Developmental Labs Project Whitepaper

Jean-Christophe Baillie

2023

1 Introduction

Deep learning, and more generally differential programming, is today's dominant paradigm in Artificial Intelligence [Bengio et al., 2017]. While it has achieved remarkable successes in difficult tasks, like classification on high dimensional input spaces, or prediction on highly non-linear signals, and also more recently LLMs, it also has shown limitations that are today widely discussed inside and outside of the community of deep learning. Some of the current critics of deep learning highlight weaknesses on topics such as compositional reasoning, explainability, handling of causality (vs statistical correlations), lack of embodiment and non explicit representations with grounding, and more generally rejection of all form of explicit symbolic representation [Mitchell, 2021, Marcus, 2020, Shanahan and Mitchell, 2022]. In fact, the debate is often more generally framed as a debate between symbolic and non-symbolic approaches. In reality, researchers in deep learning are already working on some form of symbol processing via one-hot vectors and disentangled latent representations [Vincent et al., 2008, Hinton and Salakhutdinov, 2006, Goodfellow et al., 2014]. Embodiment is more and more expressed as a requirement, as seen also in recent interactive intelligence experiments done by Deepmind [Abramson et al., 2020] or data collection from first person view embedded video stream by Meta [Grauman et al., 2021]. Questions about causality and world models are also put high on the agenda of the most advanced research in machine learning [Chevalier-Boisvert et al., 2018, Goyal and Bengio, 2020].

When looking more carefully, there is actually something deeper than this surface difference about symbols that is at play here and it has to do with the learning experience itself. There are mainly two axis where we can see a spectrum of variation:

1. How is new information internally modifying the system, in other words: the learning mechanism itself.
2. How is new information effectively acquired and fed into the system.

Let's examine how today's neural network and deep learning approaches position themselves relative to these two questions, and what alternative (and complementary) path can exist, which would lead to fruitful extensions of the current paradigm and potentially help solve the shortcomings that the critics of deep learning have been voicing.

1.1 Differentiability is not all you need

One aspect that characterizes all current research into deep learning inspired methods, not only multilayered networks but all sorts of derived architectures (transformers, RNN, more recently GFlowNet, JEPA, etc), is not the rejection of symbols as we said in introduction, at least not in their emergent form. It is rather the requirement for end-to-end differentiability of the architecture, so that some form of gradient-based method can be applied to learning. It is a very strong constraint applied to the type of solutions that are explored and is presented as the only option if you don't want to do an exhaustive search of the solution space, which obviously would not scale (a critic often directed against classical AI symbolic methods).

The idea of gradient descent mixes two things: gradient, and descent. The "descent" part rightfully argues for a process of small step updates towards improvement of a local solution: indeed, if there is no oracle that could reveal what has to be done, then directly "jumping" to a good solution starting from anywhere is not possible. There is simply no criteria to "calculate" the global jump. All that can be done is try to move progressively towards a better solution, starting from an initial position.

The "gradient" part relates to the the question of how to exactly perform that incremental move. If it is possible to compute a gradient, then this is the optimal choice: simply moving slightly along this gradient in the direction of improvement, recomputing the gradient once arrived at destination and repeating until a (local) minima is reached. There are of course many refinements around this idea (SGD, ADAM, etc), but the core principle is the same: if a gradient is computable, then there is a "local oracle" that can inform on where to go locally to improve the current position.

Actually, the crucial part here is not the "gradient", but it is the "descent", or the recognition that it is needed to move by small increments around the current position (also called "graduate descent"). If there is no gradient available, it is still possible to probe for nearby solutions (at random or with some heuristic) and figure out where to go next in order to improve the current situation by taking the best among the probed locations. Having a gradient is simply more efficient (optimal, in fact), while picking a set of random directions to probe the local landscape and then pick the best bet is the least efficient. And all sort of intermediary positions along this axis can be imagined, if some domain specific bias can be introduced in the probing selection, instead of simply picking randomly.

