AGI RESEARCH (1985-2026)

Aaron Turner, 15 September 2026 (updated 2026-11-10)

(note — the exact dates/order of events presented here are almost certainly imperfect)

1. As soon as I started thinking properly about (what we now call) AGI at Sinclair Research Ltd (SRL) in late 1984 / early 1985:
    1. it was immediately obvious to me (a) the extent to which, if taken to its logical conclusion, AGI had the potential to positively transform the world beyond imagination, and (b) that no other technology or project could ever come close to that positive potential; from that moment, the pursuit of AGI became my lifelong obsession
    2. it was always my intention that this would be something that we ultimately built (i.e. not just an intellectual/theoretical exercise) — this means that, for every component, I had to be able to "see" in my mind how it might be implemented
    3. I treated it as a "top-down recursive decomposition" engineering problem ("breadth-first" would be added later)
    4. I started with a blank sheet of paper, and drew a bubble with "AI" in it, plus some sub-bubbles with "?" in them
    5. I treated the overall problem as a perverse jigsaw puzzle (for which you don't have a complete picture of what the final puzzle should look like, and for which you only have an incomplete set of pieces derived from multiple different puzzles); each new sub-bubble (and sub-sub-bubble etc) becomes a piece of the jigsaw puzzle
2. My philosophy in respect of new bubbles/puzzle pieces was always to:
    1. search for any existing ideas/tech that might fully satisfy the logical requirements of the puzzle piece in question
    2. if I could find something already existing that did the job perfectly, then great, I would use that
    3. otherwise I would consider modifying/customising an existing idea in order to create a puzzle piece that was a better/perfect fit
    4. otherwise I would try to create my own puzzle piece from thin air (or, more exactly, by working backwards from the logical requirements)
    5. in other words:
        1. as an engineer (AGI designer), I am completely implementation-neutral — if it works (to my demanding satisfaction) then I will use it, otherwise it stays in the toolbox
        2. I am completely unconstrained by conventional wisdom — I don't use something just because it's considered "cool" or is currently in vogue/fashion or "that's how it's currently done" — it has to meet my strict (albeit subjective) standards as an uncompromised high-quality puzzle piece solution
3. Bootstrapping from 0 to 1 was particularly difficult:
    1. I spent a lot of time introspectively thinking about my own thinking (my navel-gazing phase)
    2. accordingly, I contemplated natural language, meaning (from a comp-sci perspective) and understanding, as well as generic planning and problem-solving
    3. I also borrowed Clive Sinclair's 3-volume copy of "The Handbook of AI", hoping for inspiration
    4. I also (by this point) had free use of SRL's Heffers card, and started buying gazillions of AI-related books (c/o SRL)
    5. … including a couple of massive books on neuroscience (NS); however, I quickly concluded:
        1. NS was such a massive and complex field it would take someone (e.g. me) 50 years to fully understand it
        2. even then, it would still be impossible, even given that knowledge and understanding, to reverse engineer a human-level intelligence, because NS is so incomplete
        3. even if it were possible, human intelligence is severely flawed (not just IMO, there is overwhelming scientific evidence of many such flaws), and any reverse-engineered intelligence would inherit those flaws
        4. thus it's much better (in the long view, which is generally my preference) to treat "intelligence" as a concept in its own right, and to work everything out without resorting to NS
    6. I also appealed to the dictionary definitions of intelligence etc, and followed them to maybe 3-4 levels deep
4. I distinctly remember one other dictionary-related thought experiment:
    1. I imagined getting an electronic copy of an English-language dictionary (very rare in 1985, but I believe a few existed at that time), and using it to bootstrap the system's "knowledge base"
    2. but I then realised that all the definitions would necessarily be circular (otherwise the dictionary would be infinite)

