considering using these algorithms for solving problems with very large datasets[10] (as is common in AI), **close to**

9 We should first note that many types of efficiency improvements will affect both training efficiency and runtime efficiency similarly. On the other hand, efficiency improvements that enable training smaller models (i.e. achieving the same performance with fewer parameters) may tend to impact training efficiency more than runtime efficiency. If we assume all efficiency improvements are of this type, then we'd expect training efficiency to grow twice as quickly as runtime efficiency. This conclusion is implied by a landmark 2022 paper from DeepMind, often informally referred to as the "Chinchilla paper." The Chinchilla paper found that when training an LLM with a fixed computational budget, it's best to scale the size of the model and the amount of data used to train the model equally; therefore, if a model can be made Z times smaller, the runtime efficiency will grow by Z, while the training efficiency would grow by Z^2 (since training on each data point will require $1/Z$ as many computations, and you will only need to train on $1/Z$ as many data points as well).

On the other hand, certain other types of algorithmic improvements, such as distillation or quantization, target runtime efficiency, specifically. As there are many types of improvements that hit both runtime efficiency and training efficiency similarly, some that hit one stronger than the other, and some that hit the other stronger, overall we may expect the two types of efficiency to grow at roughly similar rates.

10 In particular, solving problems where N = 1 billion.

half of the algorithm classes saw basically no efficiency improvements, around a quarter saw average efficiency doubling times of around 1 to 3 years, and around a quarter saw average efficiency doubling times of under a year.

	Image recognition	Language translation	Game playing	Large language models	Algorithms writ large (with large datasets)
Result #1 Doubling time	15 months (runtime efficiency)	4 months (training efficiency)	4 months (Go, training efficiency)	8 months (base models, training efficiency)	Basically no gains (close to half of algorithm classes)
Result #2 Doubling time	16 months (training efficiency)	6 months (training efficiency)	25 days (Dota, training efficiency)	3.6 months (runtime cost efficiency)	~1-3 years (around a quarter of algorithm classes)
Result #3 Doubling time	9 months (training efficiency)		10-18 months (Atari games, training data efficiency)		<1 year (around a quarter of algorithm classes)

Table 1: *Summary of results from various studies investigating the efficiency doubling times of AI systems in several different domains. Note that most of these studies investigated training efficiency (how much computing resources are needed to train an AI system to a specific level of capabilities) instead of runtime efficiency (how much computing power is needed to run the resultant system).*

Of the different categories above, we put the most weight on the large language model results, because LLMs are more likely to form the basis for ASARA than the other algorithms, and because the LLM training efficiency analysis uses significantly more data than the other analyses of AI algorithms. Conveniently, the LLM analyses also yield the median growth rate of the five categories – roughly speaking, LLMs saw faster efficiency growth than image recognition systems and algorithms writ large, but slower growth than language translation and game playing.

Looking at the LLM analyses, we may consider that the training efficiency estimate is likely conservative at an ~8 month doubling time, as it does not account for post-training enhancements. The runtime efficiency estimate of ~4 months, meanwhile, is likely aggressive, as it includes cost savings outside of software (such as from hardware and market forces); though even this latter estimate ignores some post-training enhancements. We believe it's reasonable to split the difference

between these two estimates and conclude that both training efficiency and runtime efficiency have a **~6 month doubling time**.[11]

We should also note that, besides efficiency improvements, capability improvements have been substantial, and may be an even larger factor than efficiency improvements. Consider:

- Recently, new capabilities in AI systems have done much more to increase the usefulness of these systems than increased efficiency in already-existing capabilities. For instance, as AI has become more economically important over the past decade, most of this economic importance has primarily come from *new* AI capabilities, as opposed to *old* AI capabilities requiring less computational power.[12]
- Anecdotally, LLMs seem to further back up this idea. The biggest software advances in the use of LLMs tend to look more like capabilities advances than efficiency advances. For instance, Reinforcement Learning from Human Feedback (RLHF) allows for "fine-tuning" LLMs to fill certain roles (such as a helpful assistant) instead of simply imitating internet text. Additionally, prompt engineering techniques such as prompting an LLM to "think step by step" can be used to enhance the reasoning abilities of LLMs, or to otherwise elicit latent abilities. In addition to the efficiency benefits of these techniques, mentioned in the above section, they also greatly increase the usefulness of various AI systems.
- Additionally, efficiency improvements in training LLMs can be converted into capability improvements, by effectively scaling systems up so that new capabilities emerge. Consider two possible ways for AI companies to incorporate training efficiency improvements into LLMs: A) create LLMs that can perform equally well as previous systems, while being faster and operating at lower computational costs; and B) create systems that are the same (or larger) computational cost, with improved capabilities. AI companies do both of these things (e.g., the shift from GPT-3.5 to GPT-3.5 Turbo was mostly an example of (A), while the shift from GPT-3.5 to GPT-4 was an example of (B)). Notably, developers are generally much more excited by examples of (B) than examples of (A), and they tend to choose to incorporate the most powerful model available to them into their processes. If efficiency gains were instead the main story going on, we'd expect (A) to instead drive more enthusiasm.

11 This conclusion would also be consistent with the logic in footnote 9, which argued that runtime efficiency and training efficiency would tend to grow at roughly similar rates.

12 This comparison is somewhat unfair, however, because improved capabilities aren't due just to software improvements, but also to more hardware being available to train larger models. With that said, software improvements are still responsible for a substantial portion of improved capabilities.

So to summarize:

- The efficiency of AI software (both runtime efficiency and training efficiency) is doubling every **~6 months**, with substantial uncertainty.[13]
- This estimate ignores some post-training enhancements, which bring significant further gains.
- Software progress also enables qualitatively new capabilities, which are much more important than pure efficiency gains.[14]

AI progress will likely speed up as we approach ASARA

In all likelihood, before we create AI systems that can automate *all* AI R&D tasks (i.e., ASARA), we'll create AI systems that can automate *most* AI R&D tasks.[15] Such systems will presumably help researchers with both software R&D and hardware R&D, though in this piece we're focusing just on the software side.

Already, AI has started to help AI researchers with AI software R&D. For example:

- LLMs are commonly used by AI researchers to summarize and analyze research papers, enabling them to work faster.

13 For reference, this rate is faster than the oft-quoted rate for hardware progress associated with Moore's law, in which the number of transistors per chip and the amount of computation available per dollar each tend to double around every two years. Interestingly, this rate is similar to the rate at which large AI systems have been scaled up since ~2010, where the computational resources used to train notable AI systems has tended to double every 6 months or so.

14 You might think that we could estimate the total speed of software improvements by adding together the rate of efficiency improvements and capabilities improvements, which would imply that the overall rate was going substantially more than twice as fast as the rate of efficiency improvements alone (if we assume capability gains are responsible for more progress than efficiency gains). Unfortunately, we can't do this, as not all improvements stack in this manner (though some do). For instance, improvements in training efficiency can either be put towards improvements in runtime efficiency (with a faster model) or towards capabilities improvements (with a more powerful model). But we can't count the entire training efficiency improvement on both sides at the same time. Therefore, all we can say is that software improvements must be advancing at least as fast as efficiency improvements alone, and likely substantially faster.

15 The alternative would presumably imply even faster AI progress upon ASARA, as it would imply a large discontinuity in capabilities from pre-ASARA systems to ASARA.

- GPT-4 can help researchers design new "architectures" for AI systems. With that said, as far as we're aware, this technique has not yet been used to help advance the current frontier of AI systems.
- Programs such as GitHub Copilot act as autocomplete for computer code, allowing AI software researchers to work more quickly. Anecdotally, we've heard from several AI researchers that Copilot has increased their productivity while programming by around a factor of 2.
- LLMs can be used to generate high-quality training data to train *themselves* further on, leading to improvements in their capabilities. LLMs can additionally engage in prompt engineering, designing prompts for other LLMs that, at least under some conditions, yield better results than human-designed prompts.
- According to the AI evaluation nonprofit METR, OpenAI's recent system o1-preview is able to make headway on crafting scaffolding for other LLMs and on fine-tuning other LLMs, both tasks within METR's AI R&D evaluation task suite, meant to capture challenging aspects of frontier AI R&D (though their evaluation did not find that o1-preview could make meaningful headway on any of the other five tasks in the suite).
- Etcetera.

How should we expect AI software progress to change as more and more software R&D tasks are automated, often by systems that substantially outperform humans within their domain? Intuitively, this dynamic would speed up software progress considerably. Even if overall progress becomes bottlenecked by tasks that only humans are able to perform, human researchers would still be able to spend much more time focusing on these particular bottlenecked tasks, as AI systems would be performing all the other tasks.

What happens when we reach ASARA?

By the time we reach ASARA, it will be possible to automate all tasks within AI software R&D. Notably, the amount of computing power it takes to train a new AI system tends to be *much* larger than the amount of computing power it takes to run a copy of that AI system once it's already trained. This means that if the computing power used to train ASARA systems is then repurposed to run these systems, a gigantic number of these systems could be run in parallel, likely implying much larger "cognitive output" from ASARA systems collectively than what's currently available from human AI researchers.

Putting some numbers on this – if you have enough computing power to train a frontier AI system today, then you have enough computing power to subsequently run probably hundreds of thousands of copies of this system (with each copy producing about ten words per second, if we're talking about LLMs). But this number is only increasing as AI systems are becoming larger.[16] Within a few years, it'll likely be the case that if you can train a frontier AI system, you'll be able to then run many millions of copies of the system at once (assuming that efficiency gains are used to run more copies of the system instead of to run each copy faster). What that means is, by the time we reach ASARA, the total cognitive output of ASARA systems will likely be equivalent to millions of top-notch human researchers, at least if we assume that each of these copies can match the performance of a top human researcher (we address the possibility that this assumption will not hold in a paragraph below). A large portion of this cognitive output could then be aimed at performing AI software R&D.

Today, there are perhaps hundreds of thousands of researchers in the world doing AI software R&D of some form or another,[17] but the vast majority of these researchers are neither working on improving state-of-the-art AI capabilities nor are they close to the limits of human potential in terms of abilities. When we think about AI software R&D after the advent of ASARA, however, we should imagine a pool of virtual researchers equivalent to millions of top-notch human researchers, with a large portion plausibly focusing on advancing state-of-the-art capabilities.[18]

As already hinted at, however, the above comparison relies on assumptions that may wind up not holding. In particular, ASARA might initially only be able to fully automate AI R&D work by having AI systems "thinking" for thousands of seconds just to produce as much useful work as 1 second of

16 The reason this ratio is increasing as systems become larger has to do with Chinchilla scaling laws (as discussed in footnote 9). According to these scaling laws, if the computational budget for training an LLM increases by a factor of Z (or if training efficiency increases as much), then the size of the model should increase by a factor of $\sqrt{Z}$ (with the number of datapoints used in training also increasing by a factor of $\sqrt{Z}$). Because the computational resources required to run the LLM (once it's trained) will be proportional to model size, the computational cost to run the model will likewise increase by a factor of $\sqrt{Z}$. Since the computational cost to train the model is increasing at a faster rate ($\sim Z$) than the computational cost to run the model ($\sim\sqrt{Z}$), this means, as time goes on and models are scaled up, you can run more copies of a cutting-edge model with whatever computing resources were needed to train it to begin with.

17 Here's one way to estimate the number of AI researchers in the world. In 2023, total attendance across 14 major AI conferences was 63,000, with the highest attended AI conference (NeurIPS) having 16,000 attendees. Because some researchers attend multiple conferences, the total number of individuals who attended at least one of these conferences must be between 16,000 and 63,000 (though realistically, probably closer to the 63,000 end). Most AI researchers would not attend any of these 14 conferences in a given year, though a sizable minority would, indicating the global AI researcher population is probably within the hundreds of thousands.

18 While it's possible that the proportion of AI research focused on advancing state-of-the-art capabilities will be no higher after ASARA than it is now, there are several reasons to suspect it will: 1) State-of-the-art research may be harder than many other forms of AI research, and most human researchers might not be talented enough to contribute meaningfully to it. 2) Human researchers often specialize and would have a hard time retraining for state-of-the-art research if they didn't initially focus on the relevant areas (vs computer hardware can be repurposed to run different AI systems). 3) Once ASARA exists, it'll presumably be abundantly clear that improving ASARA systems is very valuable (more so than the extent to which advancing towards ASARA is seen as valuable today).

thought from a human expert (in which case even 10 million AI systems would initially only be able to do as much cognitive work as a few thousand human experts). Indeed, the performance of recent "reasoning models" can be significantly improved by drastically ramping up how much computing power is used to run each copy, allowing each to think for longer. This suggests that the first AI systems to match human experts across the board may do so using very large amounts of computing power for each task. Even so, efficiency improvements imply that after ASARA is developed, these computational costs will fall. And soon after ASARA is developed, AI will likely contribute orders of magnitude more cognitive labor to AI R&D than humans do.[19]

So how fast would we then expect AI progress to be? We can only speculate. But if the current rate of software progress implies AI efficiency has a doubling time of ~6 months (or less, if we include post-training enhancements), then the extra researcher capacity might significantly increase the rate of progress – as a ballpark figure, perhaps to a doubling time of a month or two.