The typical example of a search using random probing around the current position is evolutionary dynamics. In the case of genes, small moves around a current genome are done when mutations occur, and this constitutes a blind exploration of the solution space around the current position, with a descent method but without a gradient. In general, several locations are explored in parallel to avoid local minima and speed up the search. Evolutionary dynamics in general is applicable to many domains, starting with the evolution of language, and is a successful method to explore certain solution spaces for which no gradient could simply be computed (they are continuous, but non-differentiable spaces).

There are two notable caveats here: first, we still assume that the solution space is somehow continuous, which means that the quality of a solution does not vary abruptly in a small vicinity in the solution space, and it makes sense to probe for local variants. A counter example of this is, for example, hash functions which are designed precisely against this property. If a particular output is targeted for a hash function, then the only option is to randomly try until a match is found (what bitcoin hash challenges are based on). So there is really nothing to "learn" here and asking for continuity, or at least piece-wise continuity, is probably a reasonable constraint to hold in the context of learning. Second, the dimensionality of the solution space could be so large that random search, even simply around a known solution point, is not tractable. This is a typical case somewhere in the middle of the gradient-random axis, where heuristics to guide the search should be introduced. This is, to some extent, what is already done with AlphaGo or similar recent systems when a deep learning oracle is used to guide an otherwise more systematic search. Also, algorithms like CMA-ES effectively compute an approximated gradient from a purely random sampling strategy and would come right in the middle of the axis, as well as what are called generally "black box optimization" algorithms.

So, regarding the kind of learning algorithm used, the core limiting issue with deep learning, or more generally "differentiable programming", is not its rejection of symbols (symbols are somehow handled in an emergent way), but its focus on handling only solution spaces that happen to allow for a gradient computation. There is of course nothing wrong with studying these cases, and whatever the solution to strong AI is, it is reasonable to assume that gradient-based methods are going to be a crucial part of it, but contrary to what is often claimed there is simply no scientific reason to think this should be the only way forward, with a strong rejection of anything not "fully differentiable". The differentiable/non-differentiable axis exists and it is likely that many core problems of AI, like language grounding or even self driving cars, require solutions that are located somewhere along this line. Combining differentiable methods when applicable together with evolutionary dynamics on discrete (=symbolic) structures when appropriate has the potential to yield substantial progress in the field.

There are many examples of interesting coupling between old symbolic approaches and recent neural approaches to complement their respective strength. Neural networks can be used to make non-scalable exhaustive searches tractable via heuristics, act as oracle/predictors in fitness calculation for evolutionary search, implement flexible bi-directional bridges between symbols and the sensorimotor space, enable smooth pattern matching between graph structures (also covered by the recent Graph Neural Network approach) and potentially many other coupling opportunities [Garnelo et al., 2016, Marcus, 2020, Bader and Hitzler, 2005].

The crucial point here is to allow the use of explicit data representation in the final architecture of the system, in a direction that is perhaps more inspired by engineering considerations than neuroscience accuracy. It is very possible that equivalent full-neural end-to-end differentiable implementations could be achieved in principle, but it would come at a price both in computation cost (implementing trivial symbolic operations could require networks with very large numbers of hidden parameters and days of training), and, perhaps more importantly, in research time cost: most of the operations that can be almost immediately implemented in a traditional algorithmic approach are in fact currently whole unsolved research sub-fields of current neural-based AI, because of the added difficulty of trying to implement them through fully differentiable operators.

By contrast to the unified gradient-based approach, having several interacting learning methods, adapted to the particular type of problems at hand (including gradient descent when appropriate), seems to be a more flexible alternative that could help getting results faster and is, at least, worth exploring.

2 Differentiable functions vs programs

If we give up the constraint that the function we try to learn is differentiable, because we allow ourselves to use gradual descent instead of gradient descent, what kind of representation space can we use to describe this function? Well, the simplest answer to this is to move one step up in terms of generality and consider programs. They can be as simple as binary decision trees, or as complex as some elaborated python-like code or some DSL (Domain Specific Language) adapted for AI.

