Boundless Socratic Learning with Language Games

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

An agent trained within a closed system can master any desired capability, as long as the following three conditions hold: (a) it receives sufficiently informative and aligned feedback, (b) its coverage of experience/data is broad enough, and (c) it has sufficient capacity and resource. We justify these conditions and consider what limitations arise from (a) and (b) in closed systems, when assuming that (c) is not a bottleneck. Considering the special case of homoiconic agents with matching input and output spaces (namely, language), we argue that such pure recursive self-improvement, dubbed 'Socratic learning,' can boost performance vastly beyond what is present in its initial data or initial knowledge, and is only limited by time, as well as gradual misalignment concerns. Furthermore, we propose a constructive framework to implement it, based the notion of language games.

1 Introduction

On the path between now and artificial superhuman intelligence [ASI; 11] lies a tipping point, namely when the bulk of a system's improvement in capabilities is driven by *itself* instead of human sources of data, labels, or preferences (which can only scale so far). As yet, few systems exhibit such *recursive self-improvement*, so now may be a prudent time to discuss and characterize what it is, and what it entails. We focus on one end of the spectrum, the clearest is not the most practical one, namely pure self-contained settings of '*Socratic*' learning, closed systems without the option to collect new information from the external world. We articulate conditions, pitfalls and upper limits, as well as a concrete path towards them that builds on the notion of language games. The central aim of this brief position paper is to clarify terminology and frame the discussion, with an emphasis on the long run. It is not to propose new algorithms, nor survey past literature; we pay no heed to near-term feasibility or constraints. We start with a flexible and general framing, and refine and instantiate these definitions over the course of the paper.

Definitions Consider a *closed system* (no inputs, no outputs) that evolves over time. Within the system is an entity with inputs and outputs, called *agent*, that also changes over time. External to the system is an *observer* whose purpose is to assess the *performance* of the agent. If performance keeps increasing, we call this system-observer pair an *improvement process*.

The dynamics of this process are driven by both the agent and its surrounding system, but setting clear agent boundaries is required to make evaluation well-defined: in fact an agent *is* what can be unambiguously evaluated. Similarly, for separation of concerns, the observer is deliberately located outside of the system: As the system is closed, the observer's assessment cannot feed back into the system. Hence, the agent's learning feedback must come from system-internal *proxies* such as losses, reward functions, or critics.

The simplest type of performance metric is a *scalar* score that can be measured in finite time, that is, on (an aggregation of) episodic tasks. Mechanistically, the observer can measure performance in two ways, by *passively* observing the agent's behaviour within the system (if all pertinent tasks occur naturally), or by *copy-and-probe* evaluations where it confronts a copy of the agent with interactive tasks of its choosing.

Without loss of generality, we distinguish three types of elements within an agent; *fixed* elements are unaffected by learning, such as its substrate or unmodifiable code (genotype). *Transient* elements do not carry over between episodes, or across to evaluation (e.g., activations, the state of a random number generator). And finally *learned* elements (e.g., weights, parameters, knowledge) change based on a feedback signal, and their evolution maps to performance differences.

We can distinguish improvement processes by their implied lifetime; some are *open-ended* and keep improving without limit [7], while others converge onto their asymptotic performance after some finite time. Note that neither case needs to invoke a notion of optimality.

2 Three Necessary Conditions for Self-improvement

Self-improvement is an improvement process as defined above, but with the additional criterion that the agent's own outputs (actions) influence its future learning. In other words, systems in which agents shape (some of) their own experience stream, potentially enabling unbounded improvement in a closed system. This setting may look familiar to readers from the reinforcement learning community [RL; 18], who build agents whose behaviour changes the data distribution it learns on, which in turn affects its behaviour policy, and so on. Another prototypical instance of a self-improvement process is *self-play*, where the system (often a symmetric game) slots the agent into the roles of both player and opponent, to generate an unlimited experience stream annotated with feedback (who won) that provides direction for ever-increasing skill-learning.

From its connection to RL, we can derive necessary conditions for self-improvement to work, and help clarify some assumptions about the system. The first two conditions, feedback and coverage, are about feasibility in principle, the third (capacity) is about practice.