After that point, you might think this very fast progress would quickly hit diminishing returns, especially considering that (by assumption) hardware would then be held constant. This sort of "fizzling out" is certainly one possibility. But it's not the only possibility.

To see why, let's return to our diagram of the factors relevant for AI progress after the creation of ASARA:

19 We've already seen how AI runtime efficiency is improving fast – we estimated earlier that it's doubling around every 6 months (implying a ~4x improvement per year). Further, we've discussed how AI software progress will likely speed up considerably from that rate as we approach ASARA. Even in a "pessimistic" scenario where the first ASARA systems are just barely able to fully automate AI R&D without providing much more cognitive output, that scenario won't last long. The fast pace of progress in AI efficiency would imply orders of magnitude more cognitive output from future ASARA systems within a short amount of time – either months or a small number of years, depending on how much the speed of AI progress had increased by then.

And there are also multiple ways the above line of thinking is conservative. Even the earliest ASARA systems will have a host of significant advantages over human researchers. ASARA systems could make copies of themselves after developing a research plan (thus improving coordination while following the plan), revert to backup copies of themselves that were saved at specific times, do focused work 24 hours a day, think more quickly than humans, run more copies thinking more quickly on the most important tasks, etc. ASARA systems would have a different set of relative advantages and disadvantages compared to human researchers, and we should not expect the first ASARA systems to just barely match human experts across all tasks. Instead, by the time AI systems can match human experts on the AI R&D tasks that are hardest to automate, AI systems should substantially outperform human experts on many other AI R&D tasks, implying greater overall cognitive output for ASARA, even initially.

Further, human researchers could also spend their efforts on the tasks that ASARA systems are worst at, where these systems do initially just barely match human performance and require large computational costs. In effect, many more researchers would focus on these specific tasks than occurs today, further increasing the total amount of cognitive effort put towards AI R&D. It's even possible that this effect will lead to the cognitive effort going toward AI R&D to increase by orders of magnitude even before developing ASARA (e.g., once AI systems are able to automate 95% of tasks, human researchers will be squeezed onto the final 5% of tasks, implying these human-performed tasks will get 20x more effort compared to today).

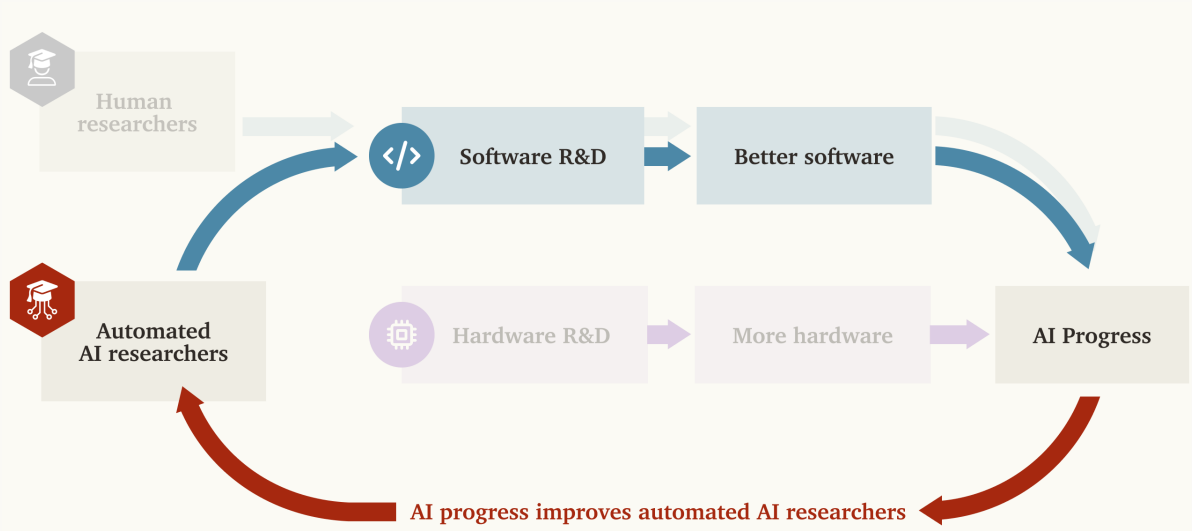


Figure 4: *Diagram of AI progress after the creation of ASARA and if hardware is then held constant (repeated from Figure 2).*

With humans completely cut out of the cycle, the feedback loop might go explosive, with progress getting faster and faster – i.e., an SIE. But whether or not we'd get that outcome would depend on the power of the feedback loop compared to countervailing forces.

A toy model to demonstrate the dynamics of a software intelligence explosion

Let's consider a toy model, representing the above feedback loop in the aftermath of achieving ASARA, with total computational power held constant. This toy model will demonstrate two competing dynamics:

- Diminishing returns to software R&D, as software improvements get harder and harder to find.
- The positive feedback from increasingly powerful ASARA systems.

The toy model will also involve several simplifying assumptions.

One, we'll assume that ASARA systems can all be broken up into distinct artificial general intelligences, or "AGIs," each of which is able to perform all the tasks involved in software R&D. Two, we're assuming all AI progress involves writing "papers" in AI, with each paper representing an incremental amount of progress, such that progress can be measured simply by the number of cumulative papers written. Three, all AGIs are equally productive as each other, and this productivity can be described simply as the number of papers written per unit time. And four, AGIs cannot be made more (or less) productive over time, but they can be made more "computationally efficient" – requiring less computing power to run each AGI. (Astute readers may realize that this toy model ignores capability improvements and only considers efficiency improvements.)

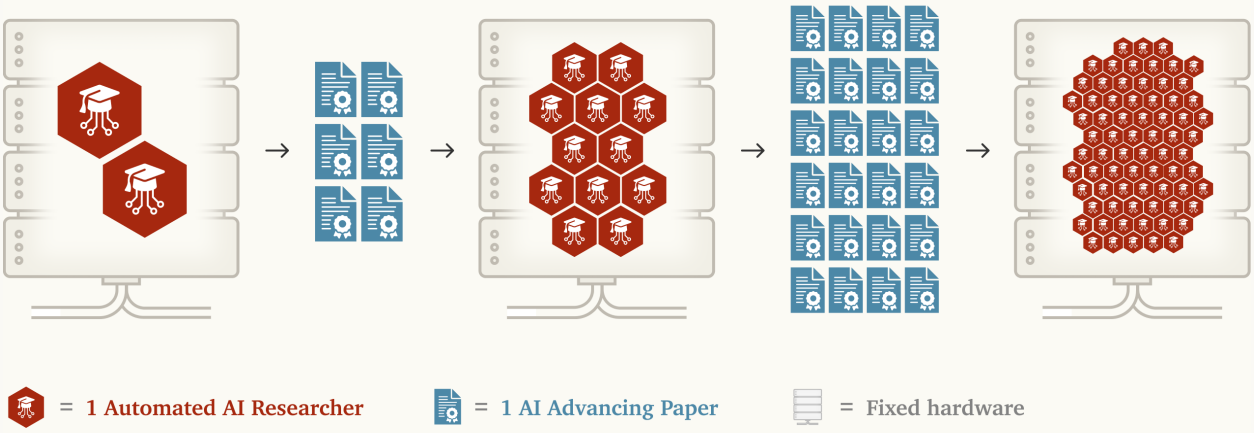


Figure 5: *Illustration of the toy model. In the toy model, we're assuming that AGIs (represented in this figure by neural nets in graduation caps) perform software R&D, resulting in "papers," and these papers allow for increasing the efficiency of our AGIs, enabling the fixed amount of hardware to house more AGIs, resulting in more total papers per unit time, and so on.*

Okay, first we'll consider what a "fizzle" looks like in the toy model (see Figure 6 below to follow along with this example). Let's say initially, there's 1 AGI (for simplicity). We'll also assume that the AGI's productivity is 1 paper per month, and that computational efficiency can be doubled after 2 papers have been written. Then after 2 months have passed, efficiency has doubled, so the same amount of hardware can be repurposed to house 2 AGIs. Both AGIs will be able to write 1 paper per month, meaning the total productivity is now 2 papers per month. But, because of diminishing returns to software R&D, the amount of papers needed to double efficiency again will increase – let's say it's now 3x higher, at 6 papers. With 2 AGIs each writing 1 paper per month, it will take 3 months to write the 6 papers needed to double efficiency for the second time. At that point, the hardware will allow for 4 AGIs (efficiency has doubled twice, and 2*2 = 4).

The number of papers needed to double efficiency for a third time will be higher still – let's imagine it's tripled again, to 18 papers. With 4 AGIs, this next doubling will take 4.5 months (because 18 papers / (4 papers / month) = 4.5 months).

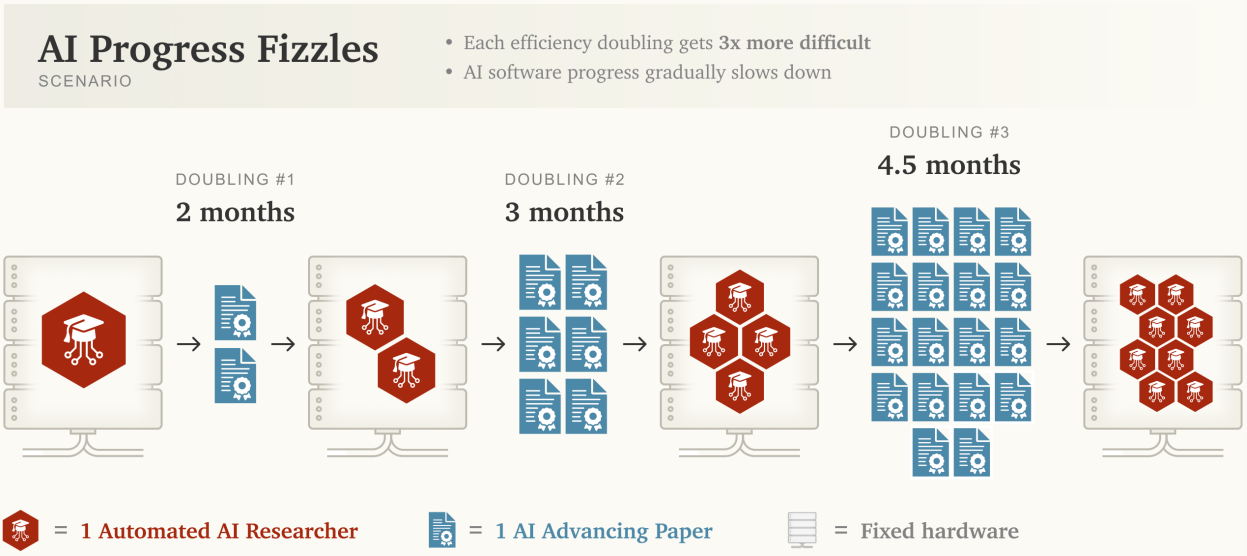


Figure 6: *Illustration of an example "fizzle" within the toy model. In this example, each AI efficiency doubling requires three times as many papers to be written as the last. With twice as much researcher capacity, each efficiency doubling therefore takes 50% longer than the last.*

Progress in this case is "fizzling," in the sense that it's taking longer and longer for each subsequent doubling – first 2 months, then 3, and then 4.5.

Now let's consider an SIE (see Figure 7 below to follow along). Again, we'll assume there's initially 1 AGI that can produce 1 paper per month, and that the first efficiency doubling requires 2 papers to be produced. This time, the second efficiency doubling will still require more papers than the first efficiency doubling – there are still diminishing returns to software R&D – but it won't require *that* much more. Let's imagine the second doubling in this case requires 3 papers – 50% more than the first doubling. With 2 AGIs, this number of papers can be produced within just 1.5 months (i.e., 3 papers / (2 papers / month)). And so on. Doublings are getting faster and faster.

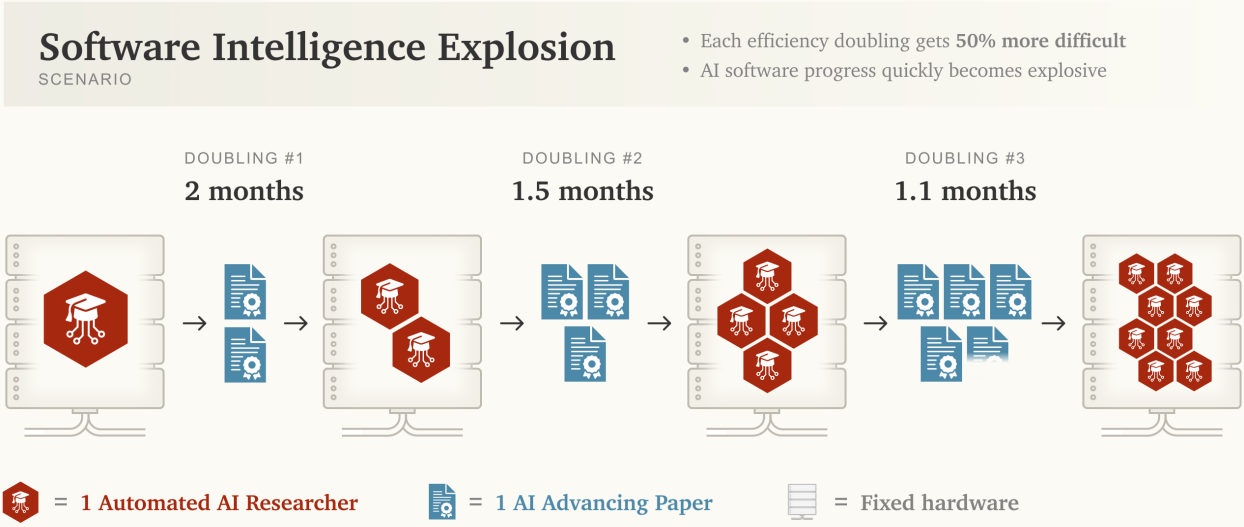


Figure 7: *Illustration of an example SIE within the toy model. Here, each AI efficiency doubling requires 1.5 times as many papers to be written as the last. With twice as much researcher capacity, each efficiency doubling therefore takes 75% as long as the last.*

In this case, progress is speeding up. If we simply extrapolate the dynamic within the confines of the toy model, it would imply infinite progress within finite time.