Importantly, programs have the following properties:

- they are discrete in structure and therefore can easily be subject to mutations, powering a local non-gradient based exploration of solutions
- they are compositional, allowing not only mutations but crossover
- they are hierarchical in nature, allowing the building blocks from which programs are made to be programs themselves, enriching the expressive power of the search when trying to find new programs

Navigating the space of programs at random would not be efficient. Here is another case for hybridization: using deep learning methods, for example graph neural networks or LLMs, to help guessing a starting point, and also to provide an oracle when probing around a given existing solution while performing the descent (instead of mutating the programs at random). We end up here somewhere in the middle of the gradient-random axis that we talked about above, with a not completely random but also not completely deterministic probing for a step in the descent. Note that a lot of very promising research on graph neural network already provide results on the oracle part mentioned above [Yang et al., 2022, Allamanis et al., 2017], and also more recently some extremely interesting developments using LLMs to generate motor control programs [Ma et al., 2023].

Another example of what can be done in terms of hybridization is how we could use a genetic algorithm to evolve a formal grammar for a particular language, using a large language model like GPT-4 as an oracle to compute a fitness by comparing the kind of sentences that the current grammar candidates are producing and assessing how statistically realistic they sound. This is a nice coupling of statistical evaluation (with all its approximations, but for a fitness it is acceptable) and formal structure evolution, which comes with many computational advantages once the final grammar has been stabilized.

As a side note, it's interesting to see that the requirement of differentiable functions to process input data brings along the requirement to "flatten" data into vectors for input (or matrices, tensors, etc). This process loses the potentially recursive structure of the elements of the input space, which could be easily exploited by a program in a non-differentiable framework (think of graphs, hypergraphs, or even programs themselves as inputs). Of course, it is possible to recover some part of the structure in a neural network framework, in particular using transformers and attention, but it is essentially trying to recover something that is already a given in the natural initial form of the input data. Considerable efforts in terms of research time and computational time are devoted to work around the constraint of vectorization of compositional/recursive complex information, in order to recover what was already there to start with.

3 Human intelligence emergence is culturally driven

Let us now consider the second main point, about how new information is effectively acquired and fed into the system. Learning typically involves feeding the system with new information. It turns out that the particular way information is presented plays a central role here. Not just in terms of how fast it can converge, but, for all practical purposes (assuming finite time), in terms of being able to converge at all.

In most machine learning instances, information is fed to the system in batches. This is true in supervised learning, but also in unsupervised learning, where large datasets of images or videos are assembled to train the system. This leads to the establishment of benchmarks against which competing models are compared. The community is well aware of the fact that the order of presentation of learning examples is important (this is called "curriculum learning" [Bengio et al., 2009]), and that a careful selection of a good strategy to order the examples can lead to much better convergence speed, final performance or generalization.

What is much less often integrated is the importance of interactive learning, where the system not only absorbs data that is fed to it, but also actively participate in the building of learning situations from which new data can be acquired. This is called "active learning" [Settles, 2009]. Fundamentally, it allows to explores questions related to causality (where, ultimately, an active "acting" part is needed in order to go beyond correlation extraction), intrinsic motivation, and embodiment since the active part requires some form of actionable "motor" dimension in the system. These points are not really ignored by the deep learning community, at least not by people at the forefront of research, but the dataset approach remains dominant (in part also because it allows for well defined measures of progress and comparison of results, which is a priori desirable but comes also with issues in terms of generalization capacity). More generally, the prevailing philosophy is that more data leads to better learning, and that raw data is essentially all you need. It relates to the inductivist view that knowledge is somehow contained in data, and that we can extract it out by appropriate automated/deterministic means.