Feedback Feedback is what gives direction to learning, without it, the process is merely one of self-modification. Feedback must have two properties for self-improvement to work, one fundamental, one practical. First, system-internal feedback must be *aligned* with the external observer, and remain aligned throughout the process. This places a significant burden on the system at set-up time, with the most common pitfall being a poorly designed critic or reward function that becomes exploitable over time, deviating from the observer's intent. RL's famed capability for *self-correction* is not applicable here: what can self-correct is behaviour given feedback, but not feedback itself. Second, the efficiency criterion for feedback is that it be reliable enough, and contain enough information (not too sparse, not too noisy, not too delayed) for learning to be feasible within the time horizon of the system.

Coverage By definition, a self-improving agent determines the distribution of data it learns from. To prevent issues like collapse, drift, exploitation or overfitting, it needs to preserve[1] coverage of the data distribution everywhere the observer cares about. In most interesting cases, where performance includes a notion of generalisation, that target distribution is not given (the test tasks are withheld), so the system needs to be set up to intrinsically seek 'sufficient' coverage, a sub-process classically called *exploration*.

Capacity The research field of RL has produced a lot of detailed knowledge about how to train agents, which algorithms work in which circumstances, an abundance of neat tricks that address practical concerns, as well as theoretical results that characterize convergence, rates of progress, etc. It would be futile to try and summarize such a broad body of work here. However, one general observation that matters for our argument is that 'RL works at scale': in other words, when scaling up experience and compute sufficiently, even relatively straightforward RL algorithms can solve problems previously thought out of reach [high-profile examples include: 19, 10, 15, 16, 21, 1]. For any specific, well-defined practical problem, the details matter (and differ), and greatly impact the efficiency of the learning dynamics; but the asymptotic outcome seems a foregone conclusion.

[1] This may entail conditions on how the system is initialised, as the agent needs to see a first set of inputs before it can produce its own.

3 Socratic Learning

The specific type of self-improvement process we consider here is *recursive self-improvement*, where the agent's inputs and outputs are *compatible* (i.e., live in the same space), and outputs become future inputs.[2]This is more restrictive but less mediated than the general case where outputs merely influence the input distribution (but it is less restrictive than homoiconic self-modification and self-referential systems).This type of recursion is an attribute of many open-ended processes, and open-ended improvement is arguably a central feature of ASI [see 7].

An excellent example of such a compatible space of inputs and outputs is *language*. A vast range of human behaviours are mediated by, and well-expressed[3] in language, especially in cognitive domains (which are definitionally part of ASI). As argued by [4], language may well be sufficient for thinking and understanding, and not require sensory grounding. Plus, language has the neat property of being a *soup of abstractions*, encoding many levels of the conceptual hierarchy in a shared space. A related feature of language is its extendability, i.e., developing new languages within it, such as formal mathematics or programming languages. While special-purpose tools for these are important for efficiency, natural language may be sufficient as a basis: just like humans can reason 'manually' through mathematical expressions when doing mental arithmetic, so can natural language agents [12]. And of course, it does not hurt that AI competence on language domains has radically improved recently, with a lot of momentum since the rise of LLMs.

For the remainder of the paper, we will use '*Socratic learning*' to refer to a recursive self-improvement process that operates in language space. The name is alluding to Socrates' approach of finding or refining knowledge through questioning dialogue and repeated language interactions, but, notably, without going out to collect observations in the real world—mirroring our emphasis on the system being closed. We encourage the reader to imagine an unbroken process of deliberation among a circle of philosophers, maybe starting with Socrates and his disciples, but expanding and continuing undisturbed for millennia: what cultural artifacts, what knowledge, what wisdom could such a process have produced by now? And then, consider a question that seems paradoxical at first: how can a closed system produce open-ended improvement?

