Memory and FLOP/S Hardware Limits to Prevent AGI?

Joshua J. Cogliati*

October 27, 2025

Abstract

Existing discussions of AGI safety have primarily involved preventing dangerous programs from running on computers. This article focuses instead on preventing independence gaining AGI from running based on hardware memory and floating point operations per second limits. We show that a 64 KiB memory and storage limit can be used make an independence gaining AGI practically impossible and show that higher limits are likely possible. These limits are substantially below what is required for current state of the art AI, but the state of the art is expected to advance, so future limits are useful for longer term planning.

1 Introduction

Stuart Russell has discussed methods of having safe AI software (Russell, 2019) and proposed in an interview (Chia and Cianciolo, 2023) "we need to ensure that the hardware and the operating system won't run anything unless it knows that it's safe." For sufficiently powerful computers (such as ones that could effectively out think an entire university), this requires fully understanding the software runs on the computer. Restricting the software to software that has been formally verified to be safe is one way to demonstrate safety. However, this paper will show that if the computational space and speed of the hardware is sufficiently limited, the software can be unrestricted. The threat model is that either intentionally or accidentally a human will create an AI program that is sufficiently intelligent to be able to gain independence, such as by creating a self replicating computer capable of obtaining energy and other things needed to achieve goals without any further assistance from humans. Note that some AI techniques and algorithms are well understood and probably could be proved to be safe even when run on extremely powerful computers including minimax search with a fixed game of GO evaluation function and a climate general circulation model.

2 Definitions

For this paper, the definition of Artificial General Intelligence (AGI) is artificial intelligence that is capable of performing any scientific, technological, engineering or mathematical (STEM) task that a human could do that is needed to gain independence. For example, an independence gaining AGI connected to today's internet might complete intellectual tasks for money and then use the money to mail order printed circuit boards and other hardware. An independence gaining AGI with access to 1800s level technology might mine coal and build a steam engine to power a Babbage like computer and then bootstrap to faster computing elements. An independence gaining AGI on Earth's moon might be able to produce solar panels and CPUs from the elements in the moon's crust, and produce an electromagnetic rail to launch probes off the moon. So an independence gaining AGI can use knowledge about the world the AGI is in to design ways to scale up computational capacity. Note that independence gaining AGI is a subset of general AGI, so if an independence gaining AGI is prevented, that also prevents the less narrow AGI.

Super-intelligence (ASI) is harder to define, but a working definition is that a super-intelligence AGI would be capable of out thinking an entire university or research laboratory for any STEM task necessary to gain independence; alternatively this is the intelligence of approximately a thousand trained people working together. For this definition, the intelligence comes from the people in the university or research laboratory, not the electronic computing hardware, otherwise the floating point operations per second would be primarily from the computers there. For this definition a university or research laboratory in roughly 1940 or before would automatically fit it, but for more modern ones, the computation from the computers would have to be subtracted off to get the intelligence from just the humans. There are two

*mailto:cogljosh@isu.edu

reasons for the "gain independence" limitation to be included. The first is to prevent needing to simulate human brains, for which humans might have an inherent advantage. The second is that if the AGI has the ability and goal to "gain independence" this is sufficient to be dangerous if the AGI is not aligned with human goals and ethics.

This paper is concerned with an AGI that is capable of achieving independence. There are three basic ways that an AGI could use to achieve independence. The three are convincing humans to help, creating hardware in the environment, or expanding into other computer infrastructure. Expanding into other computer infrastructure is already something that has been done by computer viruses for decades, and may lead to the AGI gaining other resources which can be used for one of the other methods to achieve independence. Computer virus can be written in 10s to 100s of instructions, so preventing this is the computer security problem of securing potential targets and in many cases can be solved by shutting down the infected computers. A computer virus is completely dependent on the computers that it runs on and so is not physically independent, and while escaping to other computers to gain resources or to computers with more access to the physical world might be an important step to a computer gaining physical independence, this paper leaves that discussion and research to others.

This paper is primarily concerned with the other two methods. Convincing humans probably requires at least some level of fluency in language and some understanding of how to motivate or negotiate with humans. Creating hardware in the environment requires both some knowledge of the environment and some ability to simulate it. The method this paper uses to show that it is practically impossible for an AGI to gain independence is to show that the available computing power does not allow fluent language and does not allow sufficient simulations.

3 AGI Limits

Having hardware limits for independence gaining AGI and superintelligent AGI would be useful because these help assess the risk of running an experimental piece of software. If the software is run on a higher powered computer, then we need to show the software is safe. On lower powered computers the risk is reduced. Running software on a lower powered computer may require more programming effort or not be possible. Note that the AGI definition does not include any speed or timing considerations, only the super-intelligence definition includes speed. This is intentional, since an AGI that gains independence slower than a human would in similar circumstances, still gains independence.

It is worth noting that the Halting Problem and Rice's theorem are for Turing machines with an infinite tape; this paper is dealing with machines with finite space (memory + disk storage), so those theorems do not demonstrate that finding a hardware limit for having the property of being able to run an independence gaining AGI is impossible.

Estimates of when sufficient compute is available for simulating a human brain have been done such as 1 exa FLOP/S in Chen et al. (2019). Computer transistors and brain neurons are both capable of remembering information and boolean logic, however, using transistors to simulate brain neurons performing a task will be less efficient than directly performing the task[1] on the computer transistors. 1 exa FLOP/S is likely a much more powerful computer than will be needed for AGI (with the possible exception of tasks that require a deep understanding of human brains). The inefficiencies of simulating neurons with transistors has been pointed out in Byrnes (2023). A literature review did not find many existing estimates for a limit below which an AGI is not possible. One estimate is that a human level AGI could be done on an Intel 286 if the programmer is a superintelligent AGI or a "home computer from 1995" (which roughly corresponds to a 90 MHz Pentium) if the programmer is a human (Yudkowsky, 2022) but no method for how this estimate was calculated is provided which greatly limits the use of this estimate. Byrnes (2023) estimated that 10^{14} FLOP/S[2] and 16 GiB of RAM would likely (75%) be sufficient for human-level human-speed AGI. This was calculated by estimating that the amount of information a human brain stores is possibly less than 10 GiB and probably less than 100 GiB. Combining that memory estimate with an estimation that humans are likely memory-bandwidth limited led Byrnes to a conclusion that 10^{14} FLOP/S is capable of processing all the memory-bandwidth. Byrnes estimated a sufficient level for human-level AGI and did not estimate a lower limit since his estimate was already at the level of currently available high-end gaming PCs.