But the interactive nature of learning is not the only issue and again, some researchers in machine learning are acknowledging this and working on it. We would argue that an isolated system, equipped with active learning and a body in a rich environment, would still not be able to reach significant levels of what we could recognize as human-level intelligence. Or perhaps, it would be able to reach a certain level of animal-level intelligence (which would arguably be a great achievement). It is often argued by the deep learning community that animal-level intelligence is a crucial step towards human-level intelligence, because the genome of animals being in fact quite similar to the one of humans (98% of common genes with apes, and more than 90% with most other mammals), so there cannot be a big step forward to human-level intelligence once animal-level intelligence has been reached. Therefore, we should focus on animal-level intelligence, and the rest would follow quickly. We believe this point of view is wrong. It is a genome-centric view that basically says that the developmental and cognitive trajectory of the organism is mostly explained by genetic factors. What is ignored in this reasoning is the large influence of another evolutionary body: human culture. The correct picture most likely involves a complex and entangled co-evolution of genes and culture in the case of homo-sapiens, and the 2% genome difference with apes might not code for an extra intelligence capacity, but simply for the extra bit of complexity that unlocks cultural dynamics, the "cultural hook" (more neurons to handle things like theory of mind, building joint plans with congenerates [Tomasello, 1999], etc), with otherwise a very similar general intelligence capacity. In fact, it would probably be possible to build an AI with a very limited animal-level intelligence, but once the "cultural hook" is added, it would be boosted to something close to human-level very quickly. In other words: to get to human-level intelligence, an agent need to be actively learning while being immersed in a physical and cultural environment which is strongly influencing the developmental trajectory.

The different "ways" of learning that we talked about are not only related to curriculum learning (the order of presentation of examples), but to a vast repertoire of "learning games" that are ritualized learning interactions and strategies practiced by adults with infants, and then children and young adults, from kid's play up to the more advanced ritualization that is modern schooling systems. In short: animal-level intelligence + cultural hook + cultural environment = human-level intelligence. And the animal-level part might not be the most crucial part in this equation. The cultural environment is important and complement the genetic environment, as it defines good strategies on "how" to learn, vs the current machine learning paradigm that takes a simplified view based on the non-interactive, non-social strategy of showing training examples from a dataset (which humans actually never do).

To summarize, a proper learning strategy that has a chance to catch up with the complexity of all that is to be learned for human-level intelligence probably needs to build on culturally grounded and socially experienced learning games, or strategies.

4 Developmental AI

The above picture about learning fits particularly well with what is called the developmental approach in AI (also in robotics [Cangelosi and Schlesinger, 2013]), taking inspiration from developmental psychology in order to understand how children are learning, and in particular how language is grounded in the first years.

Key elements of the developmental AI program include, in order of complexity:

- embodiment and grounding symbolic abstractions into sensorimotor invariants
- interaction with a physical world (naive physics) but also with a social world (theory of mind)
- language evolution as a scaffold for socially grounded intelligence
- building human-specific social constructs like joint plans, joint attention, narratives (see for example [Tomasello, 2019] for a detailed analysis of human-specific social capacities)

Language in particular is approached here in a very different way compared to the today's dominant NLP paradigm: instead of superficially treating words and grammatical structures as statistically related token, it sees words and grammar as the result of an evolutionary process trying to maximize a communication fitness (see also [Steels and Hild, 2012, Steels, 2015, 2005a]), aiming at the execution of socially and environmentally grounded joint tasks, requiring cooperation and agreement on grounded meanings. The grounding is such that words refer to perceptual and conceptual categories that have proved useful to efficiently communicate, reach goals together and reduce ambiguity (words or grammatical structures that fail to meet these needs are "selected away" from the emergent language, via an evolutionary dynamics). This creates a grounded basis for language, immediately connected to physical and social reality, to goals and situations, to particular ways of categorizing the world such that the resulting categories are leading to successful interactions. Capturing this complexity in a machine would allow it to build grounded meaning (as a link between internal sensorimotor or conceptual categories and external word/grammatical token) and help resolve common sense reasoning traps in which the current NLP statistical machines fall and will continue to fall as long as no proper semantics is integrated in the learning process.

The semantic layer is not contained in the data, but in the process of acquiring this data, so the particular learning approach of current deep learning methods, focusing on benchmarks and batch processing, cannot capture this important dimension. The process of acquiring data, that we could call “learning games” is itself subject to evolutionary pressure and constitutes an asset that an intelligent human benefit from immediately after being born, when the first interactions with the parents start, and then further on when subject to learning via the ritualized methods invented by his/her community. This crucial aspect of learning has to be integrated into the design of intelligent machines if we hope to reach human-level intelligence, or strong AI.