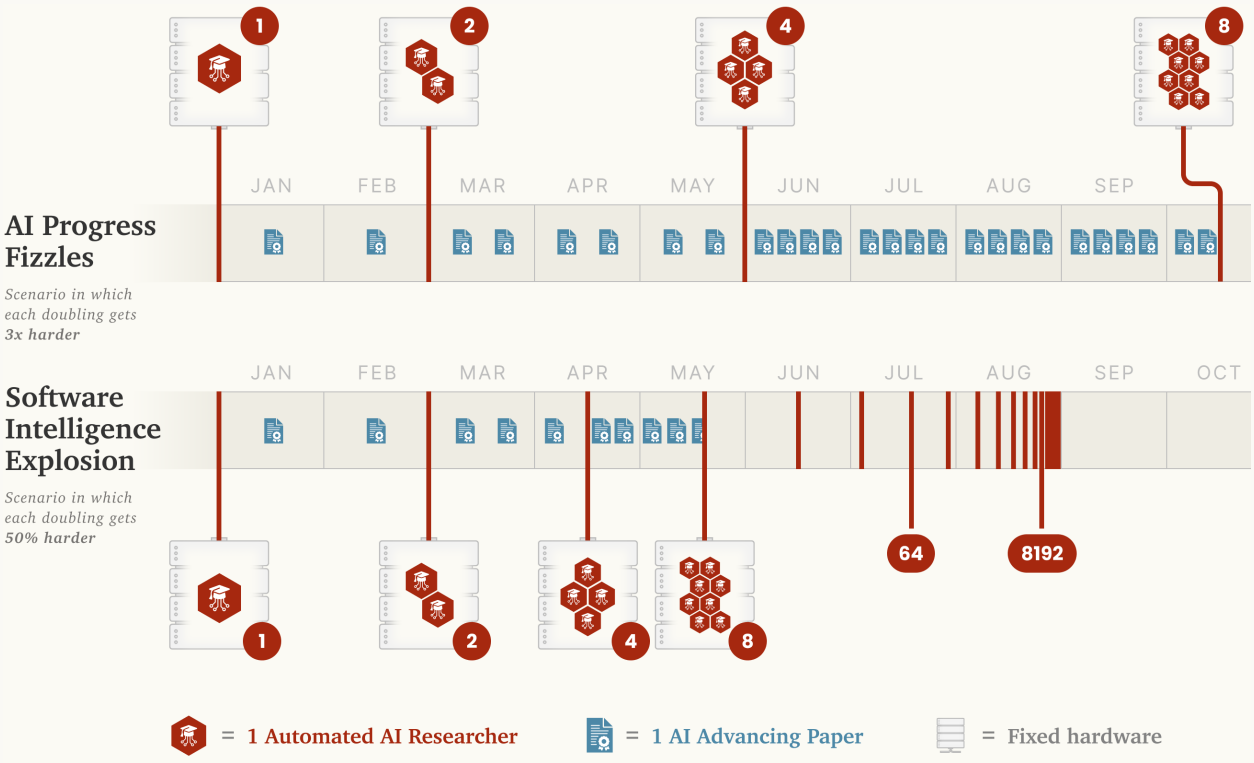


***Figure 8**: Comparison of our fizzle scenario and our SIE scenario. While progress gradually slows down in the fizzle, in an SIE it speeds up without limit. Given the parameters of our toy model, the SIE has an asymptote at 8 months after AI R&D is fully automated.*

In each of these two cases, we see both a) it gets "harder" over time to increase efficiency, requiring more papers for each subsequent doubling, and b) the number of AGIs increases over time, increasing the total effort going towards improving efficiency. The differentiator between the fizzle and the SIE is the relative strength of these two effects. Specifically, in a fizzle, each doubling of efficiency requires *more than twice* as many papers as the last (e.g., from 2 –> 6 –> 18), implying progress is getting harder at a faster rate than the AGI labor force is improving. Meanwhile, for an SIE, each efficiency doubling requires *less than twice* as many papers as the last (e.g., from 2 –> 3), implying progress is getting harder at a slower rate than improvements in the AGI labor force. If each efficiency doubling required papers to *exactly* double, then we'd see sustained exponential growth in efficiency, as each efficiency doubling would continue to take 2 months (e.g., with 2 AGIs needing to produce 4 papers, 4 AGIs needing to produce 8, ...).

	Papers for first efficiency doubling	Papers for second efficiency doubling	Relative increase *for each subsequent doubling*
SCENARIO #1 **AI Progress Fizzles**	2	6	3x (from 6 / 2 = 3)
SCENARIO #2 **Software Intelligence Explosion**	2	3	1.5x (from 3 / 2 = 1.5)
SCENARIO #3 **Exponential growth**	2	4	2x (from 4 / 2 = 2)

Table 2: *Chart demonstrating the differentiator between a fizzle and an SIE in our toy model. Note the specific number of papers listed for each doubling in each scenario is simply illustrative; the important point is that the tipping point between a fizzle and an SIE is when the papers required for a subsequent doubling in efficiency is twice as large as the papers required for the current doubling. Colored font used to help readers follow along.*

Being more mathematically concrete: *returns to software R&D*

We can capture the above considerations using a single variable, *returns to software R&D* (r). The variable r quantifies, as software improves, how much harder it becomes to improve AI software further, with lower values of r indicating it becomes much harder. Since we're now outside the toy model, we're not just focused on efficiency improvements but on capability improvements as well. (We'll say that a capability improvement "doubles" software if it increases AI's cognitive output by the same amount as if we had doubled efficiency.[20])

The value of r is set such that $r = 1$ corresponds to exponential growth, with each doubling in software capacity needing 2x as much research effort as the last. When $r < 1$, we get a fizzle, with each software

20 This is just a choice about how to define the units for "software." According to our definition, if you can run twice as many instances of all your AIs, and their capabilities are fixed, software has increased by 2x. With ASARA systems, that change would improve your ability to make research progress by some amount. If some capability improvement yielded the same improvement, then we would say that it too increased software by 2x. In other words, once we have ASARA, we measure the magnitude of software improvements via their practical effect on the ability to make further AI research progress. (Before we reach ASARA, the definition of a unit of software is, admittedly, murkier.)

doubling needing >2x as much research effort as the last. And $r > 1$ corresponds to an SIE, with each software doubling needing <2x as much research effort as the last.

More specifically, *r* gives the number of times software doubles for each time the cumulative work on software R&D doubles.[21] See Appendix for theoretical and empirical justification of using this formulation between these factors, based on lines of evidence from several fields of technology, and see Table 3 for clarification on how *r* is calculated in the context of the toy model.

	Relative increase *in papers for each subsequent doubling*	**Doublings in papers** *required for each additional efficiency doubling*	**Returns to software R&D** (*r*)
SCENARIO #1 **AI Progress Fizzles**	3x	1.58 (from: $\log_2(3) = 1.58$)	0.63 (from: 1/1.58 = 0.63)
SCENARIO #2 **Software Intelligence Explosion**	1.5x	0.58 (from: $\log_2(1.5) = 0.58$)	1.72 (from: 1/0.58 = 1.72)
SCENARIO #3 **Exponential growth**	2x	1 (from: $\log_2(2) = 1$)	1 (from: 1/1 = 1)

Table 3: *Continuation from Table 2, updated to demonstrate the derivation of r. Again, this table uses numbers that are illustrative of each scenario in our toy model, and colored font is used to help readers follow along. The tipping-point condition between an SIE and a fizzle is r = 1. Note the first column in this table is copied over from the last column in Table 2.*

21 Note that, while in the toy model, we were focused on the amount of work performed (i.e., papers written) between efficiency doublings, outside of the toy model we are focused on the *cumulative* amount of work performed, which incorporates all prior work performed in the field (the equivalent in the toy model would be to add up all the papers ever written until the time in question). Focusing on the work performed between efficiency doublings is simpler, while focusing on the cumulative amount of work performed is more useful when we're concerned with how parameters vary continuously instead of how they vary between discrete steps of software capacity doublings. Both formulations lead to similar results.

Being a bit more concrete about what this could all mean post-ASARA, let's imagine the software doubling time is down to 1 month when ASARA is first developed.[22] If r = 0.7, each subsequent doubling in AI software capacity will take 35% longer than the last one,[23] meaning the second doubling would happen in 41 days, then the third in 55 days, then 74 days, then 100 days; this would correspond to a ~30x improvement in AI software capacity in a bit under a year (with subsequent years seeing substantially slower progress). As a point of comparison, this yearly growth rate is perhaps somewhat similar to the rate of improvement in the capacity of cutting-edge AI systems today,[24] even though improvements in cutting-edge AI systems today include not only software advances, but also hardware advancements and increases in spending on hardware. Notably, these advances would be happening at a time when AI systems would already be extremely capable, making the situation more concerning than the same rate of progress occurring with today's systems. Of course, this comparison is incredibly rough (and the relevant metric is still fuzzy); the comparison isn't meant to be interpreted as a precise claim but instead as simply a plausible ballpark claim.

22 In a previous section, we speculated that the increased researcher capacity from ASARA systems might decrease the AI efficiency doubling time from ~6 months to ~1-2 months. Now that we're dealing with r, however, it's not enough to focus just on software efficiency – we need to instead focus on cognitive output from AI systems more generally, which includes both efficiency advancements and capability advancements (as described in footnote 20). Considering both of these factors together would imply substantially faster AI progress than focusing on efficiency improvements alone. Speculating further, we'll imagine this consideration reduces the doubling time down to a single month. Note that none of the conclusions in this piece depend on this exact figure, and the value of one month is picked largely for illustrative purposes.

23 We can see this result from the following math. If r = 0.7, then that means that if the cumulative amount of work performed on software R&D doubles, there will be 0.7 doublings in AI software capacity. We can alternatively formulate this relationship as 1 doubling in AI software capacity requiring (1/0.7) = 1.43 doublings in cumulative software R&D. Since doubling a variable x times amounts to increasing the level of the variable by a factor of 2^x, this increase in cumulative software R&D would amount to an increase by a factor of 2^1.43 = 2.69 – i.e., an increase of (2.69 - 1) = 1.69 of its original value. That is, the n + 1st doubling in software capacity will require 1.69 as much software R&D as had ever occurred before, and the nth doubling in software capacity will require (1.69/2.69) = 0.63 of the software R&D that ever occurs through the end of the nth doubling. The n + 1st doubling will therefore require (1.69/0.63) = 2.69 as much software R&D as the nth doubling. But the software capacity over the n + 1st doubling will be twice that of the nth doubling. The time to complete the n + 1st doubling will therefore be (2.69/2) = 1.35 that of the nth doubling, corresponding to 35% longer.

24 We saw above that AI software efficiency is likely doubling around twice a year, and we might update our rough estimate for AI software progress to doubling four times a year (i.e., increasing by 16x per year) to account for capability improvements. In addition to software, we're also currently seeing increases in hardware used for the most powerful AI systems (due to both hardware improvements and more money being spent on hardware) – since 2010, we've seen the computational resources used to train these systems increase by ~4x per year. If we simply multiply these numbers through, then we'd get an increase of ~64x per year. Given how rough and uncertain this whole calculation is, it seems appropriate to consider this in the same ballpark as the ~30x improvement estimated in the first year of a hypothetical fizzle.

Meanwhile, if $r = 3$, then each doubling will take 63% as long as the last one,[25] implying the next several doublings will take: 19 days, 12 days, 7.6 days, 4.8 days, and so on (until r itself decreases, potentially due to physical limits being approached[26]).

25 With $r = 3$, a doubling in AI software capacity will require (1/3) = 0.33 doublings in cumulative software R&D, corresponding to an increase by a factor of 2^0.33 = 1.26 (or by an amount of (1.26 - 1) = 0.26 of its original value). The n + 1st doubling will therefore require 0.26 as much software R&D as has ever occurred before, and the nth doubling will require (0.26/1.26) = 0.21 as much software R&D as has ever occurred through the end of the nth doubling. The n + 1st doubling in software capacity will therefore require (0.26/0.21) = 1.26 as much software R&D as the nth doubling in capacity. Again, the software capacity will be twice as high in the n + 1st doubling as in the nth doubling, implying the time to complete the n + 1st doubling will be (1.26/2) = 0.63 that of the nth doubling, or 63% as long.