    3. but, if all the definitions were circular, then where was the meaning? (in the sense of being something finite and concrete that you could look up and/or otherwise calculate)
    4. if all the definitions were circular, then any attempt at looking up the “meaning” of something would never bottom out
    5. and so I abandoned that idea (~5 years later, Harnad (1990) would name this the “symbol grounding problem")
5. After about 6 months, my emerging “AGI architecture” comprised four active components (understanding, reasoning, planning, and learning — not all of which I entirely understood at the time, much beyond their name) all connected by a central “knowledge base” — TBH, it's astonishing just how well this architecture has survived the test of time!
6. I immediately realised that the architecture was too large and complex, with too many unknowns, to implement in one go
7. problems with early versions of the Sinclair Spectrum 128 editor then suggested an initial application — automated formal software verification (FSV)
8. I knew absolutely nothing about FSV (or logic/deduction/ATP etc), beyond the high-level concept, and decided to just hit the books some more — but then Aubrey de Grey (Aubs) joined SRL! :-)
    1. amazingly, Aubs agreed with all of my thinking up to that point
    2. Aubs formulated his own ideas for FSV (which I pretended to understand, but didn’t!)
    3. so that’s when Aubs embarked on the development of his FSV tool
    4. and I started the long hard road of trying to catch up with him technically (but I was a long way behind!)
    5. and then the next year (1986), post-SRL’s demise, Aubs and I incorporated Man-Made Minions Ltd (MMM)
9. by 1992, Aubs’ FSV could automatically verify a number of classic, but nevertheless relatively small, algorithms such as bubble sort (including finding all the invariants, and formally proving (via resolution, I believe) all the verification conditions)
    1. when a tentative funding deal with Siemens Plessey Defence Systems fell through, and my (by then, six year) freelance contract at Verran Micro-maintenance / Verran Electronics came to an end, we (i.e. MMM) imploded financially (forcing me to move back to Northampton)
    2. Aubs and I agreed to soldier on (I distinctly remember thinking “death or glory!”), and we swapped roles within MMM
    3. I then embarked on the “formalisation and generalisation” of all of Aubs’ FSV ideas, despite still being way behind technically (albeit slightly less so than was the case 5 years previously)
10. it was not until 1995/96, when I did my year at Oxford, that I finally understood deduction (in FOL/Z, thanks to the lovely Joe Stoy)
11. at that time, I was mostly still thinking about formal knowledge representation (for the architecture’s central “knowledge base”/belief system), plus associated “reasoning” mechanisms (e.g. induction, deduction, and abduction)
    1. after a great deal of thought (much of it extremely confused, as I continued to persevere with my never-ending learning curve), I ultimately gravitated towards first-order NBG set theory (highly expressive, based on FOL which is very widely known, and the lingua franca of all mathematics, also strongly sound and complete, which is a vitally important bonus)
    2. I was also starting to realise that the system would need to be able to reason metamathematically, not just at the object level (as it became increasingly clear to me that a great deal of mathematics occurs at the metamathematics level)
    3. in order to get the system to be able to think metamathematically, I knew that I would need to build a non-trivial library of set-theoretic definitions and theorems (now called Maths4AGI), and that this would require (a) some technically tricky conservative extensions to FOL, and (b) a new custom theorem-prover (for the extended FOL)
    4. at the time, all of these things were right at the limit of my understanding (thankfully rather less so today!)
12. in Dec 2015 (~20 years later!), probably inspired by a conversation with Aubs, I bought my first modern ML book (by Murphy, now a classic) — it was all heavily mathematical, and thus near-impenetrable by me, but nevertheless a solid reference
13. in Jun 2016 I bought Bostrom’s “Superintelligence”, and became aware of the AI/AGI/ASI alignment problem for (I suspect) the first time, which I hadn’t really given much thought previously; I wasn’t overly impressed with Bostrom’s book, but I was at least now aware of the problem; there was a section on Coherent Extrapolated Volition, which I didn’t entirely understand at the time
14. by 2018, I had pretty much worked out the following (at least in my head, but with nothing implemented):
    1. what I then called “inner cognition” — mathematically-founded, comprising an NBG-set-theory-based belief system (formerly, "knowledge base") operated on by three cognitive primitives (deduction, abduction, and induction — although I was still a bit fuzzy on the latter two), and supporting a generic problem-solver (on top of which, in theory, one could then build not just Aubs’ FSV, but also formal program synthesis, and even formal hardware synthesis, both of which would later play vital roles in my AGI construction sequence (as it currently appears in the full Normative ASI Superalignment paper))