Some of the key scientific questions that are more naturally addressable by developmental AI approach include:

- Grounding causal Bayesian networks into interactions, making use of the “do” operator in the machinery of probabilistic conditional reasoning [Pearl, 2018, Gasse et al., 2021]
- Grounding language via structured interactions with an external agent [Steels and Baillie, 2003, Baillie et al., 2005]
- Formally advance the definition of a framework for joint attention (e.g Kaplan and Hafner [2006]), as described by developmental psychologists [Tomasello et al., 2005], but still not covered by current AI/robotics.
- Intrinsic motivation in the framework of reinforcement learning [Colas et al., 2020, Schmidhuber, 2010, Oudeyer and Kaplan, 2007, Oudeyer et al., 2005]
- Theory of mind [Dunbar, 1998, Baker et al., 2017, Leslie et al., 2004]

Most of these questions are recognized today as crucial structural questions to make progress in AI by the community [Goyal and Bengio, 2020, Chevalier-Boisvert et al., 2018, Colas et al., 2020, Bengio et al., 2019], with very ambitious and promising experiments already taking place in virtual environments [Abramson et al., 2020, Ma et al., 2023, Fan et al., 2022].

To summarize, developmental AI stresses the importance of learning in the context of a self-supervised, open-ended interaction with the environment, as opposed to using static batches on large datasets. This allows the system to interactively perform actions and probe causal relationships. Crucially, interactions can then extend to the social domain, involving elaborate interactive games performed with an external agent, in a way similar to what babies experience in the first years of life, as it has been extensively studied in the field of developmental psychology [Vygotsky, 1994, Tomasello, 1999, Smith and Gasser, 2005, Tomasello et al., 2005, Carpenter et al., 1998, Sheridan et al., 2007]. Such an approach recognizes the importance for an intelligent system to learn by building and scaffolding gradually more and more complex grounded representations of its physical and social environment, up to the stage of evolving language and complex structured interactions with others. Meaning emerges as sensorimotor, physical and social invariants are dynamically formed and grounded into abstractions. This necessarily calls for an integrated approach where several learning time-scales and abstraction levels are mixed, instead of being separated in carefully prepared experiments, and where long experiments are run uninterrupted for long periods of time, also known as ”life-long learning” [Ruvolo and Eaton, 2013, Silver et al., 2013].

In this context, and due to the strong inspiration from developmental psychology, the metaphor for the target system shifts from trying to build a fully-formed competent adult-like intelligence built by processing large amount of data, to trying to emulate the cognitive development of a child [Chevalier-Boisvert et al., 2018, Guerin, 2011], getting back to the root inspiration hinted at by Turing himself in his famous 1950 seminal paper [Turing, 1950].

5 From robotics to simulation in VR

At a more concrete level, realizing the above program for developmental AI involves building child-like machines that are immersed in a rich cultural environment, involving humans, where they will be able to participate in learning games. Learning games involving only the physical world can easily be run in simulation, with accelerated time, and this is already done to some extent by the AI community nowadays. For the social dimension, involving interactions with humans, virtual reality headset are a very promising interface allowing natural interactions that include face expression (some VR headset support this already), gaze following, gesture and body posture, and many other non-verbal communication signals that are crucial to bootstrap the learning of language.

Without the help of VR, the only option that was possible was to actually build robots that can interact with humans. This was the developmental robotics program. One problem with robotics is that it is not possible to avoid facing all the problems of embodiment and sensorimotor grounding. While those problems are interesting, they also constitute a very severe limitation: it makes it impossible to study higher level cognitive questions like language evolution, social interactions or causality induction, without first solving low level issues like grasping, motor control, walking, etc. Adding these issues to the fact that robots are expensive, require maintenance and lots of engineering efforts, this creates a high barrier to entry.

The very interesting aspect of the VR approach is that it allows to shortcut these issues if needed (for example, if we have good reasons to believe that the building up of the low level is not somehow crucial to scaffold the high level). One can provide a “grasping function” that will simply perform inverse kinematics with a ”magic” grasp and let us focus on the social/theory of mind aspects of a particular learning game. We could go as far as providing a scene graph of existing and visible objects, assuming that identifying and locating objects could potentially be done via deep networks further down the architecture (with potential top-down influence). The point is here to focus on the study of the cultural interaction and how the ”cultural hook” works, not on the animal-level intelligence which is, in this developmental approach, not necessarily the most important part to get to human-level intelligence.