26 Physical limits come into play for a couple reasons. First, the hardware stock introduces limits in how fast improvements can be made to software. For instance, signals can only travel so fast within the hardware, and software improvements cannot occur faster than these improvements can be implemented in the hardware. Second, given a fixed stock of physical hardware, there is a (incredibly large, yet still technically) finite number of distinct algorithms that could be run on the hardware. The finite number of possible algorithms sets a fundamental limit on how intelligent an AI system on the hardware could be. As these physical limits are approached, the rate of software improvement (and also r) must decrease. It's also possible other limits exist well below these limits, or that r will decrease well before these limits are approached for other reasons.

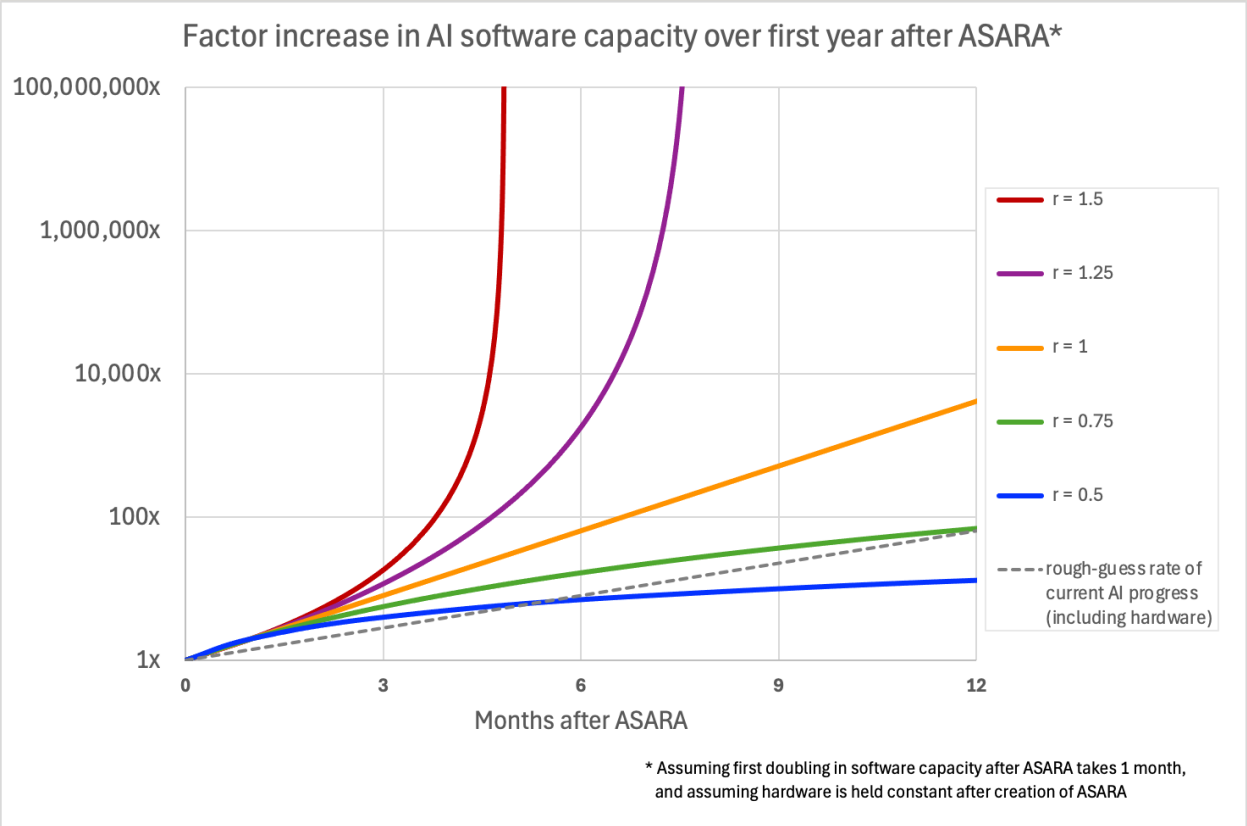


Figure 9: *Graph showing AI software progress over the first year after ASARA, as a function of r, assuming the first doubling in software progress takes 1 month and hardware is held constant upon the creation of ASARA. Note that for r > 1 (red and purple lines above), representing an SIE, progress becomes faster and faster, without limit (in reality, physical limits would eventually reduce r and slow progress, but it's unclear whether those physical limits would start biting before tremendous progress was achieved – see Figure 10 for more). For r < 1 (green and blue lines above), representing a fizzle, progress slows down over time, but could still be quite fast for a period of time (for reference, the dashed line on the graph corresponds to a very rough guess of the current rate of progress in AI capacity, including gains derived from hardware growth). Note that this graph makes simplifying assumptions, such as ignoring the "stepping on toes" effect.*[27]

27 The "stepping on toes" effect is an economics effect that captures inefficiencies introduced due to parallelization of work. By ignoring the effect, the graph assumes that 10 people working for 1 month can achieve the same amount of progress as 1 person working for 10 months (or, in the case of ASARA systems, 10x as much computing power being used on ASARA systems for 1 month can yield similar progress as 1x as much computing power being used for 10 months). This is unrealistic. A more realistic model that incorporated this effect would find that the value of *r* makes somewhat less difference to the growth trajectory, with both high and low values of *r* having trajectories closer to when $r = 1$. (The $r = 1$ trajectory would be unchanged, and still exponential, after incorporating "stepping on toes.") On the graph, this would mean all the lines would be somewhat closer to the orange line. With that said, we expect the "stepping on toes" effect to be smaller for ASARA systems than it is today, since coordination across ASARA systems working in parallel is likely to be easier than it is for people (e.g., because such systems could simply be duplicated and run in parallel, or could share databases in a way that humans cannot simply share memories).

A couple points from the above discussion are worth noting. First, the brief mention of sustained exponential growth may seem implausible to the point of silliness, as *r* would have to be exactly 1, and *prima facie* that seems very unlikely. But it's possible that a human response could keep us on this knife's edge. Perhaps human reactions will oscillate between wanting to slow everything down if progress starts accelerating and everything seems "too fast" (leading people to implement barriers to fast progress) and wanting to speed everything up when progress seems "too slow" (leading people to remove these barriers). This situation would mirror the collective response to COVID, in which, at times when COVID rates were high, lockdowns and other countermeasures were enacted, and at times when rates had fallen, these measures were largely abandoned.[28]

Alternatively, humans might intentionally craft policy to target constant exponential growth of AI capabilities, with the hope of having a more gradual increase in the capacity of AI systems. At the end of this piece, we list some preliminary policy suggestions for achieving this outcome, if it is in fact desirable.

Another thing to note is that even if we get a fizzle, we could still experience fast progress for some time. As the above discussion shows, if we had an initial doubling time of 1 month with an *r* of 0.7, within a year that would yield a relative increase in AI capacity that's perhaps similar to the yearly relative increase in the most powerful AI systems today, at a time when AI systems would be incredibly powerful.[29]

In an SIE, meanwhile, things would get much wilder than that. It's uncertain how far an SIE would continue before progress started slowing down; it's plausible that it could continue over a very large range before progress began slowing down.

28 This behavior kept the reproductive number (incidentally, also known as *r* within epidemiology) for COVID around 1 for long periods of time in many countries, despite forecasters initially expecting *r* would instead either be above 1 (relatively quickly leading to most people becoming infected with COVID and hospitals being over capacity) or below 1 (relatively quickly leading to COVID zero).

29 It's also possible that the rate of progress in that first year of a fizzle would be somewhat faster than the current rate, but it is very unlikely that a fizzle would see AI capacity increase at a rate (averaged over a year) that was *extraordinarily* faster than today. For example, if the initial doubling time was as short as *three days*, and if *r* was at 0.9, we would then see only 30 doublings in AI capacity over the first year, compared to around 6 doublings in AI capacity per year now; i.e., AI capacity would increase in that first year by about 5 years' worth of progress at today's rate. This rate would certainly be fast, but it wouldn't be the dizzyingly fast rate that we'd expect from an SIE. And of course, assuming hardware continued to be held constant, progress would further slow down from there – the next year would see around 1.5 years' worth of progress at today's rate, and the year after that would see around 1 year's worth of progress.

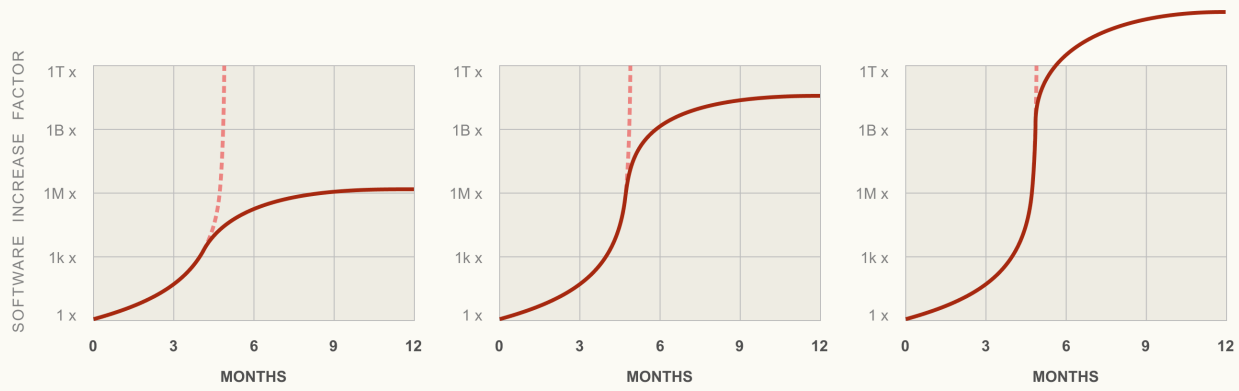


***Figure 10**: Even if there is an SIE, physical limits (as described in footnote 26) would eventually slow progress, if progress doesn't slow for other reasons first. There is tremendous uncertainty about how high these limits are above the first ASARA systems, though, and we may see extraordinary progress before these limits are approached.*

As we've discussed, whether an SIE would happen at all depends on whether *r* is above 1.

In the real world, are *returns to software R&D* greater or less than one?

Luckily, the formulation of *r* above allows us to examine this variable before reaching ASARA. Once ASARA is achieved, we'll simply be in the special case where the "work performed on software R&D" is being done by AI systems themselves, but today, *r* would correspond to how many times AI software doubles for each doubling in the cumulative amount of *human-performed* AI software R&D efforts. We can therefore estimate a value of *r* if we can measure how AI software R&D is growing currently and relate this growth to the growth in AI software capacity.

There are a few reasons to think that *r* is *currently* above 1:

- **Image recognition data.** We saw above that, from 2012 – 2022, image recognition algorithms have become increasingly efficient, with efficiency doubling times of around 9 months (according to the research group Epoch). This value is a measure of the growth of AI software efficiency; to derive a value for *r*, we also need a measure of the growth in AI software R&D efforts. We could then estimate *r* by estimating how much more R&D was needed for each subsequent software doubling, on average.[30] For various technical reasons,

30 That is, we could estimate *r* by simply dividing the overall value for software R&D doubling time by efficiency doubling time.

however, this estimate would be rough and susceptible to noise; more sophisticated analysis could better capture uncertainty and/or incorporate more data to better model the underlying dynamics.

Epoch has performed this exact analysis, where they assume software R&D can be proxied by the number of researchers in the field.[31] According to their analysis, the median likelihood value for r in computer vision is 1.4, though with large uncertainty – the 5th to 95th percentiles for r spanned 0.8 to 2.4, and uncertainty would be wider still if we also consider that their choice of variable for software R&D work may not have matched the true research input in the field (e.g., perhaps the average researcher quality changed over this time, or perhaps some other measure of the number of researchers in the field would be more apt). Still, the bottom line is that each time image-recognition training efficiency doubled, the work needed to double it again likely *less than* doubled. And preliminary internal analysis from Epoch suggests that the value of r is as high or higher for LLMs.

That said, we should interpret these results with a bit of caution. Epoch's analysis on image recognition algorithms focussed on the efficiency of *training* these image recognition systems, not the efficiency of running the systems, like in our toy model, and therefore $r > 1$ cannot straightforwardly be interpreted as a condition for an SIE. Still, these results hint that software returns in runtime efficiency alone may enable an SIE; the prospect of converting training efficiency improvements into larger AI systems with capability improvements only increases the chances of an SIE.[32]

31 To estimate the number of researchers in the field at various times, they used the number of unique authors who, according to the OpenAlex database, have published papers that touched on both computer vision and deep learning.

32 While converting between training efficiency and runtime efficiency isn't trivial, we should expect runtime efficiency to grow at a relatively similar rate to training efficiency. When OpenAI performed their own analysis on image recognition efficiency improvements over a portion of the same timeframe, they found that training efficiency and runtime efficiency wound up growing at very similar rates – 15 month runtime efficiency doubling time vs 16 month training efficiency doubling time. (Footnote 9 also argued that for LLMs, runtime efficiency will tend to grow at a roughly similar rate to training efficiency.) If r is similar between runtime efficiency and training efficiency, then the $r > 1$ condition for training efficiency would still be sufficient for an SIE, as it would imply $r > 1$ for runtime efficiency as well.