[1] Unless the task is simulating neurons.

[2] For comparison, an NVIDIA RTX 4090 retail gaming GPU from 2022 has 8.3e13 FLOP/S.

4 Limiting Estimates

This section will show that an AGI restricted to 64 KiB of readable and writable space (RAM and rewritable drives, with no read only storage) is very unlikely to be capable of gaining independence. Being fluent at human language or the ability to simulate the hardware needed to gain independence almost certainly will require more than 64 KiB of space. This section will also discuss requirements for this restriction to hold, and IO restrictions and which types of IO might be allowed. This section will also discuss mixtures of read only, write once/read many and read/writable storage.

Fluent English language understanding and creation appears practically impossible in 64 KiB. A typical human vocabulary of 20,000 words would not fit in this amount of space without compression[3] and adding definitions would increase the size even more, so it would not be possible to fit a program that used unrestricted language fluently. This lack of ability means that programs on the computer would be less able to ask for help from humans, and also less able to make use of English documents to understand more about the world.

A 64 KiB computer is unlikely to be able to simulate enough of the world to design hardware to gain independence. For example, a molecular simulation that used 4 bytes per float, 3 vectors of 3 floats, and a beginning and end state could not fit a simulation of a 10x10x10 cube in 64 KiB.[4] Another reference point is P16C5x microcontroller (an LGPL version of the PIC16C5x microcontroller), which is described by 19853 bytes of Verilog source code.[5] This microcontroller is a very reduced instruction set computer and does not include features like multiplication. A full hardware description would also need to describe how to make the hardware like registers and logic functions, as well as wire routing information and so would be significantly larger. In addition, to run a computer also requires energy and matter (otherwise, the computer will still be dependent on humans), which will require more parts.[6] Digital parts are made from analog components, and analog components need to be simulated with a sufficiently accurate physical model.[7] Fitting simulations into 64 KiB requires reducing the number of elements in a molecular simulation or a finite element simulation, carefully choosing the provided parameters to fit what is needed, and getting this wrong results in faulty simulations. Simulating an independence gaining computer would likely require creating small memory models of all necessary components, and having enough room left over to design the rest of the computational and support equipment. From this, we see that holding the design of even a very simple computer and support components would require compression and other tricks to work in 64 KiB. An AI project is unlikely to have the needed electronic component data, and very unlikely to be able to design the needed components from scratch in 64 KiB, so this method of AGI independence gaining also appears to be practically impossible in 64 KiB.

So a 64 KiB computer can do neither fluent language nor complicated simulations. Note that this is not an exact calculation (64.5 KiB for example is also limited), nor is this a proof.

There are several clarifications that need to be made about this 64 KiB limit. Networking together multiple 64 KiB computers would allow the memory space to be increased, which could allow AGI to be done on the combined computers by effectively creating a computer with larger memory space. Even without an intentional network, there can be side channels that transmit and receive data, such as with radio waves, so fully enforcing the limit requires preventing all networking. If time for computation is ignored (as it is for this paper's definition of independence gaining AGI) it does not matter if the storage is RAM, floppy drive, hard drive or flash drive; these all increase capabilities. Register or vector storage on the CPU needs to be counted as well, however,

[3] `https://www.mit.edu/~ecprice/wordlist.10000` for example is 75880 bytes. Those 10,000 words can be compressed down to 27848 bytes using `gzip -9` (version 1.13). This is a word list, not words with the meaning. Word vectors, which do have some meaning attached, usually have vector length of at least 100 (Pennington et al., 2014), so 64 KiB would not even fit a 1000 basic words with the vectors. However other architectures that do not use word vectors are possible, so this is not definitive.

[4] 4 bytes/float $*$ 3 floats/vector $*$ 3 vectors/molecule $=$ 36 bytes/molecule. The simulation will either require keeping two states, or keeping a state and the state delta, so this doubles the ram. So $10*10*10*2*36 = 72000$ which is more than 64 KiB. For reference, a silicon atom is about 0.2 nanometers, transistors currently are on the order of 5 nanometers, so simulating a transistor at the atomic level could require simulating a cube of $25*25*25 = 15625$ atoms, and simulating the quantum effects needed to determine the electronic band structure is more complicated than a simple molecular simulation.

[5] `https://opencores.org/projects/p16c5x` and processed with `cat *.v | sed 's/.*//' | uniq | wc -c` to remove comments and redundant lines. Note that memory representation and Verilog source code can be different.

[6] Note that without fluent language, obtaining the parts from humans would likely be very difficult.

[7] While many simulations can be approximated with less memory, this usually requires having some method to check that the approximation is correct. These include having experimental data to benchmark against, or being able to run a mesh convergence study, which requires being able to run finer meshes then needed.

for CPUs with less than 1 KiB of registers, this will not change the analysis; 65 KiB is still unlikely to be able to do fluent language nor complicated simulations.

One type of IO likely can be allowed, which this paper will refer to as manual W1|R1, which is IO that can either be written once or read once,[8] and rereading or rewriting requires manual intervention. Early computers often had punch card readers and writers.[9] The workflow for writing would be punch cards would come from the punch hopper and be written to and then go to output stacks. Cards to be read would come from a read hopper, and then be read and then go to output stacks. Re-reading a punched card would require manually moving the card from an output stack to the read hopper. An example from early home computers of manual W1|R1 was the use of a tape player,[10] and in order to read the tape, the user would press play on the tape player, and to write would need to press the record button, and to re-read or rewrite would require rewinding and then pressing the needed button. Since the output data cannot be automatically reread in manual W1|R1, this makes using this restricted type of IO impossible to use as a way to extend usable memory without assistance from the human operator or adding additional hardware.

Write once, read many (WORM) media (such as paper tape, punch cards, CD-R or DVD-R) or media where there is manual work needed (such as original cassette drives that required the user to manually switch from reading to recording or UV erasable programmable read only memory (UV-EPROM)) are significantly different than RAM because they can only be written once without intervention. Only writing once is a significant limitation for most uses in simulation or learning algorithms.[11] In addition, if the data cannot be overwritten at the bit level,[12] the data can be read back to see what computational data was being stored, which allows retrospective forensic analysis.