    2. I realised that I would need to design an “outer cognition”, built on top of the inner cognition, that interacted with the real world (i.e. as an agent), and which somehow formulated and maintained its internal belief system (which I would later call a Theory-of-the-Universe ToU, or Theory-of-the-Physical-Universe ToPU, depending on context) from those observations
    3. thus I was starting, rather reluctantly, to realise that my emerging “agent” design would somehow need to be aligned
    4. as usual, I had no idea how to design and implement these new puzzle pieces, and so my learning curve continued
15. in early 2018, I wrote “How to Build an Intelligent Machine” (self-published on Amazon), my first attempt at communicating my AGI architecture/ideas to other people; I was so displeased with how it read that I very quickly un-published it from Amazon (with ~10 copies sold)
16. in Apr 2018:
    1. I bought another classic ML textbook (Chris Bishop’s *Pattern Recognition and Machine Learning*), and chewed through it (and the maths) as best I could
    2. I attempted Google’s (very early) “Machine Learning Crash Course”, which at the time was all about NNs, CNNs, etc; the course implicitly assumed a great deal of knowledge that I didn’t have, and so was rather hard going (for me at least), but nevertheless I gleaned the very basics of NNs, gradient descent, SGD etc
17. in mid-2018, I moved back to Cambridge from Northampton (and started attending every AI-related meeting or group I could find!), and in Aug 2018 I reconstituted MMM as BigMother.AI CIC (BMAI)
18. In Dec 2018 I bought the first of many additional books on ML/NNs; all, again, highly mathematical and semi-impenetrable, but nevertheless I gleaned something new from each one, and they all went into my ever growing "AGI toolbox” library — my initial impression, however, was that “iterative function approximation” was not at all what I thought of as “learning”, at least not from an AGI perspective, because the inputs and outputs were all wrong; in my mind, “AGI learning” required an agent to simply observe the universe and, from those observations, to construct/maintain a fully-interpretable internal world model (theory-of-the-physical-universe, ToPU), with a known semantics, but ML/NN didn’t seem to do anything like that
19. accordingly, I spent literally an entire year (probably all of 2019) imagining that I was a (tabula rasa) robot observing the physical universe for the first time, and trying to spot all the (multimodal) patterns, patterns of patterns, patterns of patterns of patterns etc that ultimately everything must boil down to (hint: don’t attempt this while driving!)
20. in May 2019 I bought my first book on the mathematical Pattern Theory developed at Brown University from the 1960s onwards — I believe that this theory (or some derivation thereof) will be the key to a proper, mathematically-well-founded theory (and ultimately implementation, within an AGI) of induction (that is, pattern discovery); it’s highly mathematical (probability theory, topology, etc), and was again initially impenetrable (to me); nevertheless, it’s now firmly in my "AGI toolbox"
21. before the pandemic, I tried giving a few talks in Cambridge, describing my AGI ideas, but the audiences dwindled from ~20 people to ~3 people (2 of whom were friends of mine!), so it seemed pointless to try to engage with the AI field that way
22. in Feb 2020 I joined the AI Safety Reading Group at Cambridge University (Neel Nanda was also a member)
    1. everybody in the group was at least 10x smarter than me; this is pretty much what I had come to expect at any talk/meeting of Cambridge AI researchers (I eventually got used to it!)
    2. I have a box file containing all the papers that we studied as a group
    3. it quickly became apparent to me that:
        1. 98% of the AI safety/alignment literature that the reading group studied implicitly assumed NNs as the implementation substrate; a few papers briefly suggested that other implementation paradigms/architectures might be possible, but quickly reverted to NNs — to all intents and purposes, by 2020 the AI field had become a NN echo chamber (but no-one who is trapped in an echo chamber ever realises that they are trapped in an echo chamber!)
        2. the choice of NNs as an implementation paradigm is actually the root source of many alignment failure cases, but it’s virtually impossible these days to get anyone (under 50, at least) to consider building AI/AGI in any other (non-NN-based) way (6 years on, of course, and everyone is now obsessed with NN-based LLMs)
        3. the majority of the alignment failure cases (that the reading group studied) seemed to boil down to 2 things:
            1. either the (implicit or explicit) final goal was dumb (e.g. ridiculously simple, and thereby bound to fail)
            2. or the AI system (e.g. robot) in question was dumb (as in “stupid robot did something stupid")
        4. which raised the question (at least to me): what if neither the final goal nor the AI system was dumb?
    4. LLMs hadn’t really made an impact at that point, so the prevailing wisdom within the group was that AGI would be built using RL