We may also consider that improved training efficiency allows for training larger (and therefore "smarter") models, with qualitative improvements. If it's more impactful (for AI R&D) to train these larger models than it is to instead use the efficiency improvements to train cheaper models at the same capability level, then the $r > 1$ condition for training efficiency would be an even stronger indicator of conditions suitable for an SIE (though admittedly, still not a proof). In this case, even if runtime efficiency grows somewhat slower than training efficiency, the qualitative leaps in AI capabilities enabled by larger models could more than make up for this difference in the context of an SIE.

Taking these points together, we think it is reasonable to consider the $r > 1$ condition for training efficiency as a rough indicator for a likely SIE. With that said, more research is warranted into the relationships between training efficiency, runtime efficiency, and the relative importance of each within the context of automated AI research.

- **Other areas analyzed by Epoch.** Epoch has also performed similar analysis for a few other areas of algorithmic efficiency related to AI to varying degrees: computer chess, reinforcement learning training data efficiency, Boolean satisfiability problem solvers, and linear programming. For each of these areas, they obtained estimates of *r*, and their central estimate of *r* for each, respectively, was: 0.8, 1.6, 3.5, and 1.1. That said, each of these four results would be expected to hold less relevance for the cutting edge of AI than the image recognition results above.[33]

- **Algorithms in general.** When we looked at the speed of AI software progress above, we considered how algorithms writ large provide an outside perspective to check our AI-specific results against, and we can do something similar here. While the analysis referenced in that section showed substantial differences in rates of efficiency improvements across different classes of algorithms, we can index on the median rate across all the examined algorithm classes. That analysis found the median rate of efficiency improvement to be 28% per year, when considering problems with large datasets.[34]

  Estimating the amount of work performed on relevant software R&D over time is somewhat harder, though we can note that within the US, employment within computer programming jobs increased by ~12x from 1970 – 2014, while Bachelor's and Master's degrees in computer science increased by ~25x and ~20x over the same time period, respectively. These increases would all correspond to an average yearly growth rate of around 7%, give or take a percent or two. Since 28% is 4x as large as 7%, this would all imply *r* here was around 4, though the assumptions going into this estimate are questionable, and the result should therefore be taken with an especially large grain of salt.

- **Sources of multiplicative software improvement.** Improvements in training algorithms interact multiplicatively with post-training enhancements like fine-tuning and scaffolding. Similarly, improvements in training algorithms stack with methods that *allow models to "think" more quickly* (produce more tokens per second), such as quantization; if an AI system can do 1 month's worth of thinking each day, that could significantly speed up the pace of

33 The current main paradigm of AI is deep learning, which underpins the success of everything from LLMs to self-driving cars to image recognition systems. But Boolean satisfiability problem solvers and linear programming don't typically rely on deep learning, and the particular computer chess algorithms that Epoch analyzed also don't depend on deep learning. Epoch's analysis of reinforcement learning, meanwhile, does relate to deep learning, but, crucially, this result didn't look at improvements in *computational* efficiency (what we mean by "efficiency" in the rest of this piece) but instead at *data* efficiency. Data efficiency would be expected to be somewhat related to computational efficiency, but not identical to it. Further, similar to the image recognition analysis, Epoch's analysis of reinforcement learning focused on the efficiency of *training* AI systems instead of *running* these systems, implying the same difficulties with drawing conclusions from image recognition results would also apply to the reinforcement learning results.

34 Specifically, problems where N = 1 billion.

subsequent software progress. When accounting for these additional factors that stack with training algorithms, our estimate of *r* should increase, perhaps by up to 2x.[35]

- **Capability improvements.** As we saw in previous sections, capability improvements are also a large contributor to AI progress, and they are not fully captured by the above analyses that focus on efficiency improvements. Therefore, the true value of *r* might be a fair bit larger than whatever we'd assume from efficiency alone, again plausibly by 2x.

	Image recognition	Computer chess	RL training data efficiency	SAT solvers	Linear programming	Algorithms in general
r EST.	1.4	0.8	1.6	3.5	1.1	~4?

Table 4: *Summary of estimates of r from various domains related to AI. Note that post-training enhancement and capability advancements would further increase our estimate of r.*

Considering both efficiency improvements and capability improvements together, as well as sources of multiplicative software improvement, we might currently expect that a single doubling in cumulative AI software R&D efforts would lead to a few doublings in AI software capacity (i.e., our best guess for *r* should perhaps be **~1-4**, though with high uncertainty[36]).

35 Notably, Anthropic, as part of a larger argument, recently made an "informal estimate" that post-training enhancements were responsible for improvement in LLMs equivalent to a 3x/year increase in the amount of computing power used to train cutting-edge systems. This estimate is similar to the rate they cited for algorithmic efficiency improvements in LLMs (2.8x/year). If these two rates of improvement are indeed similar, that would imply that we should double our value of *r* based on post-training enhancements.

36 Note that this uncertainty cuts both ways – while we should remain open to the possibility that *r* is substantially *lower* than our best-guess estimate, we should also remain open to the possibility that the true value is substantially *higher*.

This result might sound incredible, but it would simply put software on footing not that different from hardware; in a previous report, Tom Davidson estimated *r* for hardware and found that historically it's been ~7, while for AI chips (specifically, GPUs) from 2006 – 2022 it's been ~5 (i.e. each time cumulative R&D spend doubled, the cost of computing power halved 5-7 times) – though these estimates might be somewhat overestimated.[37] While computing hardware is famous for having grown very quickly over the previous several decades, what's less well known is that software progress may have grown similarly quickly.

However, the current value of *r* is presumably unsustainable in the long run; we expect that there is *some* fundamental physical limit to AI capabilities achievable with a constant amount of hardware, and software progress would presumably slow as we approach this limit.

But there isn't a good reason to expect this limit to be only slightly above the first ASARA systems, which may be imagined as approximately just substituting for human workers within relevant cognitive domains. Humans are presumably not the most intelligent lifeform possible, but simply the first lifeform on Earth intelligent enough to engage in activities like science and engineering. The human range for cognitive attributes is wide, and humans continue to gain from

37 Specifically, *r* for hardware in that report is defined as the number of times price-performance of hardware (in terms of FLOP/$) doubles for each doubling in cumulative, inflation-adjusted semiconductor R&D spending. Analogously to software *r*, the purpose of hardware *r* is to measure how much more difficult it becomes to improve hardware as hardware improves.

The estimates mentioned in the text for hardware *r* (~7 and ~5) may be overestimates, as they assume the only factor influencing hardware price performance is explicit R&D spending. In reality, other factors have also been at play. Notably, price performance of hardware has additionally improved due to "learning by doing" (i.e., workers at semiconductor plants becoming better at their jobs) and economies of scale, and the estimates for hardware *r* will be inflated by instead attributing these improvements to semiconductor R&D.

Additionally, hardware has improved due to spillover effects from other scientific and technological progress outside of semiconductor R&D. However, this factor only adds uncertainty to *r* instead of inflating it, as we might consider the "true" value for *r* should depend on the total relevant research that advances hardware, regardless of whether the work is technically classified as "semiconductor R&D" or not. That is, if other relevant areas of research (leading to these spillover effects) grew in spending at similar rates as semiconductor R&D spending, then *r* would be unchanged; if they grew at faster rates, then the reported values of *r* would be inflated; and if they grew at slower rates, then the reported values of *r* would be deflated.

With all that said, we believe the reported estimates remain acceptable approximations. Semiconductor R&D and closely related research likely accounts for the majority of hardware improvement over recent decades. Additionally, it's unlikely that other relevant research areas have grown substantially faster than semiconductor R&D, which has seen rapid growth. Thus, while the *r* estimates for hardware may be somewhat optimistic, they are likely not wildly inaccurate.

Note, however, that the comparison between software *r* and hardware *r* has another disanalogy – much of the increased spending on hardware R&D is spent on more expensive experiments instead of on more researcher labor, while the possibility of an SIE is dependent on automated researcher labor, specifically. (See the next section in this report, You might need fast growing computing power to discover better algorithms, for discussion of how we should adjust our expectations of an SIE in response to a related factor in software advancement.)

expanding population and specialization, as well as various cultural developments, indicating no fundamental limit in sight. In addition, ASARA will most likely be trained with orders of magnitude more computational power than estimates of how many "computations" the human brain uses over a human's development into adulthood, suggesting there's significant room for efficiency improvements in training ASARA systems to match human learning.[38]

So while *r* may currently be above 1, it will have to eventually fall – at the fundamental limits, it would need to be 0 (implying no further progress no matter how much effort is thrown at R&D), but it's unclear how quickly *r* will fall over time as we approach these limits. The further away the limits are from ASARA, however, the more likely *r* is to still be above 1 at that time, and the greater chance we'll have an SIE (which would then presumably continue until *r* dropped below 1, at which point progress would start slowing down). We may also note that the sooner we reach ASARA, the more likely it is that *r* will not have fallen to 1 by then, so the more likely we will have an SIE. Shorter timelines may therefore be thought of as increasing the chances of an SIE.

While the discussion so far does hint towards the plausibility of an SIE, it's far from a proof, and it might turn out to be wrong in important ways. Perhaps most notably, holding hardware constant may significantly decrease *r*, for a very simple reason:

You might need fast growing computing power to discover better algorithms

The analyses for AI software progress conducted by groups like OpenAI and Epoch, discussed above, all occurred in a context of increasing computing power. Perhaps humans working on AI software R&D weren't as responsible for software progress as we're imagining, and instead the key enabler of this software progress was the increasing amount of hardware. After all, hardware can be used for running AI experiments (e.g., to find better algorithms), so more hardware would mean more and/or

38 Investigations into the computational power necessary to match the human brain tend to yield estimates of around 10^15 FLOP/s. If we consider that the equivalent of "training" for a human brain is all the learning and brain processing that occurs from birth to productive adulthood, this would correspond to around 10^9 seconds (i.e., ~30 years * 3×10^7 seconds/year). This would all imply that the computational operations corresponding to human "training" would be ~10^24 FLOP (from 10^15 FLOP/s * 10^9 s). Already, the most powerful AI systems today are being trained with close to 10^26 FLOP, and this figure is rising rapidly. Once we reach ASARA systems, their training costs will likely be substantially greater, implying many orders of magnitude more training computations than what's equivalent for "training" the human brain, and thus many orders of magnitude improvement possible for training efficiency of ASARA systems just to match the efficiency of the human brain, assuming the estimates of the computational power for the human brain are not wildly off.

larger AI experiments. Take away the continued expansion of computing power, and maybe most of the software gains also dry up.

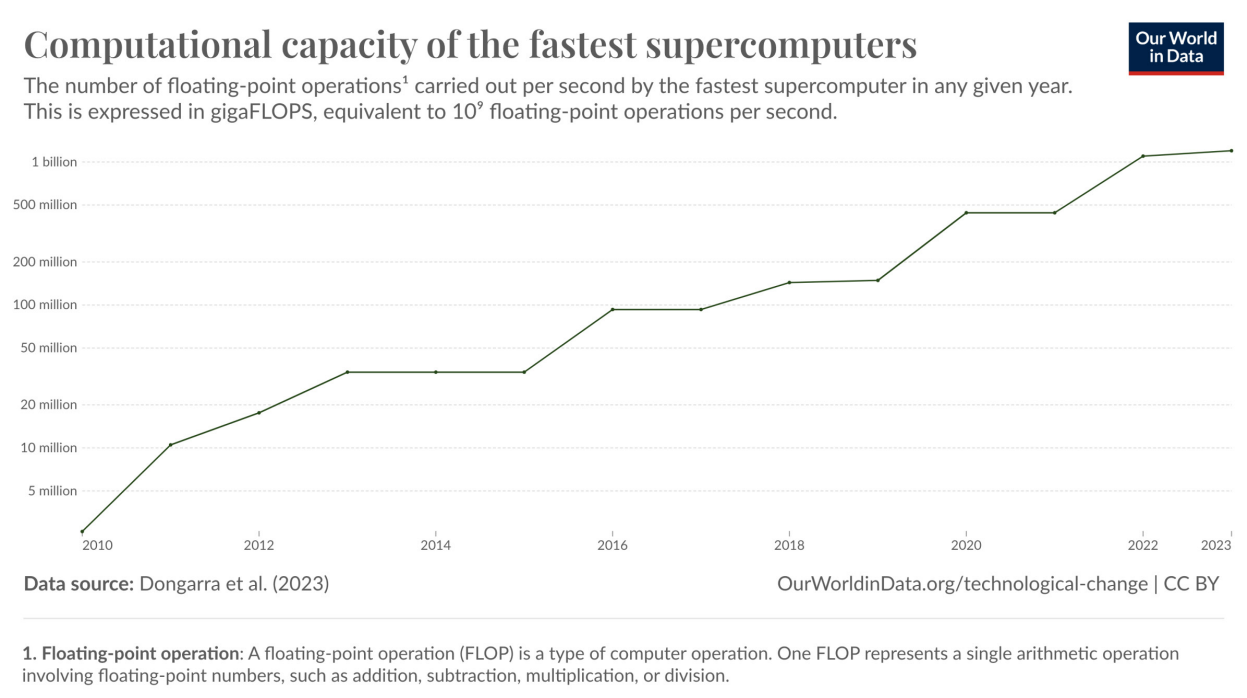


Figure 11: *Graph of the fastest supercomputer over time, from 2010, demonstrating the rise in computing power available. This increase in hardware may be largely responsible for improvements in AI software over this time period, as it would have enabled researchers to perform more AI experiments over time. If hardware were instead held constant, software progress might have also been slower. Note that this graph is a log plot (the y-axis grows exponentially), meaning the approximately straight line on this graph corresponds to exponential growth. (Figure source: Dongarra et al. (2023) – with minor processing by Our World in Data)*

On the other hand, improvements in software efficiency should lead to a *decrease* in the computational costs of running each AI experiment, all else equal. If algorithmic improvements allow for training a GPT-3-sized AI system on a laptop, for instance, then every researcher with a laptop can run their own GPT-3-sized experiment. So it will be possible to run more experiments over time even with constant hardware, and this effect may be enough to sustain fast efficiency progress.[39]

39 Even if continued software progress requires more experiments to be done over time due to lower hanging fruit already being picked, for this objection to hold, the volume of these experiments would need to increase at a rate faster than the decline in computational cost per experiment. Though on the other hand, if we improve AI capabilities (as well as efficiency), and useful experiments require scaling up systems to exhibit improved capabilities, then this will reduce the number of experiments that can be run (e.g., if the cutting edge is a GPT-12-sized system, then experiments with GPT-3-sized systems may not be particularly informative, and experiments may instead require something like training GPT-10-sized systems).