Note that the statement that an AGI restricted to 64 KiB of read and writable space is very unlikely to be capable of gaining independence does not imply that the code to create an independence gaining AGI would require more than 64 KiB. It seems possible that 64 KiB of binary RISC-V RV64GCV machine language code (or similar computers with hardware floating point) would be able to include a machine learning training and running program,[13] and a simple and less efficient[14] simulation of Feynman's classical physics formulation (Feynman et al., 1963, Vol. 2 Table 18-4). Alternatively, the program could fit a mathematical representation of the standard model and general relativity instead, and possibly the code to simulate those equations in 64 KiB. So a small program that is enough to get to a near AGI and a basic understanding of the universe possibly could fit in 64 KiB (or close to it) of code if run on a large and fast enough computer. Note that if we want to create a safe AGI, the code also would need to have the ethics and know how to coexist with current life, which would likely be substantially more complicated;[15] an AGI missing these would likely be a very dangerous AGI. So 64 KiB might be enough to store the binary code for a seed AGI program, however, running the program would require access to more RAM and storage.

5 Nonlimiting Estimates

This section goes through several sample comparisons to give an order of magnitude sense of scale, but none of these provide hard bounds. These include comparison to existing naturally evolved independent life, symbolic AI and self-replicating automaton examples, and modern machine learning codes. These can be compared to the 64 KiB limit estimated in the

[8] Note that most human interface devices such as keyboards and displays are trivially manual W1|R1.

[9] For example, the IBM 1402 Card Read-Punch from the 1960s was a manual W1|R1.

[10] Such as with the Commodore 64.

[11] Substantially more write once/read many storage than R/W storage is needed if a simulation step cannot be fit in the R/W storage. For example, if a simulation needed 64 KiB of data that was updated each timestep and there were a 1000 timesteps, then a computer with 64,000 KiB of WORM drive could do a calculation even without having RAM to store an individual timestep. We can roughly show how this would work for 64 KiB limit of read/write storage with the approximate formula $S + W/m \leq 64$ KiB where S is the amount of read/write storage, W is the amount of WORM space, and m is a heuristic multiple approximately determined by how many times the state will change in search or simulation. In short, more WORM storage may be allowed than read/write storage.

[12] For example, on a paper tape using ASCII, a delete (0b1111111) can overwrite other characters.

[13] A feed-forward multi-layer neural network with back-propagation and stability improvements might be able to be fit into 64 KiB of code and with sufficient parameters in additional memory might be able to do the work of current transformer models.

[14] By choosing simpler algorithms, the compiled code can be smaller, but the run-time speed and run-time memory usage will be worse.

[15] Is the energy being gathered being taken away from an existing ecosystem? Are the atoms being used for construction part of something living? What is life versus non-life? How can I find sentient beings in the neighborhood and communicate with them and understand their values? Those are some of the types of questions that a safely coexisting AGI probably needs to carefully and ethically answer before taking actions.

previous section, and provide reference numbers that suggest a higher limit.

The smallest known cell able to replicate independently in nature is *Pelagibacter ubique* and has a genome with 1,308,759 base pairs (Giovannoni et al., 2005). The largest protein in it is an amino acid sequence of length 7317 (National Center for Biotechnology Information, 2024). Amino acids have between 10 and 27 atoms with an average of 19.2 (Foulquier and Ginestoux, 2001). Designing the sequence requires representing each atom, which if that takes six single precision floating point numbers designing the largest protein would take at least 3 MiB of storage.[16] As each protein is designed, the DNA sequence to create it would need to be stored, so the memory needed will be for the protein currently being designed, plus the DNA encoding all the proteins already designed. So in addition, storing the genome uncompressed (4 pairs per byte) would take another 300 KiB. To the extent that that *P. ubique* is the minimum viable independent organism, designing it gives a lower memory limit of above 3 MiB of storage. Note that this could be both an over estimate or an underestimate. It is possible that substantially smaller replicating organisms could be designed compared to *P. ubique* which would lower the number of atoms to simulate. Actual simulations of quantum electrodynamics are usually more memory intensive than six floating point numbers per atom, and the atoms that the proteins are interacting with also need to be simulated, so substantially more memory might be required than we hypothetically assigned above. Additionally, the design of an independent AGI might be done at the component level, not at the atomic level, so the relevant memory use would depend on the number of different components, their complexity, and how the components' use is described. So if an AGI designs an independently existing machine the design calculations need to be very efficient memory wise, and in some sense simpler than *P. ubique* since these considerations point to needing more than 3 MiB for designing *P. ubique*.

The SHRDLU program was a 1970s natural-language computer program that was only capable of discussing stacking blocks and had a vocabulary of approximately 500 words[17], and it used approximately 450 KiB (100 to 140 K of 36 bit words from the README in Winograd (1972)). Using the SHRDLU program with 500 words per 450 KiB and assuming that the vocabulary is expanded to 5000 words to be more fluent in more topics in English and assuming this is linear on the number of understood words gives an estimate of 4500 KiB of storage needed for fluent English. These are likely false assumptions however. This could be an overestimate if language understanding can be done more efficiently than SHRDLU (such as by more efficient coding, by better than linear scaling (possible if adding new concepts are less work because existing concepts can be leveraged), or fixed costs (such as operating system overhead and common parts such as parsing) become a smaller percentage of the storage needs). Conversely, this can be an underestimate if the concepts in English that SHRDLU interpreted are easier than typical English concepts (which seems likely since the stacking blocks are a physically simple system), and 5000 words is more along the lines of basic proficiency than fluent English, so more vocabulary would be needed. Assuming that improving SHRDLU's efficiency of word representation is reasonably optimal, and that increasing the vocabulary and concepts understood will be linear or more, this suggests a lower-end estimate. This is a back-of-the-envelope calculation deliberately designed to be on the lower end, so an AGI would very likely need much more than approximately 4.5 MiB to be fluent in English.

Another way to get an estimate of the size of data needed for a self-replicating computer is to examine self-replicating computers in simulated environments such as cellular automaton environments. There is a minimum size as stated in a paper by Burks and von Neumann (1987):

> There is a minimum number of parts below which complication is degenerative, in the sense that if one automaton makes another, the second is less complex than the first, but above which it is possible for an automation to construct other automata of equal or higher complexity.