5. when I asserted that I knew (more or less) how to build an AGI, they all just looked at the floor like I was crazy
6. when I suggested to an extremely smart guy sitting next to me (probably a CU student/PhD) the possibility of building an AI system that was sufficiently smart and pre-educated such that it knew as much about AI safety / alignment / ethics as any AI researcher in the world (including him) and would therefore know what not to do (in terms of behaving in accordance with human preferences) his eyes were like a deer in the headlights as he realised (?) he couldn't fault it
7. it was at this point that I realised I might be onto something, and I started formulating what is now TTQ (although, tongue-in-cheek, I initially called it "Turner's Three Laws"!), and posted it to the BMAI website — it then took another 6 years or so to painstakingly refine TTQ (and ultimately the TTQ+NAP combo) into its current version
8. At some point in 2020, soon after they were published, I purchased LessWrong's series of 5 booklets entitled "A Map That Reflects the Territory"; I can't say that I read every word, but I did selectively skim them

23. in Aug 2020 I started working on the SILDX system for detecting anomalies in industrial X-ray images using deep learning; this is where I started to gain hands-on knowledge and experience of ML using NNs; TBH, although it might seem cool from a young programmer's perspective to be able to "grow" capability just from data plus a NN architecture plus a loss/objective function, from an engineering perspective it's all very hit and miss — no-one really knows how they work, so when they don't (work) the only way to fix them is to try various things at random (e.g. changing a hyperparameter from 12 to 16) until they (magically) start working (or at least seem to be working, for some finite series of tests), but you never really know why; also, you don't so much "train" them to do what you want, it's more like "teasing" them (e.g. by tweaking the loss function parameters); coming from a formal verification background, it's extremely low precision engineering, clearly near impossible to engineer/align in any precise/reliable way, and virtually impossible to verify/validate to any serious certification criteria
24. at some point, I'm not sure exactly when, I formulated my "Babysitter Analogy" — no-one "grows" a babysitter complete with built-in babysitting instructions; instead, people write/imagine the instructions first, then they work out what qualities a babysitter will need to possess in order to understand/execute those instructions, then they find a babysitter matching those requirements; my approach (novel or not) was to treat alignment in the same way — first write the instructions/final goal (the TRUE final goal, not a proxy), then work backwards to determine the properties (eventually called the Outer Alignment Precondition, but now called the Normative Alignment Precondition) that a "non-agentic goal-less precursor AGI" would need to have in order to be able to comprehend and faithfully execute that final goal, then build the precursor to that specification, and then, as a final step, communicate the final goal (as a message) to the goal-less precursor; I then wrote it up accordingly on the BMAI website
25. in the UK, the first COVID lockdown came into force in March 2020 (with a second lockdown starting in November), at which point it became impossible to do any further in-person AGI talks; I tried making some AGI videos, but they didn't really work; in Nov 2020, I started using Overleaf (at Aubs' suggestion), so this must be roughly when I abandoned trying to give talks / make videos and started trying to communicate my AGI thoughts as an "academic" paper (i.e. v001 — we're now on v253)
26. at first, I didn't really know how to focus/organise the paper (because basically I had ~35 years of ideas to get out of my head); it started out as "the BigMother Manifesto" to design and build an AGI/ASI — only when I realised that this would take an indefinitely-long series of papers did I gradually pivot to focus primarily on "Outer Alignment" (subsequently renamed "Normative ASI Superalignment") for this first paper (which would necessarily include definitions etc for the entire future series of papers)
27. sometime in 2021 (I believe), Professor Emeritus Leslie Smith at Stirling University (and associate editor at the Journal of Artificial General Intelligence) discovered the BMAI website, liked the babysitter analogy in particular (for some reason), and gradually became my de facto mentor/rabbi — he has since read multiple versions of the paper, assures me that it's "special", and now suggests that (if it proves impossible to publish in a peer-reviewed journal, as is likely to be the case due to (a) its multidisciplinary nature, and (b) its monograph-like length) I should submit it to e.g. Springer/MIT Press/etc as a book.
28. by 2026, the paper had ballooned to ~300-pages, making it essentially non-publishable; I therefore decided to mothball and then cannibalise it, thereby generating a stream of smaller, more publishable papers – the first such "short" paper was entitled "An Implementation-Neutral Solution to the ASI Superalignment Problem", first published to Zenodo (as a preprint) on 2026-11-10