Additionally, if hardware limits do become more of a bottleneck to software progress, then AI companies could run smaller and cheaper experiments to compensate and extrapolate conclusions to larger systems. One reason to think it'll be possible to extrapolate significantly from smaller experiments is that LLMs and other cutting-edge AI systems often involve very clear relationships between a) the amount of computing power used to train the system and b) the resultant performance of the system. This point was demonstrated by GPT-4; OpenAI found that certain properties of GPT-4 were highly predictable from experiments they previously ran with AI systems trained with <1/1,000th as much computing power as GPT-4. It's conceivable that ASARA systems performing software R&D would similarly be able to generally infer the likely results of large AI experiments from running much smaller AI experiments, in which case they might tend to forgo the large experiments entirely.

Abundant cognitive labor from ASARA systems may also dramatically improve the quality, efficiency, and information value of AI experiments through several paths, including: eliminating bugs and subtle experimental design flaws before running experiments, more heavily prioritizing the most promising avenues of research, designing more valuable experiments with better reasoning from first principles, analyzing the results of each experiment in depth, synthesizing the results of each experiment with all other experimental results and pieces of evidence, constantly monitoring experiments and terminating them as soon as important results are in, etcetera.

Additionally, AI software R&D could shift to avenues that rely less on large experiments to begin with. For instance, methods for fine-tuning, scaffolding, and prompting do not generally involve tons of computing power, and experiments in these sorts of methods may continue to yield substantial gains.

It's even possible that, in the context of a strong hardware limit and a quickly expanding pool of ASARA systems for AI R&D, the field of AI would shift away from machine learning (with its computationally expensive training processes) and towards a new paradigm relying less on experimentation, perhaps even one that disposes of training altogether and instead relies on explicit design of desired AI systems, reminiscent of GOFAI.[40] While it's unclear exactly how the field of AI would change, we might suspect smart ASARA systems would find effective ways around hardware bottlenecks.

40 This scenario would be antithetical to the trend in AI over the past several decades, in which, "General methods that leverage computation are ultimately the most effective, [especially compared to methods that leverage human knowledge or techniques that don't scale well with computation]." But the reason for this historic trend has arguably been the exponentially increasing amount of computational power available (with increases in AI researcher labor being much smaller, relatively speaking). If instead we saw the situation flipped – computational power held constant with AI researcher capacity (from ASARA systems) fastly increasing – then we'd expect methods that leverage AI researcher labor to become increasingly competitive.

On the other hand, even if limitations on experimentation from the fixed amount of hardware aren't enough to stop software progress in its tracks, these limitations may still slow progress compared to the counterfactual. Consider the above workarounds that may allow for ASARA systems to still make substantial software progress despite hardware limits – implementing those workarounds may slow progress compared to if they weren't needed, substantially decreasing *r*.[41]

It's also possible that diminishing returns during an SIE will be generally steeper than in the historical data. Historically, computational resources were growing, so researchers could invent new algorithms that only work at new computational scales for which no one had previously tried to to develop algorithms. Researchers may have been plucking low-hanging fruit for each new scale of hardware. But this won't be possible in an SIE, when the hardware stays fixed. Restricting to algorithms at a fixed computational scale might make the diminishing returns much steeper.

Regardless, the objection addressed in this section is still an open question. Interested researchers could investigate whether historic advances in AI software (e.g., development of the transformer architecture) were enabled by rapidly increasing hardware resources or not.

Accounting for the constant hardware, we might reduce our best-guess estimate of *r* to ~0.5-2, with the estimate lower if progress requires large experiments and higher if improvements like prompting and scaffolding can go a long way.

And there's also another major reason that an SIE might be hindered, even if the value of *r* implies that it "should" occur:

Progress might become bottlenecked by the time required to train new AI systems

Under the current paradigm of AI, the most powerful systems are typically trained in two phases – a long "pretraining" phase and a much shorter "fine-tuning" phase. To simplify somewhat, the pretraining phase can be thought of as where the system primarily develops its capabilities, and the fine-tuning phase can be thought of as for modifying the system's behavior in desired ways or honing specific capabilities. For instance, for LLMs, the pretraining phase could yield a system being able to imitate internet text (requiring traits such as fluency with grammatical structures, ability to do basic reasoning, internalizing various relationships between aspects of the world, and so on), and the fine-

41 A more sophisticated analysis could model the dynamics here as software progress being elastic with respect to both AI researcher capacity and computational power and try to estimate the extent of elasticity.

tuning phase could involve steering the system into acting according to a specific role (such as being a helpful and harmless question-answering system).

For the most powerful systems, the pretraining phase can be long indeed, requiring continuous use of large data centers for several months. Recent advancements in AI change the story somewhat, as they hint at the fine-tuning phase becoming both longer and more central for developing capabilities than has been the case traditionally (though fine-tuning currently remains far shorter than pretraining). Regardless, it's precisely these long training phases – whether through pretraining, increasingly extensive fine-tuning, or other training phases that have yet to be developed – which might bottleneck AI progress, slowing down an SIE. If each generation of ASARA systems is able to create systems only so much more intelligent than themselves, and further, each subsequent generation needs to undergo a lengthy training process, then that could drastically dampen progress.

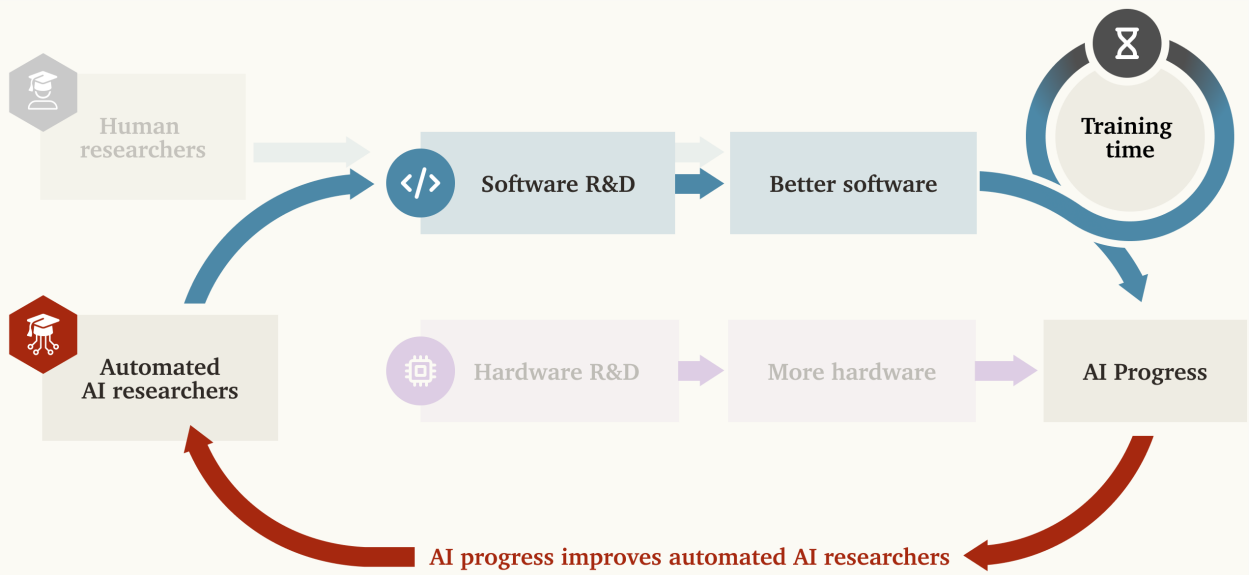


Figure 12: *Diagram of post-ASARA, software-based feedback loop, modified to demonstrate how the time needed to train new AI models could bottleneck AI progress, potentially preventing an SIE.*

But there are also several reasons that long training processes might not wind up bottlenecking progress. Progress might be able to be sustained by methods other than lengthy training processes – such as by focusing on scaffolding, prompting, and shorter fine-tuning phases. Additional methods might also be developed that allow for progress without retraining, such as by modifying parts of already existing systems in novel ways. And, as alluded to in the above section, a shift away from the current paradigm in AI might circumvent these hurdles even more clearly. If training new systems

becomes a bottleneck on progress, that bottleneck would lead to large incentives for the field to search out other ways of sustaining improvement.

And even if training new systems from scratch remains necessary, it's still plausible that an SIE could occur, as training new systems could become quicker than it is now. Specifically, algorithmic improvements could allow for greater efficiency in training new systems, thus requiring less time per each training run. For instance, if training runs for ASARA systems are initially two months, and then algorithmic improvements increase training efficiency by 30x, instead of plowing all of the efficiency improvements into training more powerful systems, these improvements could be split to train systems that are both more powerful and computationally less intensive (e.g., reducing the training time by 3x and having a further 10x improvement in efficiency to use for making the system larger). As long as each training run can be made somewhat faster than the last, training runs could eventually approach zero duration, and AI progress could become extremely fast.[42] Therefore, the bottleneck from having to train new AI systems would likely delay an SIE rather than prevent it.[43]

It's worth noting that the time necessary to train frontier AI systems isn't an immutable property inherent to the current AI paradigm, but instead is a compromise between various competing dynamics – including the value to finishing training earlier, the price of computing power, the expected change in price of computing power over time, and so on. In an SIE, the balance would shift heavily in favor of finishing training earlier (progress would be going so fast that your system would likely become outdated quickly, implying you'd want to deploy it faster), which might simply lead to much shorter training runs.

42 Consider, if each training run takes k times as long as the previous training run, where $k < 1$ (implying each training run is faster than the previous training run by some proportional amount). In that case, completing m training runs would take an amount of time equal to the first training run times the sum $\Sigma(k\hat{}n)$ from $n = 0$ to $(m - 1)$. Since $\Sigma(k\hat{}n)$ from $n = 0$ to ∞ converges for $|k| < 1$, an arbitrarily large amount of training runs could be accomplished in finite time, still allowing for an SIE. In order for an SIE to still be feasible, however, we must simultaneously have $r > 1$ and $k < 1$. In effect, you'd have to "spend" some of your software efficiency gains on making your training runs shorter, meaning software progress would accelerate less quickly than it would if you didn't have to do this.

43 By prolonging the time between ASARA and an SIE, the delay would give society more time to prepare. We can estimate how long this delay may be, doing a little bit of math. Note that for $|k| < 1$, the sum $\Sigma(k\hat{}n)$ from $n = 0$ to ∞ converges to $1/(1 - k)$. If $k = 0.9$, then this sum would equal 10, implying that if the first training run after ASARA takes 2 months, then it would take 20 months to complete an arbitrary number of training runs. With $k = 0.75$, the sum would instead yield 4, implying 8 months for an arbitrary number of training runs (again, assuming the first training run takes 2 months). Tom Davidson also created a toy model which more directly calculates the time from ASARA until strongly superhuman AI, comparing scenarios where a) improved AI systems require further training (though where retraining doesn't have to start over from scratch each time but could instead be continuous and cumulative) and b) where no training is necessary and AI R&D instead instantly leads to better AI systems. Based on this toy model and related analysis, Tom estimated that including the bottleneck related to training (as in scenario a) would lead to a ~1-3x increase in the time between the first ASARA systems and superhuman AI.

Again, this bottleneck is also an open question. Despite the possibilities above, it's also possible all approaches for progress that don't involve long training runs will either fail to pan out entirely or fail to sustain progress sufficiently, and maintaining $r > 1$ might preclude each subsequent training run getting shorter and shorter.