In cellular automaton environments, self-replicating computers have been created. An early example is John von Neumann's universal constructor (von Neumann and Burks, 1966). Devore's self-reproducing automaton is smaller and ran in a world where each cell had 8 possible states and fit into a rectangle of 259 cells by 366 cells (94,794 cells) (Koza, 1994) and so would require about 36 KiB of information uncompressed.[18] Devore's self-reproducing automaton was also computationally

[16] One simple representation of the atoms would be to store 3 floats for the position vector and 3 for the velocity vector. This ignores storing information about the electron's quantum state. For the 6 floats per atom the calculation is 4 bytes/float$*$ 6 floats/atom $*$ 7317 amino acids
$*$ 19.2 atoms/amino acids $*$ $1/1024^2$MiB/byte $\approx$ 3.215 MiB

[17] estimated from counting the DEFS in the file dictio in the source code for SHRDLU

[18] 8 states can be described in 3 bits, so $94794 * (3/8) =$

universal (Turing complete). Note that this does not prove that designing a self-replicating computer requires 36 KiB since there is no proof that Devore's automaton is the minimum.[19] In addition, a self-replicating computer in standard model physics would likely be significantly more complicated, because of requirements such as obtaining energy and the needed atoms, that are present in real world chemistry but are absent in the simple cellular automaton simulations. Creating a self-replicating computer in the real world is both difficult and dangerous, so these toy models are useful for theory, and further research on them potentially provides a possibility of establishing hard limits. If it can be theoretically shown that there is a minimum number of parts for computationally universal automata, this shows that physical computationally universal automata must be at least that complexity since a cellular automaton universal constructor will be simpler than a real-world universal constructor. However, present uncertainties limit the information current cellular automaton theory provides for this paper's estimates for limits for independence-gaining AGIs.

The AlphaFold program (Jumper et al., 2021; Abramson et al., 2024), predicts protein structure, and since predicting protein structure is a key component for designing biological hardware, this can be used for estimating the computation power needed for that. An example computer used to run AlphaFold2 was an Intel Xeon W9-3495X with 56 cores, 512 GB of RAM and 1.92 TB of SSD storage (Exxact Corporation, 2023) which shows that AlphaFold 2 can run on a 2.3 TFLOP/S computer with 0.5 TiB of RAM. Since AlphaFold uses a trained neural network the calculation's prediction can be incorrect and predicting protein structure is only part of what is needed for designing biological hardware, so information about AlphaFold does not prove this computer is sufficient for independence gaining. This does provide an example showing current state of the art computations for biological molecules are substantially above the low end memory requirements estimate for designing *P. ubique* earlier in this section.

For LLM models, the compute used for training them is on the order 10^{20} floating point operations and GiB to TiB of memory. The phi-1 small model (Gunasekar et al., 2023) used 350M parameters and 135 hours of training on an A100 GPU or about 1.3 GiB of RAM and $1.5 * 10^{20}$ floating point operations.[20] The LaMDA model (Thoppilan et al., 2022, Section 10) used $3.55 * 10^{23}$ floating point operations for training a model 137B with parameters or about 0.5 TiB of RAM.[21] Once an LLM is trained, the LLM can be run on typical current computers with sufficient RAM. For example, the gpt-oss-20b model (OpenAI et al., 2025) can be quantized and run on computers with 16 GiB of RAM (Hugging Face, 2025). LLMs were originally designed for next token prediction and language translation (as opposed to reasoning or efficient model representation), and yet often can intelligently answer questions. Assuming that different and more efficient algorithms are possible, the LLM runtime requirements suggest that typical 2025 laptop and desktop computers with 10s of GiB of RAM and 500 GFLOP/S processing might be capable of supporting an independence gaining AGI.

This section has been a brief look at examples for an order of magnitude sense of scale. Further research by subject matter experts on each of these examples could improve the accuracy and precision of the estimates. Primarily from the consideration of *P. ubique* and SHRDLU, it seems likely that a computer with 2 MiB of memory and storage[22] would be unlikely to be able to run an independence gaining AGI.[23] If instead we work down from examination of the state of the art algorithms, RAM and storage limits of 1 GiB and processing power limit of 500 GFLOP/S would prevent many current AI and simulation programs such as LLM training and AlphaFold from running. It is possible lower resource algorithms for an independence gaining AGI exist (and that the lower 2 MiB or 64 KiB limit would be needed to be used to prevent them) and it is possible that an independence gaining AGI would need more than 1 GiB and 500

35547.75 and it is worth noting that there is repetition so compression is possible.

[19] If just replication is needed, and not computational universality, very small models, such as Langton Loops are possible (Langton, 1984).

[20] A Nvidia-A100 GPU has a theoretical computing ability of 312 TFLOP/S (NVIDIA Corporation, 2021) so 135 hours $*$ 312 TFLOP/S $*$ 3600 seconds/hours = 151632000 TFLOP = $1.51632 * 10^{20}$ FLOP. Note that this means that a computer capable of a processing rate of 500 GFLOP/S could have trained the phi-1 small model in under 10 years of total processing: $1.51632 * 10^{20}$ FLOP/(10 years$*$365 days/year$*$24 hours/day$*$ 3600 seconds/hour) $\approx 4.808 * 10^{11}$ FLOP/S As a quick comparison, the 2020 Apple M1 chip used in Mac laptops is capable of 2600 GFLOP/S (which may not be reachable due to bandwidth limitations for LLM training).

[21] 137B parameters or $137 * 10^9$ parameters $*$ 4 bytes/parameter/1024^4 byte/TiB = 0.4984 TiB with the note that this assumes 32 bit floats for the parameters and ignores other storage needed for training.

[22] This is taking the minimum of 3 MiB and 4.5 MiB and rounding down to the nearest power of two

[23] A 1976 Cray I computer had 166 MFLOP/S and 32 MiB of RAM (Patterson and Hennessy, 1998, pg. 43), so computers with more than this amount of RAM have existed for a long time and so at minimum creating an independence gaining AGI on them is hard.

GFLOP/S.