The possibility to significantly speed up time is also an advantage, at least in the phase of development when the agent is simply interacting with the environment with no social contact.

6 Research program

6.1 Hybrid approach and developmental program

The experimental side of our work will focus on running a virtual avatar "agent" in a simplified but realistic 3D world, which can interact with a user equipped with a VR headset and controllers. This is in line with the objective of transposing the developmental robotics framework within a simulated virtual environment, while allowing rich social interactions with a human caretaker. The VR aspect here is key to provide a natural interaction medium for the human with the agent, as if the interaction would be happening in the real world with a robot.

The core inspiration here is the notion of language games [Steels and Hild, 2012], but instead of having a static structure for the game, we propose to have this structure evolved and negotiated by the agents, alongside the language itself.

The nature of the interaction, which is open-ended, can be framed within the general context of reinforcement learning, or more generally multi-agent reinforcement learning. Several aspects however are specific here:

1. The agent would be storing its policy as a program (see inspiration from [Tsividis et al., 2021, Lake et al., 2015]). The environment is modeled in part as a predictive model that can be implemented as a deep network [Arulkumaran et al., 2017], and in part also as a program that is in charge of formally representing the rules of a "learning game" that the agent intends to play with the user. This "environment" program needs to be synchronized with the user, via language, in order to attempt to play the same game.

2. The agent also stores a program representing what it believes is the policy of the user, enabling the development of a form of theory of mind [Baker et al., 2017]. In a more general sense, the user will be considered as an object alongside other inanimate objects in the scene (e.g. [Forestier and Oudeyer, 2017])

3. Policy programs instructions include actions such as giving a reward to the other, making the reward of the RL framework coming not from an external predefined source, but from the other player behavior, introducing interesting coupled dynamics.

From a technical point of view, this work will potentially leverage techniques from inverse reinforcement learning [Fu et al., 2017], meta-learning [Botvinick et al., 2019] and multi-agent reinforcement learning [Bullard et al., 2020, Bloembergen et al., 2015, Busoniu et al., 2008], and program induction [Valkov et al., 2018, Gaunt et al., 2016, Lake et al., 2015], and more generally hybrid approaches to AI, combining inspiration from symbolic formal processing and neural networks.

6.2 Coupling deep learning and evolutionary dynamics

As discussed above, evolutionary dynamics appears as a natural way to explore non differentiable problem spaces [Katoch et al., 2020]. A particularly interesting approach is the idea of representing learned functions as discrete stochastic programs (in contrast to differentiable programs) [Dechter et al., 2013, Ellis et al., 2015, Parisotto et al., 2016]. These programs can undergo a form of selectionist evolution (instead of a stochastic gradient descent) to match a fitness. In this context, the fitness could be provided by a deep network, acting as an oracle, and trained on experienced raw statistical data. While the statistical regularity of the environment is captured by the deep network, the formal and causal nature of its structure gets represented in a discrete structure that is adjusted to match the statistical signature using fast forward evolution that can take place at high speed within the system. This contrasts with some neurosymbolic approaches which attempts to extract symbolic information directly from the network parameters [Lehmann et al., 2010], or to mirror a symbolic structure onto a network architecture [Valkov et al., 2018, Manhaeve et al., 2018].

Not only could deep networks be used to implement a form of fitness function for evolutionnary dynamics, but it could also potentially provide a better-than-random way of handling mutation during the search, capturing heuristics on efficient strategies based on past successes. To our knowledge, this is still open research.

6.3 Artificial curiosity to drive language evolution

In the context of the above experiment, we will study how we can characterize interesting games as games which maximize the rate of reward progress, defined as artificial curiosity [Oudeyer and Kaplan, 2007] (a form of intrinsic reward), but also games that can be successfully synchronized via languages. Synchronized games have the property that both participants are running similar game definitions that can work well in sync as to produce progress on the rate of reward progress.