Bringing it all together

Given all the above, it seems at least decently likely that an SIE would occur if hardware were held constant upon the creation of ASARA and human social factors didn't prevent it, though we can't be confident either way.[44] If an SIE does occur, it would *very quickly* lead to *huge* gains in AI capacity – soon after ASARA, progress might well have sped up to the point where AI software was doubling every few days or faster (compared to doubling every few months today).

Even if an SIE does not occur, and we instead get a "fizzle," we could still see a period of AI software growing fast, perhaps about as fast as AI capacity is increasing now, considering not just software progress but also hardware progress and increases in spending on hardware. In other words, if we decided to *completely* pull the brakes on hardware increases once we reached ASARA *and* software progress is slow enough that it doesn't "spiral out of control," we might still expect to have a year or so of relative increases in AI capacity similar to today's rate of progress, but occurring when AI systems are already powerful enough to automate AI R&D. Any plans for there being a pause in AI improvements around the time of ASARA should take this possibility into consideration.

And considering that hardware will most likely not be held constant around ASARA, the chances of an intelligence explosion after ASARA should be even higher. Regardless, we should not be confident that hardware limitations mean AI progress will continue to be gradual and won't become incredibly fast.

More research is warranted into both the likelihood of an SIE and how to govern it. In particular, further research into the likelihood of an SIE should try to approximately pin down r for AI software R&D, and further research into governing an SIE should consider governance mechanisms that could either prevent an SIE or otherwise ensure successful governance of AI would continue throughout an SIE. Additionally, those evaluating AI governance proposals and scaling policies should consider if these policies are robust to an SIE.

44 If we had to quantify our forecast, we'd give a probability of somewhere between 30% and 60% of an SIE occurring under these conditions. We're somewhat hesitant to state the probability we'd put on an SIE happening, since we don't want readers to anchor too strongly on the exact probability we give.

What can we do if an SIE is possible?

Even if it turns out that an SIE is the expected default outcome, that does not make it a foregone conclusion. The actions leading up to an SIE would all be human choices, and likewise, human choices could set up processes that either avert this outcome or send it down a positive course. Setting up the right processes may be very difficult and require a high level of coordination, but such coordination is possible.

The following are preliminary governance and policy ideas that might help either avoid an SIE or direct it in positive directions. Our point with listing these preliminary ideas isn't to say that they should be taken as definitive solutions to an SIE, but instead simply to show that there are ways that we can act to start addressing the possibility, as well as to begin a conversation on the topic. We've recently seen leading AI companies take a few initial steps in these directions, for which we commend them.[45] These sorts of policies could be more broadly adopted either voluntarily by AI companies or through government regulation:

- **Ongoing measurement of software progress, disclosed to trusted 3rd parties.** Much of the analysis above relies on work from OpenAI and Epoch involving measurement of software progress in vision recognition systems and other AI systems. Without these measurements, we would have a much harder time understanding what was going on with AI software progress. But our current understanding is still spotty.

  Accurate measurement of software progress in frontier AI companies may give advanced warning about an impending SIE, potentially allowing society to take precautions before it's too late. Further, any entity responsible for assessing if an AI company is behaving responsibly and implementing sufficient safeguards should know whether the company might soon face substantial acceleration in software progress, which is difficult to know if we're not actively measuring progress.

- **Pre-training and pre-deployment assessments of the potential to automate AI R&D.** AI companies currently conduct evaluations on various safety and other attributes of

45 A few recent developments stand out. First, Google DeepMind's exploratory Frontier Safety Framework identifies machine learning R&D as a domain in which highly capable AI systems may pose severe risks, necessitating security and deployment mitigations once certain thresholds are reached. Second, OpenAI unveiled MLE-bench, a benchmark designed to assess AI systems' capabilities in ML engineering tasks, and their accompanying paper acknowledges the risks of AI systems accelerating progress beyond humanity's ability to ensure safe development. Third, Anthropic's recent Responsible Scaling Policy update explicitly recognizes that systems with autonomous AI R&D capabilities would require heightened safeguards, and the updated policy identifies a specific threshold at which these heightened safeguards would be needed.

their systems, such as the risks these systems pose to cybersecurity and biosecurity. These evaluations could be extended to include risks from accelerating AI progress due to new systems' capabilities in AI R&D tasks.

Companies could evaluate whether AI systems can competently perform a set of AI R&D tasks autonomously (while outfitting the system with the best scaffolding and tools, to avoid underestimating the capabilities of the system). They could also supplement these evaluations with real-world data by carefully measuring, over time, the productivity gains that existing AI tools provide their employees and extrapolating these trends forward for more powerful systems.[46] This would help inform assessments of how close we are to ASARA.

- **Adopting a threshold level of substantial AI-led software acceleration, which companies commit to avoid exceeding without proper precautions.**[47] AI companies could intentionally avoid developing systems which would speed up AI software progress, across the entire field of research, beyond some threshold pace (e.g. 5x the recent pace of software progress), unless specific precautions were in place.

  In order to get early warning signs that this threshold may be approaching, these companies could use proposals from the above bullet points. First, they could monitor the rate of software progress over time, to assess whether the rate of progress has increased to within some buffer of the threshold amount. Second, they could evaluate new systems' ability to perform AI R&D tasks, to assess the risks that any specific system would cause a surprisingly large jump in AI R&D capabilities that plausibly would lead to the pace of progress exceeding the threshold rate. Third, they could extrapolate based on how much existing AI systems are boosting employee productivity.

  Before developing systems that pass the threshold rate, companies would commit to putting specific protective measures in place. These measures could include: 1) strong information security, robust to both hacking attempts from well-resourced state actors and potential attempts from (rapidly increasingly capable) AI models to self-exfiltrate; 2) alignment, boxing, and internal monitoring techniques that could preclude the possibility of misaligned AI from "poisoning" the process of subsequent AI development; and 3) publicly legible external oversight, requiring 3rd party sign-off on high-stakes decisions, including decisions surrounding rapid advancements in AI progress. See more discussion here.

46 These ideas are discussed further in this draft from Tom Davidson: How Can AI Labs Incorporate Risks From AI Accelerating AI Progress Into Their Responsible Scaling Policies?

47 This idea (as well as some of the specifics in the rest of this bullet point) is discussed further in this draft from Tom Davidson: How Can AI Labs Incorporate Risks From AI Accelerating AI Progress Into Their Responsible Scaling Policies?

- **Other good governance practices.** The possibility of an SIE raises the stakes of AI governance practices more generally, as the situation could start to spiral out of control quicker than otherwise assumed. By the time AI companies start to notice AI software progress accelerating, it may be too late to begin instituting needed practices for combatting the risks. The list of governance practices that may be helpful here is too numerous to enumerate, but they should include formal practices that promote good governance (e.g., whistleblower protection for workers at AI companies[48]), as well as a culture of taking safety concerns seriously (e.g., instead of viewing safety as simply a checkbox exercise or something done primarily for PR reasons).

We'll likely find it easier to coordinate successfully on policies if all relevant actors have mutual assurance that none are developing superhuman AI too quickly (since fast development of superhuman AI would hint at the possibility of that actor pulling far ahead of the others on short notice). The governance proposals above would all help with this sort of assurance, in addition to the more direct benefits they offer. We'd therefore expect protective governance policies in this space to be somewhat self-reinforcing, as enacting these sorts of policies may make actors more comfortable coordinating further to enact further policies. In particular, mutual assurance around these issues would work best if such assurance can be accomplished via mechanisms that don't rely on high levels of trust between parties and that are compatible with software progress being hard to observe externally.

By the time we see clear signs that an SIE may be approaching, it might be too late to implement necessary changes. Unless we can rule out the possibility, we should be proactive and figure out how to navigate the terrain ahead of time.

48 As a general principle, whistleblower protections for those at AI companies may help uncover and disincentivize reckless behavior from those companies and may further ensure these companies follow other safety procedures.

Acknowledgements

We would like to thank the following people for providing helpful feedback and comments on a draft of this piece (in alphabetical order by last name): Tamay Besiroglu, Taylor Eth, Paul Ferrell, Lukas Finnveden, Holden Karnofsky, Eli Lifland, William MacAskill, Toby Ord, Bonnie Ross, Jaime Sevilla, Carl Shulman, Philip Trammell, and Lily Zhang. We are especially grateful to Sam Manning, Sören Mindermann, and Girish Sastry for significant contributions in crafting the summary of this work, and to Alex Savard for visuals and design work. We would like to thank Open Philanthropy for funding this work.

References

AI Impacts (2015). "Trends in the cost of computing." *AI Impacts*.
https://aiimpacts.org/trends-in-the-cost-of-computing/

AI Index (2024), with minor processing by Our World in Data (2024). "Annual attendance at major artificial intelligence conferences." *Our World in Data*.
https://ourworldindata.org/grapher/attendance-major-artificial-intelligence-conferences

Alexander, S. (2019). "Do Neural Nets Dream of Electric Hobbits?" *Slate Star Codex*.
https://slatestarcodex.com/2019/02/18/do-neural-nets-dream-of-electric-hobbits/

Altman, S. (2023). "Breakthrough potential of AI." [Interview]. *Imagination in Action, YouTube*.
https://www.youtube.com/watch?v=T5cPoNwO7II

Altman, S. (2024). "The Intelligence Age."
https://ia.samaltman.com/

Amodei, D. (2024). "Machines of Loving Grace: How AI Could Transform the World for the Better."
https://darioamodei.com/machines-of-loving-grace

Anthropic (2024). "Announcing our updated Responsible Scaling Policy." *Anthropic*.
https://www.anthropic.com/news/announcing-our-updated-responsible-scaling-policy

Anthropic. (2024). "Responsible Scaling Policy." *Anthropic*.
https://assets.anthropic.com/m/24a47b00f10301cd/original/Anthropic-Responsible-Scaling-Policy-2024-10-15.pdf

Appenzeller, G. (2024). "Welcome to LLMflation – LLM inference cost is going down fast." *Andreessen Horowitz*.
https://a16z.com/llmflation-llm-inference-cost/

Bloom, N. *et al.* (2020). "Are Ideas Getting Harder to Find?" *American Economic Review*.
https://web.stanford.edu/~chadj/IdeaPF.pdf

Carlsmith, J. (2020). "How Much Computational Power Does It Take to Match the Human Brain?" *Open Philanthropy*.
https://www.openphilanthropy.org/research/how-much-computational-power-does-it-take-to-match-the-human-brain/

Cavin, R., Lugli, P., & Zhirnov, V. (2012). "Science and Engineering Beyond Moore's Law." *Proceedings of the IEEE*. https://ieeexplore.ieee.org/document/6186749

Cottier, B. *et al*. (2024). "How Much Does It Cost to Train Frontier AI Models?" *Epoch AI*. https://epoch.ai/blog/how-much-does-it-cost-to-train-frontier-ai-models

Cotton-Barratt, O. (2014). "Theory behind logarithmic returns." *Future of Humanity Institute*. https://www.fhi.ox.ac.uk/theory-of-log-returns/

Davidson, T. (2023). "Continuous doesn't mean slow." *Planned Obsolescence*. https://www.planned-obsolescence.org/continuous-doesnt-mean-slow/

Davidson, T. (2023). "What a Compute-Centric Framework Says About Takeoff Speeds." *Open Philanthropy*. https://www.openphilanthropy.org/research/what-a-compute-centric-framework-says-about-takeoff-speeds/

Davidson, T. (2024). "How Can AI Labs Incorporate Risks From AI Accelerating AI Progress Into Their Responsible Scaling Policies?" [Draft]. https://www.forethought.org/research/how-can-ai-labs-incorporate-risks-from-ai-accelerating-ai-progress-into

Davidson, T. (2024). "Will the need to retrain AI models from scratch block a software intelligence explosion?" [Draft]. https://www.forethought.org/research/will-the-need-to-retrain-ai-models

Davidson, T. *et al.* (2023). "AI capabilities can be significantly improved without expensive retraining." *arXiv*. https://arxiv.org/abs/2312.07413

Delionback, L. (1975). "Guidelines for Application of Learning/Cost Improvement Curves." *NASA*. https://ntrs.nasa.gov/api/citations/19760006882/downloads/19760006882.pdf

Dongarra *et al*. (2023), with minor processing by Our World in Data (2024). "Computational capacity of the fastest supercomputers." *Our World in Data*. https://ourworldindata.org/grapher/supercomputer-power-flops?time=2010..latest

Dorner, F. (2021). "Measuring Progress in Deep Reinforcement Learning Sample Efficiency." *arXiv*. https://arxiv.org/abs/2102.04881

Dragan, A., King, H., & Dafoe, A. (2024). "Introducing the Frontier Safety Framework." *Google DeepMind*. https://deepmind.google/discover/blog/introducing-the-frontier-safety-framework/

Droppo, J. & Elibol, O. (2021). "Scaling Laws for Acoustic Models." *Interspeech 2021*. https://www.isca-archive.org/interspeech_2021/droppo21_interspeech.pdf