6 Superintelligence Limits?

This section will discuss determining super-intelligence limits with this paper's working definition of super-intelligence (ASI). Unlike the independence-gaining AGI definition, the ASI definition considers if the ASI can think faster than many humans working together on STEM tasks. For some STEM tasks, even 1950s computers are vastly faster than multiple people working together, however, for other scientific and engineering tasks current computers with current software are not capable of matching human performance. This section will also examine the differences between human brains and computer elements, and discuss full human brain emulation, which can provide an upper limit on the needed computing power.

One method of estimating the amount of compute for ASI is to consider the amount of compute needed to simulate human brains at the neuron and synapse level. The amount of computational power to simulate the approximately 100 billion neurons (and roughly 10,000 synapses per neuron) in a human brain is estimated to be approximately 1 exa FLOP/S (10^{18} FLOP/S) (Chen et al., 2019).[24] To the extent this number is accurate, this provides an upper limit for both AGI and superintelligence. Since a human is a general intelligence, then 1 exa FLOP of performance with enough memory for the all synapses (approximately 1 petabyte) would be sufficient for AGI. Similarly, a superintelligence could be created by simulating 10,000 humans, so multiply the AGI limits by 10,000 to get 10^{22} FLOP/S and 10^{19} bytes.[25] Full human brain simulations have not been done, so the previous values are estimates. For *Caenorhabditis elegans*, which has 302 neurons in the adult hermaphrodite, simulations have been created that replicate the behavior of the organism. The BAAIWorm simulation modeled the 136 neurons of the sensory and locomotion functions, the muscle cells and the environment, and replicated behavior such as the organism swimming towards an attractor. It should be noted that the compartmental neural network models used by the BAAIWorm simulation of *C. elegans* (Zhao et al., 2024) likely used substantially more computation and memory per neuron since they simulate each segment of the dendrite. A neuron level simulation estimate gives a rough order of magnitude maximum estimate of what amount of compute is required for human level thinking and beyond human level thinking, but this is not an exact estimate.

This, however, is likely to be an overestimate of the computing power needed for doing a specific intellectual task because of the different characteristics of computers versus human brains.[26] Signals in human neurons travel at about 60 m/s (Stetson et al., 1992) and signal transitions take about 1 millisecond (Kandel et al., 2000, pg. 21). Signals in computers travel at near light speed (2×10^8 m/s) and signal transitions happen on the order of 10^9 times per second. The billions of neurons in human brains often allow the brain to use parallelization when thinking, but the faster signal transition and propagation speed of electronics give significant advantages for algorithms that do not parallelize well.[27]

As an example of the difference, consider the 1964 CDC 6600 supercomputer. It could calculate at a rate of 2 MFLOP/S under optimal conditions[28]. 2 MFLOP/S of computation, even in the case where the algorithm is parallelizable, is well beyond the capability of 1000 humans (2000 floating-point calculations per second is beyond a single human). For many tasks, including visual tasks and muscle control, humans perform much of their thinking subconsciously. However, for some STEM tasks, including arithmetic, humans perform them consciously and so can only perform one at a time. For a certain class of algorithms where a hundred logical or arithmetic operations can do what a human mind can do in a second, a 100 kFLOP/S computer[29] is already at superhuman speeds. This is a rough analogy and would not apply to all types of human thinking, but for

[24] A back-of-the-envelope example calculation of this type is (Kurzweil, 1999, pg 103): "With 100 trillion connections, each computing at 200 calculations per second, we get 20 million billion calculations per second. This is a conservatively high estimate; other estimates are lower by one to three orders of magnitude." (or $2 * 10^{16}$)

[25] Computers can do tricks like copy the full brain quickly, and then start running in parallel that humans cannot do, so 10,000 computer simulations of a human brain could likely do more productive thinking on a specific problem than a collection of 10,000 humans.

[26] As Byrnes (2023) noted, performing AGI by simulating a human brain's neurons is similar to multiplying a number by "do[ing] a transistor-by-transistor simulation of a pocket calculator microcontroller chip, which in turn is multiplying the numbers."

[27] This factor of a million difference in both transmission speed and effective clock speed means that for many cases, the computer can do in an hour something that would take a human over a century.

[28] The CDC 6600 had two multiply units that could multiply a 60 bit floating-point number in 1000 nanoseconds and one floating point add unit that could add a floating point number in 400 nanoseconds (Thornton, 1970, pp. 77 and 88)

[29] 100 kFLOP/S = 100 FLOP/S × 1000, analogous to 100 operations per second × 1000 humans. This does assume the algorithm parallelizes well.

some short (on the order of 10 millisecond) conscious STEM thoughts each thought could be approximated by a small number of computer logical or arithmetic instructions. Proving that all STEM tasks necessary to gain independence require thinking where humans are efficient at thinking would be necessary to prove that 100 kFLOP/S is not ASI. This likely would be difficult to prove, but it does seem unlikely. On the other hand, proving that all STEM tasks necessary to gain independence are not possible at 100 kFLOP/S would also be difficult to prove but seems more likely. So 100 kFLOP/S may not be the absolute lower limit for independence gaining ASI, but it is likely to be roughly in the right order of magnitude. It is possible to argue that the bounds might go lower or higher, but this gives rough bounds between 10^5 FLOP/S and 10^{22} FLOP/S for the computational speed required for independence gaining ASI.

Besides human intelligence, other animals use much smaller brains to interact with the world. The brain of a fruit fly has 139,255 neurons connected by 5×10^7 chemical synapses (Dorkenwald et al., 2024). As an approximation, scaling by the number of synapses would give a simulation computational requirement of 50 GFLOP/S.[30] This amount of computing power is easily available today, for example, a 2010 Intel Core i7-970 can do over 70 giga FLOP/S with a single processor (Intel Corporation, 2024). The fruit fly brain data suggest that it is likely that interacting with the physical world can be done with much less processing power than humans use, and as mentioned before, simulation of a brain will likely be substantially less efficient than rewriting the algorithms for use on computers.

Besides logic and arithmetic, a likely key factor for many STEM tasks is searching through possibilities. One useful analogue for this is the tree search used in chess programs. An Intel 5160 processor (2 cores, 3.00 GHz) with a benchmarked rate of 2 GFLOP/S of computation (PassMark Software, 2025) was used to defeat chess grandmasters (ChessBase, 2006). Human brains are low speed but highly parallel, so the methods that human use to consider chess moves will be different from computer algorithms, but this analogy does suggest that significantly more compute will be needed to match human learning and subsequent search capabilities than is needed to match human arithmetic abilities. At least for present day search algorithms, since this is for a single human, this suggests that if independence gaining ASIs need significant search, they likely need computational abilities at least in the GFLOP/S range.