The Limits of Socratic Learning

Revisiting the necessary conditions for self-improvement, we can derive some insights on how Socratic learning is limited *in principle*. For that, we can mostly sidestep the capacity concerns of Section 2, by choosing one of two premises. Either, we can assume that compute and memory constraints are but a temporary obstacle, as they keep growing exponentially, so ignoring them still produces valid high-level insights. Or, we can consider the resource-constrained scenario and study feasibility within the class of such restricted systems. The other two conditions, coverage and feedback, remain irreducible however. The system has to keep generating (language) data, while preserving or expanding diversity over time. In the LLM age, we can envision a generative agent initialized with a very broad internet-like distribution, but preventing drift, collapse or just narrowing of that distribution in a recursive process may be highly non-trivial [14].

The other requirement is for the system to continue producing feedback about (some subset of) the agent's outputs, which structurally requires a critic that can assess language, and that remains sufficiently aligned with the observer's evaluation metric. This is challenging for a number of reasons: Well-defined, grounded metrics in language space are often limited to narrow tasks, while more general-purpose mechanisms like AI-feedback are exploitable, especially so if the input distribution is permitted to shift. For example, none of the current LLM training paradigms have a feedback mechanism that is sufficient for Socratic learning. Next-token prediction loss is grounded, but insufficiently aligned with downstream usage, and unable to extrapolate beyond the training data. Human preferences are aligned by definition, but prevent learning in a closed system. Caching such preferences into a learned reward model makes it self-contained, but exploitable, and misaligned in the long-run, as well as weak on out-of-distribution data.

[2]Or at least some of them are fed back. Input and output spaces are not necessarily identical, but they intersect. For example, the agent could be generating code, but perceive natural language, (partly self-generated) code, and execution traces.

[3]"Whereof one cannot speak, thereof one must be silent." [23]

4 Language Games Are All You Need ...

Fortunately, language, learning and grounding are well-studied topics. A particularly useful concept for us to draw on is Wittgenstein's notion of *language games*.[4] For him, it is not the words that capture meaning, but only the interactive nature of language can do so. To be concrete here, define a language game as an *interaction protocol* (a set of rules, expressible in code) that specifies the interaction of one or more agents ('players') that have language inputs and language outputs, plus a scalar *scoring function* for each player at the end of the game.[5]

Language games, thus defined, address the primary needs of Socratic learning; namely, they provide a scalable mechanism for unbounded interactive data generation and self-play, while automatically providing an accompanying feedback signal (the score). In fact, they are the logical consequence of the coverage and feedback conditions, almost tautologically so: there is no form of interactive data generation with tractable feedback that is not a language game. As a bonus, seeing the process as one of *game-play* immediately brings in the potential of rich strategic diversity arising from multi-agent dynamics [as spelled out in depth in 8, 6], which is likely to address at least part of the coverage condition. Pragmatically too, games are a great way to get started, given the vast human track record of creating and honing a vast range of games and player skills [3]. A number of common LLM interaction paradigms are also well represented as language games, for example debate [9, 5], role-play [20], jailbreak defense [25], or outside of closed systems, paradigms like RL from human feedback [RLHF, 13, 2].

... If You Have Enough of Them ...

Returning to our circle of deliberating philosophers: is there any *one* language game we could imagine them playing for millennia? Instead, maybe, they are more likely to escape a narrow outcome when playing *many* language games. It turns out that Wittgenstein (him again) proposed this same idea: he adamantly argued against language having a singular essence or function.[6] Using many narrow but well-defined language games instead of a single universal one resolves a key dilemma: For each narrow game, a reliable score function (or critic) can be designed, whereas getting the single universal one right is more elusive [even if possible in principle, as argued by 17].[7] From that lens, the full process of Socratic learning is then a *meta-game*, which schedules the language games that the agent plays and learns from.

... And You Play the Right Ones

Socrates was famously sentenced to death and executed for 'corrupting the youth.' We can take this as a hint that a Socratic process is not guaranteed to remain aligned with external observers' intent. Language games as a mechanism do not side-step this either, but they arguably reduce the precision needed: instead of a critic that is aligned at the fine granularity of individual inputs and outputs, all that is needed is a 'meta-critic' that can judge which games should be played: it may be that no individual language game is perfectly aligned, but it is doable to filter the many games according to whether they make a net-positive contribution (when played and learned about). This kind of structural leniency is precisely what gives it the potential to scale.

