5 fold decomp + strategy planner + mermaid diagrams + mcp or and pynusmv to api using mcpo? Bootstrap and 7 step planner.

Learning is simply the process by which an algorithm of time complexity x is translated into one with lower time x-C for some constant C at the expense of some degradation in accuracy of output.

Therefore, on one extreme of the memory/reasoning axis is reasoning which maximises time complexity and minimises memory complexity and on the other side is memory which maximises space complexity and minimises time complexity. Conventional machine earning techniques seek to achieve low time complexity on reasoning tasks which would otherwise require intractably high time complexity by having very high space complexity. This much we know empirically from the success of deep learning approaches which seek to 'brute force' this tradeoff with curve fitting approaches. However, we know that such approaches are immensely compute intensive which is a major issue in the economics of the commercial delivery of AI (LLM) systems.

Therefore, one might consider that learning exists on a spectrum of time/space tradeoff. This much seems somewhat unremarkable until we start to consider a number of open problems in AI problem from the standpoint of a continuous axis between time and space complexity - that is hallucination, escape from training distribution and effective reasoning using ML. As we approach the limits of curve fitting approaches it forces us to ask the question of whether breaking past the compute efficient frontier will need fundamentally different methods as neural scaling laws appear to place a hard fundamental limit on resource bounded optimal learning in deep neural networks. How do we arrive at a conceptual or theoretical framework for breaking through these extremely hard problems.

Therefore, one might fully parameterise and constrain (I.e. make mathematically well defined) the notion of general intelligence as simply the ability to move between time and space complexity, appropriately selecting an algorithm according to the domain specific/task specific need for accuracy. By unifying the notion of algorithm selection along some continuous axis of time/space tradeoff with the notion of AI design, we thereby obtain an overarching framework for understanding technical approaches for training or programming AI to select between different technical approaches based on. In fact, in selecting between whether you train or program yhis algorithm selecting process, we can again apply this heuristic mathematical framework such that where assigning a task to a given algorithm is well defined it is possible to program this whereas when it is not well defined it is better handled through some sort of learning process so this might be 'outsourced' in some sense to the system itself. It is thus that we might arrive at a class of system which remains a tool rather than some autonomous 'AGI' whilst still bypassing many of the commercial issues at plan with the current state of the art.

This is because we obtain a mathematical framework within which to encode the notion of general intelligence. This means that the essence of the next evolution of AI systems can be thought of predominately from the standpoint of 'intelligent routing' whereby the compute resources are scaled and routed to an appropriate algorithmic subsystem based on the classification of the problem across various semantic axes which correspond loosely with domain specific requirements. For example, science requires balanced reasoning and memory, mathematics is far more reasoning heavy. 5 fold semantic decomp, model of models gives us this heuristic for computing an appropriate routing for a given problem using an LLM to precisely calculate the optimal algorithm for a given task, whilst also leveraging MDP bandits and parallel operation of similar tools to allow the system to learn to choose between algorithms. This is all we need to achieve CEF breakthrough/distributional escape/optimal reasoning.

This is a specialised AI designed for building narrow AI solutions tailored to the user's primary pain point!1. Evals pipeline under tasks for human to review. CSF framework?

2. Formal methods tools (cat.jl + python bindings - can use MCPs from before? Sagemath, pynusmv, z3, google OR - can have dual approach for robustness I.e. write code or submit via API - try writing code first and if this fails then try API instead for automatic handing?).

3. Expose agents via API from CLI - have methods for initialising the agent and caching?

4. Alternative RAG pipeline using markitdown and universal RAG chat for agentic RAG on custom docs?

5. Test out memory, eurisko and experta. Write down tests/come up with automated tests? Fix missing deps and test + add tools for coding etc.

6. Examples to formal spec pipeline by generating informal examples and analysing an abstraction of this to find commonality and come up with a model.

7. Bootstrap as an API + agentic RAG over bootstrap with research? Global strategy/solver. Tool selection strategy code using hashing?

8. Maths formulae? Pix to latex? Latex to code?

9. Vision interpretation + multimedia gen + blender + other libraries for multimedia? Free vs API

10. Computer use tools? Bytebot or opencua? Dynamic UI by spinning up a react app on local host?