This section has considered human brains and attempted to provide an estimate of computational speed necessary for independence gaining ASI. Unfortunately, there are significant limitations for all the examples and comparisons used in this section. As well, advances in algorithms can decrease the amount of computation needed for a given task. This section's rough bounds for independence gaining ASI of between 10^5 FLOP/S and 10^{22} FLOP/S for the computational speed required seem likely to include the true limit, but a case can be made for lower and higher numbers.

Table 1: Summary of results, note that all of these are approximate, and see text for important caveats.

Limit Type	Storage	Speed
Lower Estimates	64 KiB AGI	10 kFLOP/S ASI
Likely	2 MiB AGI	1 GFLOP/S ASI
SotA Proxies	1 GiB	500 GFLOP/S
Upper ASI	10^{19} bytes	10^{22} FLOP/S

7 Conclusions

An independence gaining AGI can made practically impossible by restricting all computers to less than 64 KiB of R/W storage without networking. Computer simulations and other uses of computers are very useful for solving other problems of humanity; alternatively, computers below the AGI limit can be used without restrictions, and only run safe software on computers above this limit. 64 KiB of R/W storage is a useful amount computer power and systems like the Commodore 64[31], the Nintendo Entertainment System and Arduino UNO all had 64 KiB or less of R/W storage and these had sales figures in the millions (Amos, 2021; Arduino Team, 2021). This limit is however substantially below almost all modern computing systems, with the notable exceptions of low end embedded systems[32] and retro computing. This paper has shown that it would be possible to allow practically any software, and still prevent independence gaining AGI, so long as there are strict hardware limits instead.

[30] The simulation requirement used in this section for a single synapse in a human brain is roughly 1000 FLOP/S so the 5×10^7 synapses could be simulated by 5×10^{10} FLOP/S (Chen et al., 2019).

[31] Note that a Commodore 64 did not have a built in disk drive. Adding an external disk drive would result in having more than 64 KiB of R/W storage, but a Commodore 64 could be used either stand-alone or with a manually operated cassette tape drive.

[32] For example, the PIC16F13113 chip was introduced in 2023 and has 256 bytes of RAM and 3.5 KiB of Flash (Microchip Technology Inc., 2024).

Determining the threshold computational speed limit for a superintelligent AGI is harder and this paper was not able to determine a good lower limit. Another way to prevent superintelligent AGI is to limit memory at the regular AGI limit.

Note that these are sufficient limits, but they may be far lower than the unknown necessary limits. As seen in Table 1 there is a large range between the lower estimated limits and the upper ASI limits in which the actual limits may exist. Further research on this would be needed before these are useful for setting policy.

8 Speculation and Future Work

Raising the limits from 64 KiB and 10 kFLOP/S seems possible, and would be useful future research. 2 MiB and 1 GFLOP/S probably could be demonstrated for the AGI and superintelligence limits, and would allow more useful unrestricted computers. Research on if and what kind of networking can be allowed would be useful. Research how much Read only, Write only, and Write once/Read many storage can be allowed would be useful.

The limits could be used in at least two ways, one would be to make it an absolute limit and the other to allow computers above the limit, but restrict what software is run on them.

The 64 KiB limit would be very restrictive. If there is no exception for manual W|R or WORM storage, these would not even allow many of the uses that computers were used for in the 1950s.

A 512 KiB computer with 1.5 MiB of read/write storage (including some that is detachable that could be carried to other computers) could be used for many things we currently use computers for including GUI word processing, spreadsheets, a C compiler, and MicroPython programming.[33] This level of computer would however be noticeably less capable than current computers at most tasks.

Limiting to below 1 GiB and 500 GFLOP/S would give computers that would be useful for most things we currently do with computers, with exceptions like high end games, simulations, and of course, many AI techniques. Note that limits to prevent using networking to exceed the limits would probably be quite noticeable at times to people used to current computers and current networks.

Using high powered computers for AI research is in some sense like using a 25 kVolt AC for experiments before fully understanding electricity. It would be much safer to experiment with 3 Volt DC. We need to have a better idea what computational amounts are low enough to be safe and which can lead to accidental AGI creation.

Lastly, there is usefulness in restrictions and regulations even if they are far above the provable limits, since the danger of accidentally creating a non-aligned independence gaining AGI increases as computational power goes up.[34]

This paper has been improved thanks to Elizabeth Cogliati as well as ChatGPT providing "analysis, critique, and identification of flaws or gaps, without suggesting any specific fixes unless asked." (Pre-ChatGPT versions: April 17, 2025 and before are available on researchgate.net)

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok

[33] This is similar to circa 1985 desktop computers such as the Macintosh 512K or an Atari 520ST which had 800 KiB or 720 KiB floppy drives as the read/write storage.

[34] Faster superintelligences can gain independence sooner. Note that there are three relevant hardware limits: L_{IGAGI}, the computing limit for an independence gaining AGI, L_{EIGAGI}, the computing limit for an ethical independence gaining AGI and L_{HAGI}, the computing limit for a human equivalent AGI, with the likely relation $L_{\mathrm{IGAGI}} \leq L_{\mathrm{EIGAGI}} \leq L_{\mathrm{HAGI}}$. There maybe a gap between L_{IGAGI} and L_{EIGAGI} in which case administrative limits set between them might be more dangerous than no limit at all, since only non-ethical independence gaining AGIs can be created. The reason there might be a gap is that it takes a certain amount of intelligence to do things like tell the difference between taking atoms from an inanimate rock versus taking atoms from a living being and putting a solar panel in orbit in a place where it will significantly decrease the light reaching Earth versus an orbit where it will not harm Earth. So setting a limit above L_{HAGI} may not be sufficient, but will not be worse than no limit, but setting a limit below L_{HAGI} but above L_{IGAGI} maybe add risks that would not exist without the limit.

Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, Jun 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Evan Amos. *The Game Console 2.0*. no starch press, 2021. ISBN 978-1-7185-0060-0.

Arduino Team. Introducing the arduino uno mini limited edition: Pre-orders now open, 2021. URL https://blog.arduino.cc/2021/11/24/introducing-the-arduino-uno-mini-limited-edition-pre-orders-now-open/.