Epoch (2024), with major processing by Our World in Data (2024). "Computation used to train notable artificial intelligence systems, by domain." *Our World in Data*. https://ourworldindata.org/grapher/artificial-intelligence-training-computation

Epoch (2024), with major processing by Our World in Data (2024). "Parameters in notable artificial intelligence systems." *Our World in Data*. https://ourworldindata.org/grapher/artificial-intelligence-parameter-count

Epoch AI (2024). "We're investigating the trajectory of AI for the benefit of society." *Epoch AI*. https://epoch.ai/

Erdil, E. & Besiroglu, T. (2022). "Algorithmic progress in computer vision." *arXiv*.
https://arxiv.org/abs/2212.05153

Erdil, E., Besiroglu, T., & Ho, A. (2024). "Estimating Idea Production: A Methodological Survey." *arXiv*.
https://arxiv.org/abs/2405.10494

GitHub Copilot (2024). "The AI editor for everyone." *GitHub*.
https://github.com/features/copilot

Good, I. J. (1965). "Speculations Concerning the First Ultraintelligent Machine." *Advances in Computers*. https://www.stat.vt.edu/content/dam/stat_vt_edu/graphics-and-pdfs/research-papers/Technical_Reports/TechReport05-3.pdf

Grace, K. (2013). "Algorithmic Progress in Six Domains." *Machine Intelligence Research Institute*.
https://intelligence.org/files/AlgorithmicProgress.pdf

Hanson, R. (2013). "Why Do Algorithms Gain Like Chips?" *Overcoming Bias.*
https://www.overcomingbias.com/p/why-does-hardware-grow-like-algorithmshtml

Henighan, T., Kaplan, J., & Katz, M. *et al*. (2020). "Scaling Laws for Autoregressive Generative Modeling." *arXiv*.
https://arxiv.org/abs/2010.14701

Henrich, J. (2015). "The Secret of Our Success: How Culture Is Driving Human Evolution, Domesticating Our Species, and Making Us Smarter." *Princeton University Press.*
https://press.princeton.edu/books/ebook/9781400873296/the-secret-of-our-success-pdf

Hernandez, D. & Brown, T. (2020). "AI and efficiency." *OpenAI*.
https://openai.com/index/ai-and-efficiency/

Hernandez, D. & Brown, T. (2020). "Measuring the Algorithmic Efficiency of Neural Networks." *arXiv*.
https://arxiv.org/abs/2005.04305

Hestness, J. *et al*. (2017). "Deep Learning Scaling is Predictable, Empirically." *arXiv*.
https://arxiv.org/abs/1712.00409

Ho, A. *et al*. (2024). "Algorithmic Progress in Language Models." *Epoch AI*.
https://epoch.ai/blog/algorithmic-progress-in-language-models

Ho, A. & Besiroglu, T. *et al.* (2024). "Algorithmic progress in language models." *arXiv*.
https://arxiv.org/abs/2403.05812

Hobbhahn, M., Heim, L., & Aydos, G. (2023). "Trends in Machine Learning Hardware." *Epoch AI*.
https://epoch.ai/blog/trends-in-machine-learning-hardware

Hoffmann, J. *et al.* (2022). "Training Compute-Optimal Large Language Models." *36th Conference on Neural Information Processing Systems*.
https://proceedings.neurips.cc/paper_files/paper/2022/file/c1e2faff6f588870935f114ebe04a3e5-Paper-Conference.pdf

Huang, J. et al. (2023). "Large Language Models Can Self-Improve." *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
https://aclanthology.org/2023.emnlp-main.67.pdf

IRENA (2024), Nemet (2009), and Farmer & Lafond (2016), with major processing by Our World in Data (2024). "Solar (photovoltaic) panel prices vs. cumulative capacity." *Our World in Data*.
https://ourworldindata.org/grapher/solar-pv-prices-vs-cumulative-capacity

Jones, C. (1995). "R&D-Based Models of Economic Growth." *The Journal of Political Economy*.
https://www-leland.stanford.edu/~chadj/JonesJPE95.pdf

Kaplan, J. & McCandlish, S. *et al*. (2020). "Scaling Laws for Neural Language Models." *arXiv*.
https://arxiv.org/abs/2001.08361

Karnofsky, H. (2024). "If-Then Commitments for AI Risk Reduction." *Carnegie Endowment for International Peace*.
https://carnegieendowment.org/research/2024/09/if-then-commitments-for-ai-risk-reduction

Kojima, T. *et al.* (2022). "Large Language Models are Zero-Shot Reasoners." *36th Conference on Neural Information Processing Systems*. https://arxiv.org/abs/2205.11916

Kurzweil, R. (2005). "The Singularity Is Near: When Humans Transcend Biology." *Viking*.
https://www.singularity.com/

Landhuis, E. (2016). "Scientific literature: Information overload." *Nature*.
https://www.nature.com/articles/nj7612-457a

Leike, J. (2023). "Self-exfiltration is a key dangerous capability." *Musings on the Alignment Problem*.
https://aligned.substack.com/p/self-exfiltration

METR (2024). "Details about METR's preliminary evaluation of OpenAI o1-preview." *Model Evaluation and Threat Research*.
https://metr.github.io/autonomy-evals-guide/openai-o1-preview-report/

METR (2024). "METR: Model Evaluation and Threat Research." *Model Evaluation and Threat Research*.
https://metr.org/

Metz, C. (2020). "Meet GPT-3. It Has Learned to Code (and Blog and Argue)." *The New York Times*.
https://www.nytimes.com/2020/11/24/science/artificial-intelligence-ai-gpt3.html

Mithas, S., Kude, T., & Whitaker, J. (2018). "Artificial Intelligence and IT Professionals." *IEEE Computer Society*.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8509563

Neumann, O. & Gros, C. (2022). "Scaling Laws for a Multi-Agent Reinforcement Learning Model." *arXiv*.
https://arxiv.org/abs/2210.00849

OpenAI (2022). "Introducing ChatGPT." *OpenAI*.
https://openai.com/index/chatgpt/

OpenAI (2023). "GPT-4 Technical Report." *OpenAI*.
https://cdn.openai.com/papers/gpt-4.pdf

OpenAI (2024). "Learning to Reason with LLMs." *OpenAI*.
https://openai.com/index/learning-to-reason-with-llms/

Owen, D. (2024). "Interviewing AI researchers on automation of AI R&D." *Epoch AI*.
https://epoch.ai/blog/interviewing-ai-researchers-on-automation-of-ai-rnd

Sevilla, J. & Roldán, E. (2024). "Training Compute of Frontier AI Models Grows by 4-5x per Year." *Epoch AI*.
https://epoch.ai/blog/training-compute-of-frontier-ai-models-grows-by-4-5x-per-year

Shern, C. *et al.* (2024). "MLE-bench." *OpenAI*.
https://openai.com/index/mle-bench/

Shern, C. *et al.* (2024). "MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering." *arXiv*. https://arxiv.org/abs/2410.07095

Sherry, Y. & Thompson, N. (2021). "How Fast Do Algorithms Improve?" *Proceedings of the IEEE*. https://ide.mit.edu/wp-content/uploads/2021/09/How_Fast_Do_Algorithms_Improve.pdf

Srivastava, P. (2023). "Fine-Tuning AI Models: A Guide." *Medium*. https://medium.com/@prabhuss73/fine-tuning-ai-models-a-guide-c515bcd4b580

Sutton, R. (2019). "The Bitter Lesson." http://www.incompleteideas.net/IncIdeas/BitterLesson.html

Wei, J. *et al.* (2022). "Emergent Abilities of Large Language Models." *Transactions on Machine Learning Research*. https://openreview.net/pdf?id=yzkSU5zdwD

Wijk, H. *et al.* (2024). "RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts." *Model Evaluation and Threat Research*. https://metr.org/AI_R_D_Evaluation_Report.pdf

Wikipedia contributors (2024). "Artificial general intelligence." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Artificial_general_intelligence

Wikipedia contributors (2024). "Deep Blue (chess computer)." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Deep_Blue_(chess_computer)

Wikipedia contributors (2024). "Deep learning." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Deep_learning

Wikipedia contributors (2024). "Elasticity (economics)." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Elasticity_(economics)

Wikipedia contributors (2024). "Experience curve effects." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Experience_curve_effects

Wikipedia contributors (2024). "GOFAI." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/GOFAI

Wikipedia contributors (2024). "Moore's law." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Moore%27s_law

Wikipedia contributors (2024). "Prompt engineering." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Prompt_engineering

Woodside, T. (2023). "Examples of AI Improving AI." *Center for AI Safety*. https://ai-improving-ai.safe.ai/

Yang, C. & Chen, X. *et al.* (2024). "Large Language Models as Optimizers." *The 12th International Conference on Learning Representations*. https://openreview.net/pdf?id=Bb4VGOWELI

Zheng, M. (2023). "Can GPT-4 Perform Neural Architecture Search?" *arXiv*. https://arxiv.org/abs/2304.10970

Zhou, A. *et al.* (2024). "Language Agent Tree Search Unifies Reasoning, Acting, and Planning in Language Models." *Proceedings of the 41st International Conference on Machine Learning*. https://arxiv.org/abs/2310.04406v1

Appendix: Justification for our formulation of *r*

To reiterate, *r* is defined as for each time the *cumulative amount of work performed on AI software R&D* doubles, how many times does *AI software capacity* double. Our toy model is also constructed to be consistent with this general relationship.

We're formulating the relationship between these factors in this manner – where each doubling in cumulative work leads to a set number of doublings in software capacity – due to some amount of both theoretical reasoning and indirect empirical evidence. This relationship has theoretical appeal – it captures the dynamics that: a) easier ideas will get discovered first, meaning ideas will get "harder to find" over time (and hence, consistent progress will require increased growth in cumulative work – e.g., a doubling in cumulative R&D); b) each idea will improve software by some proportional amount (e.g., a 2x efficiency improvement from a particular algorithmic advance), implying that a constant stream of ideas would cash out as exponential growth in software capacity (e.g., a 12-month doubling time); and c) there's no reason to suspect these two rates of growth would be equivalent, so we can allow for a doubling in one to correspond to several doublings in the other. There is also further theoretical reasoning for expecting consistent progress to require cumulative work to grow more-or-less exponentially, specifically (like in our formulation), as opposed to following some other increasing function – if an area of work has many problems or directions which can be pursued independently (like R&D in a field), where the difficulty of solving each individual problem varies over orders of magnitude, then we're likely to see work within the area yield logarithmic returns.

Empirically, many technologies exhibit a relationship whereby exponential increases in cumulative production correspond to exponential declines (at a different rate) in the cost per item, often referred to as experience curve effects. These experience curve effects are somewhat different than what we're dealing with here, as they're concerned not with exponential increases in cumulative R&D, but instead with exponential increases in cumulative volume of production, and the effect there is often thought to be driven primarily through worker and organizational "learning by doing" as opposed to via explicit R&D. That said, the underlying principle is likely similar – exponential increases in the cumulative "inputs" to technological improvement (whether R&D or hands-on practice) lead to exponential improvements in (at least the cost-efficiency of) the technology. These experience curve effects show surprisingly large applicability, spanning industries such as aerospace, shipbuilding, construction operations, and so on (with different industries or technologies seeing different rates of cost declines). In some instances, the experience curve relationship holds smoothly for multiple decades and across several orders of magnitude in both cumulative production and in cost-efficiency, as it has for both transistors and solar panels – though for those two particular technologies, R&D

spending has increased exponentially over time as well, so they may provide even more direct evidence for the relationship we're using.

The economy as a whole provides another data point supporting this general relationship. Long-term, per capita economic growth is thought to be driven primarily through technological advancement. And there has been relatively fast exponential growth in efforts to advance technology, stretching over many decades (e.g., long-term trends of exponential growth in R&D spending, in the number of STEM workers, in the number of scientific publications, etc). Per-capita GDP, meanwhile, has been growing at a slower exponential rate. So here we see exponential increases in R&D efforts yielding exponential increases in the outputs of those efforts, though at very different rates, consistent with what the theory would predict. Indeed, the semi-endogenous growth model, an economics model which bakes in similar assumptions to the above, is popular among economists and can explain why economic growth stayed roughly constant over the 20th century as the number of researchers grew exponentially (many alternative growth models struggle to explain this fact). The semi-endogenous growth model has been used to model the pace of progress in many particular technological areas, in *Are Ideas Getting Harder to Find?*

Finally, there's somewhat more direct evidence for this sort of relationship applying to AI, specifically. As hinted at in above sections, AI systems in various domains have seen their efficiency grow approximately exponentially over time (e.g., for image recognition systems and LLMs), as the number of AI researchers and funding for AI research have also grown more-or-less exponentially in time, at different rates.

With all that said, it's certainly possible that this formulation does not capture the main features of the dynamics. Most of the above reasoning and evidence is either speculative or indirect, and it's plausible that this will all turn out to be an artifact, or that some other relationship would better capture the dynamics. We believe this formulation is substantially more plausible than major existing alternatives, and also that it's likely good enough to work with, but it's still one of the more plausible areas where mistaken assumptions could turn the analysis in this piece on its head. Further research into the relevant dynamics is warranted.