Stepping out of our assumption of the closed system for a moment: when we actually build ASI, we will almost surely want to not optimistically trust that alignment is preserved, but instead continually check the process as carefully as possible, and probably intervene and adjust throughout the training process. In that case, explicitly exposing the distribution of games (accompanied with per-game learning curves) as knobs to the designer may be a useful level of abstraction.

[4] "I shall also call the whole, consisting of language and the actions into which it is woven, the 'language-game'." [24]

[5] For simplicity, assume that games are guaranteed to terminate in finite time.

[6] "But how many kinds of sentence are there? Say assertion, question, and command?—–There are *countless* kinds: countless different kinds of use of what we call 'symbols,' 'words,' 'sentences.' And this multiplicity is not something fixed, given once for all; but new types of language, new language-games, as we may say, come into existence, and others become obsolete and get forgotten." [24], emphasis in original.

[7] But, as a prescient Norbert Wiener was warning seven decades ago: "The machines will do what we ask them to do and not what we ought to ask them to do. [...] We can be humble and live a good life with the aid of the machines, or we can be arrogant and die." [22].

References

[1] T. AlphaProof and T. AlphaGeometry. AI achieves silver-medal standard solving International Mathematical Olympiad problems. *DeepMind blog*, 2024.

[2] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[3] E. Berne. *Games people play: The psychology of human relationships*, volume 2768. Penguin Uk, 1968.

[4] D. J. Chalmers. Does thought require sensory grounding? From pure thinkers to large language models. *arXiv preprint arXiv:2408.09605*, 2024.

[5] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

[6] E. A. Duéñez-Guzmán, S. Sadedin, J. X. Wang, K. R. McKee, and J. Z. Leibo. A social path to human-like artificial intelligence. *Nature Machine Intelligence*, 5(11):1181–1188, 2023.

[7] E. Hughes, M. Dennis, J. Parker-Holder, F. Behbahani, A. Mavalankar, Y. Shi, T. Schaul, and T. Rocktäschel. Open-endedness is essential for artificial superhuman intelligence. *arXiv preprint arXiv:2406.04268*, 2024.

[8] J. Z. Leibo, E. Hughes, M. Lanctot, and T. Graepel. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *arXiv preprint arXiv:1903.00742*, 2019.

[9] T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, Z. Tu, and S. Shi. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[11] M. R. Morris, J. Sohl-Dickstein, N. Fiedel, T. Warkentin, A. Dafoe, A. Faust, C. Farabet, and S. Legg. Levels of AGI: Operationalizing progress on the path to AGI. *arXiv preprint arXiv:2311.02462*, 2023.

[12] T. OpenAI o1. Learning to reason with LLMs. *OpenAI blog*, 2024.

[13] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[14] H. Shi, Z. Xu, H. Wang, W. Qin, W. Wang, Y. Wang, and H. Wang. Continual learning of large language models: A comprehensive survey. *arXiv preprint arXiv:2404.16789*, 2024.

[15] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[16] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

[17] D. Silver, S. Singh, D. Precup, and R. S. Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021.

[18] R. S. Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

[19] G. Tesauro et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[20] A. S. Vezhnevets, J. P. Agapiou, A. Aharon, R. Ziv, J. Matyas, E. A. Duéñez-Guzmán, W. A. Cunningham, S. Osindero, D. Karmon, and J. Z. Leibo. Generative agent-based modeling with actions grounded in physical, social, or digital space using concordia. *arXiv preprint arXiv:2312.03664*, 2023.

[21] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[22] N. Wiener. The machine age / In 1949, he imagined an age of robots. *MIT Archives / The New York Times*, D:8, 1949 / 2013.

[23] L. Wittgenstein. *Tractatus Logico-Philosophicus*. 1921.

[24] L. Wittgenstein. *Philosophical investigations*. 1953.

[25] Y. Zeng, Y. Wu, X. Zhang, H. Wang, and Q. Wu. Autodefense: Multi-agent llm defense against jailbreak attacks. *arXiv preprint arXiv:2403.04783*, 2024.