Arthur W. Burks and John von Neumann. *Papers of John von Neumann on Computing and Computer Theory*, chapter von Neumann's Self-Reproducing Automata. MIT Press, Cambridge, MA, 1987.

Steve Byrnes. Thoughts on hardware / compute requirements for agi, 2023. URL https://www.alignmentforum.org/posts/LY7rovMiJ4FhHxmH5/thoughts-on-hardware-compute-requirements-for-agi.

Shanyu Chen, Zhipeng He, Xinyin Han, Xiaoyu He, Ruilin Li, Haidong Zhu, Dan Zhao, Chuangchuang Dai, Yu Zhang, Zhonghua Lu, Xuebin Chi, and Beifang Niu. How big data and high-performance computing drive brain science, 2019. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6943776/.

ChessBase. The last man vs machine match?, 2006. URL https://en.chessbase.com/post/the-last-man-vs-machine-match-.

Jessica Chia and Bethany Cianciolo. Opinion: We've reached a turning point with ai, expert says, 2023. URL https://www.cnn.com/2023/05/31/opinions/artificial-intelligence-stuart-russell/index.html.

Sven Dorkenwald, Arie Matsliah, Amy R. Sterling, Philipp Schlegel, Szi-chieh Yu, Claire E. McKellar, Albert Lin, Marta Costa, Katharina Eichler, Yijie Yin, Will Silversmith, Casey Schneider-Mizell, Chris S. Jordan, Derrick Brittain, Akhilesh Halageri, Kai Kuehner, Oluwaseun Ogedengbe, Ryan Morey, Jay Gager, Krzysztof Kruk, Eric Perlman, Runzhe Yang, David Deutsch, Doug Bland, Marissa Sorek, Ran Lu, Thomas Macrina, Kisuk Lee, J. Alexander Bae, Shang Mu, Barak Nehoran, Eric Mitchell, Sergiy Popovych, Jingpeng Wu, Zhen Jia, Manuel A. Castro, Nico Kemnitz, Dodam Ih, Alexander Shakeel Bates, Nils Eckstein, Jan Funke, Forrest Collman, Davi D. Bock, Gregory S. X. E. Jefferis, H. Sebastian Seung, Mala Murthy, Zairene Lenizo, Austin T. Burke, Kyle Patrick Willie, Nikitas Serafetinidis, Nashra Hadjerol, Ryan Willie, Ben Silverman, John Anthony Ocho, Joshua Bañez, Rey Adrian Candilada, Anne Kristiansen, Nelsie Panes, Arti Yadav, Remer Tancontian, Shirleyjoy Serona, Jet Ivan Dolorosa, Kendrick Joules Vinson, Dustin Garner, Regine Salem, Ariel Dagohoy, Jaime Skelton, Mendell Lopez, Laia Serratosa Capdevila, Griffin Badalamente, Thomas Stocks, Anjali Pandey, Darrel Jay Akiatan, James Hebditch, Celia David, Dharini Sapkal, Shaina Mae Monungolh, Varun Sane, Mark Lloyd Pielago, Miguel Albero, Jacquilyn Laude, Márcia dos Santos, Zeba Vohra, Kaiyu Wang, Allien Mae Gogo, Emil Kind, Alvin Josh Mandahay, Chereb Martinez, John David Asis, Chitra Nair, Dhwani Patel, Marchan Manaytay, Imaan F. M. Tamimi, Clyde Angelo Lim, Philip Lenard Ampo, Michelle Darapan Pantujan, Alexandre Javier, Daril Bautista, Rashmita Rana, Jansen Seguido, Bhargavi Parmar, John Clyde Saguimpa, Merlin Moore, Markus William Pleijzier, Mark Larson, Joseph Hsu, Itisha Joshi, Dhara Kakadiya, Amalia Braun, Cathy Pilapil, Marina Gkantia, Kaushik Parmar, Quinn Vanderbeck, Irene Salgarella, Christopher Dunne, Eva Munnelly, Chan Hyuk Kang, Lena Lörsch, Jinmook Lee, Lucia Kmecova, Gizem Sancer, Christa Baker, Jenna Joroff, Steven Calle, Yashvi Patel, Olivia Sato, Siqi Fang, Janice Salocot, Farzaan Salman, Sebastian Molina-Obando, Paul Brooks, Mai Bui, Matthew Lichtenberger, Edward Tamboboy, Katie Molloy, Alexis E. Santana-Cruz, Anthony Hernandez, Seongbong Yu, Arzoo Diwan, Monika Patel, Travis R. Aiken, Sarah Morejohn, Sanna Koskela, Tansy Yang, Daniel Lehmann, Jonas Chojetzki, Sangeeta Sisodiya, Selden Koolman, Philip K. Shiu, Sky Cho, Annika Bast, Brian Reicher, Marlon Blanquart, Lucy Houghton, Hyungjun Choi, Maria Ioannidou, Matt Collie, Joanna Eckhardt, Benjamin Gorko, Li Guo, Zhihao Zheng, Alisa Poh, Marina Lin, István Taisz, Wes Murfin, Álvaro Sanz Díez, Nils Reinhard, Peter Gibb, Nidhi Patel, Sandeep Kumar, Minsik Yun, Megan Wang, Devon Jones, Lucas Encarnacion-Rivera, Annalena Oswald, Akanksha Jadia, Mert Erginkaya, Nik Drummond, Leonie Walter, Ibrahim Tastekin, Xin Zhong, Yuta

Mabuchi, Fernando J. Figueroa Santiago, Urja Verma, Nick Byrne, Edda Kunze, Thomas Crahan, Ryan Margossian, Haein Kim, Iliyan Georgiev, Fabianna Szorenyi, Atsuko Adachi, Benjamin Bargeron, Tomke Stürner, Damian Demarest, Burak Gür, Andrea N. Becker, Robert Turnbull, Ashley Morren, Andrea Sandoval, Anthony Moreno-Sanchez, Diego A. Pacheco, Eleni Samara, Haley Croke, Alexander Thomson, Connor Laughland, Suchetana B. Dutta, Paula Guiomar Alarcón de Antón, Binglin Huang, Patricia Pujols, Isabel Haber, Amanda González-Segarra, Daniel T. Choe, Veronika Lukyanova, Nino Mancini, Zequan Liu, Tatsuo Okubo, Miriam A. Flynn, Gianna Vitelli, Meghan Laturney, Feng Li, Shuo Cao, Carolina Manyari-Diaz, Hyunsoo Yim, Anh Duc Le, Kate Maier, Seungyun Yu, Yeonju Nam, Daniel Bąba, Amanda Abusaif, Audrey Francis, Jesse Gayk, Sommer S. Huntress, Raquel Barajas, Mindy Kim, Xinyue Cui, Gabriella R. Sterne, Anna Li, Keehyun Park, Georgia Dempsey, Alan Mathew, Jinseong Kim, Taewan Kim, Guan-ting Wu, Serene Dhawan, Margarida Brotas, Chenghao Zhang, Shanice Bailey, Alexander Del Toro, Stephan Gerhard, Andrew Champion, David J. Anderson, Rudy Behnia, Salil S. Bidaye, Alexander Borst, Eugenia Chiappe, Kenneth J. Colodner, Andrew Dacks, Barry Dickson, Denise Garcia, Stefanie Hampel, Volker Hartenstein, Bassem Hassan, Charlotte Helfrich-Forster, Wolf Huetteroth, Jinseop Kim, Sung Soo Kim, Young-Joon Kim, Jae Young Kwon, Wei-Chung Lee, Gerit A. Linneweber, Gaby Maimon, Richard Mann, Stéphane Noselli, Michael Pankratz, Lucia Prieto-Godino, Jenny Read, Michael Reiser, Katie von Reyn, Carlos Ribeiro, Kristin Scott, Andrew M. Seeds, Mareike Selcho, Marion Silies, Julie Simpson, Scott Waddell, Mathias F. Wernet, Rachel I. Wilson, Fred W. Wolf, Zepeng Yao, Nilay Yapici, Meet Zandawala, and The FlyWire Consortium. Neuronal wiring diagram of an adult brain. *Nature*, 634(8032):124–138, Oct 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07558-y. URL https://doi.org/10.1038/s41586-024-07558-y.

Exxact Corporation. Alphafold benchmarks and hardware recommendations, 2023. URL https://www.exxactcorp.com/blog/benchmarks/alphafold2-gpu-benchmarks-and-hardware-recommendations.

Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics*. 1963. URL https://www.feynmanlectures.caltech.edu/II_18.html#Ch18-T1.

Elodie Foulquier and Chantal Ginestoux. Imgt aide-mémoire amino acids, 2001. URL https://www.imgt.org/IMGTeducation/Aide-memoire/_UK/aminoacids/abbreviation.html.

Stephen J. Giovannoni, H. James Tripp, Scott Givan, Mircea Podar, Kevin L. Vergin, Damon Baptista, Lisa Bibbs, Jonathan Eads, Toby H. Richardson, Michiel Noordewier, Michael S. Rappé, Jay M. Short, James C. Carrington, and Eric J. Mathur. Genome streamlining in a cosmopolitan oceanic bacterium. *Science*, 309, 2005. URL https://www.science.org/doi/10.1126/science.1114057.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL https://arxiv.org/abs/2306.11644.

Hugging Face. unsloth/gpt-oss-20b-GGUF, 2025. URL https://huggingface.co/unsloth/gpt-oss-20b-GGUF.

Intel Corporation. App metrics for intel® microprocessors revision.06, 2024. https://www.intel.com/content/dam/support/us/en/documents/processors/APP-for-Intel-Core-Processors.pdf and https://www.intel.com/content/dam/support/us/en/documents/processors/APP-for-Intel-Xeon-Processors.pdf.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, Aug 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell, editors. *Principles of Neural Science 4th Ed.* 2000.

John R. Koza. Artificial life: Spontaneous emergence of self-replicating and evolutionary self-improving computer programs. In Christopher G. Langton, editor, *Artificial Life III*, pages 225–262. Addison-Wesley, 1994.

Ray Kurzweil. *The Age of Spiritual Machines*. Penguin Books, 1999.

C. G. Langton. Self-reproduction in cellular automata. *Physica D.*, 1984. URL https://deepblue.lib.umich.edu/bitstream/2027.42/24968/1/0000395.pdf.

Microchip Technology Inc. Pic16f13145 family full-featured 8/14/20-pin microcontrollers, 2024. URL https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/PIC16F13145-Family-Microcontroller-Data-Sheet-DS40002519.pdf.

National Center for Biotechnology Information. Vcbs domain-containing protein [candidatus pelagibacter ubique], 2024. URL https://www.ncbi.nlm.nih.gov/protein/WP_011282049.

NVIDIA Corporation. Nvidia a100 tensor core gpu, 2021. URL https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

PassMark Software. Intel xeon 5160 benchmark, 2025. URL https://www.cpubenchmark.net/cpu.php?cpu=Intel+Xeon+5160+%40+3.00GHz.

David A. Patterson and John L. Hennessy. *Computer Organization and Design 2nd Ed.* 1998.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

Stuart Russell. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking, 2019. ISBN 978-0525558613.

D S Stetson, J W Albers, B A Silverstein, and R A Wolfe. Effects of age, sex, and anthropometric factors on nerve conduction measures. *Muscle Nerve*, 1992.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao,

Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. Lamda: Language models for dialog applications, 2022. URL https://arxiv.org/abs/2201.08239.

J. E. Thornton. *Design of a Computer–The Control Data 6600*. Scott, Foresman and Company, 1970.

John von Neumann and Arthur W. Burks. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.

Terry Winograd. Shrdlu source code, 1972. URL https://web.archive.org/web/20200817093131/http://hci.stanford.edu/winograd/shrdlu/code.tar.

Eliezer Yudkowsky. Ngo and yudkowsky on scientific reasoning and pivotal acts, 2022. URL https://intelligence.org/2022/03/01/ngo-and-yudkowsky-on-scientific-reasoning-and-pivotal-acts/.

Mengdi Zhao, Ning Wang, Xinrui Jiang, Xiaoyang Ma, Haixin Ma, Gan He, Kai Du, Lei Ma, and Tiejun Huang. An integrative data-driven model simulating c. elegans brain, body and environment interactions. *Nature Computational Science*, 4, 2024. URL https://doi.org/10.1038/s43588-024-00738-w.