Achieving synchronization involves the task of grounding proper language conventions to be able to serialize the structure of games into words, and converge on a common set of vocabulary/grammar, characterized by their efficiency at communicating and helping build synchronized, successful games. We will take inspiration here from computational models of evolutionnary approaches to language evolution [Steels and Hild, 2012, Steels, 2005b, Spranger, 2011, Vogt, 2006].

The goal is to create an open-ended setup where the agent and the user can collaborate on developing a simple language capable of expressing information about the environment, and also about the structure of potential interaction games. The agent and the user can then engage in language-synchronized games, whose purpose can be to improve any aspect of the agent internal predictive models: model of the environment, language model, model of the user (theory of mind). Recent work on the emergence of communication protocols in the context of the architect-builder problem [Barde et al., 2021].

We believe this experimental setup can serve as a groundwork to implement a scalable developmental path for the AI agent, in interaction with humans. It contains all the key elements needed to approach issues such as joint attention, theory of mind, and grounded language evolution.

7 Application

To focus the work on a concrete objective, we will aim at producing a working version of this interaction game in VR, as a released application for the Meta Quest platform and VR devices by Apple.

The informal goal that will be presented to the player is to "educate" its avatar. Internal states of the avatars will be stored on a central server, or on a decentralized file system like IPFS, to allow multiple users to participate in the interactions. Interesting distributed experiments will be possible based on the community of players.

This application is in line with a tradition of online virtual "pets", that we propose to revive through the lens of a serious scientific program, aiming at producing grounded and meaningful interactions with the virtual avatar.

The above research program would be explored over a period of five to ten years, with a small team of researchers and research engineers dedicated to the realization of the VR application/experiment described above. A first iteration of a demo app could be produced within the first years, gradually improving towards the goal of a fully grounded developmental AI implementation.

References

Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press Massachusetts, USA, 2017.

Melanie Mitchell. Why ai is harder than we think. *Proceedings of the Genetic and Evolutionary Computation Conference*, 2021.

Gary Marcus. The next decade in ai: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*, 2020.

Murray Shanahan and Melanie Mitchell. Abstraction for deep reinforcement learning. *arXiv preprint arXiv:2202.05839*, 2022.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Josh Abramson, Arun Ahuja, Iain Barr, Arthur Brussee, Federico Carnevale, Mary Cassin, Rachita Chhaparia, Stephen Clark, Bogdan Damoc, Andrew Dudzik, et al. Imitating interactive intelligence. *arXiv preprint arXiv:2012.05672*, 2020.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. *arXiv preprint arXiv:2110.07058*, 2021.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. *arXiv preprint arXiv:1810.08272*, 2018.

Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *arXiv preprint arXiv:2011.15091*, 2020.

Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*, 2016.

Sebastian Bader and Pascal Hitzler. Dimensions of neural-symbolic integration-a structured survey. *arXiv preprint cs/0511042*, 2005.

Kang Yang, Huiqun Yu, Guisheng Fan, Xingguang Yang, and Zijie Huang. A graph sequence neural architecture for code completion with semantic structure features. *Journal of Software: Evolution and Process*, 34(1):e2414, 2022.

Miltiadis Allamanis, Marc Brockschmidt, and Mahmoud Khademi. Learning to represent programs with graphs. *arXiv preprint arXiv:1711.00740*, 2017.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models, 2023.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Burr Settles. Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences*, 2009.

M. Tomasello. *The cultural origins of human cognition*. Harvard University Press, 1999.

Cangelosi and Schlesinger. *Developmental Robotics*. Cangelosi and Schlesinger, 2013.

Michael Tomasello. *Becoming human*. Harvard University Press, 2019.

Luc Steels and Manfred. Hild. *Language grounding in robots*. Springer, 2012.

Luc Steels. The talking heads experiment. *Computational Models of Language Evolution*, 2015.

Luc Steels. The role of construction grammar in fluid language grounding. *Unpublished manuscript*, 2005a.

Judea Pearl. Causal and counterfactual inference. *The Handbook of Rationality*, pages 1–41, 2018.

Maxime Gasse, Damien Grasset, Guillaume Gaudron, and Pierre-Yves Oudeyer. Causal reinforcement learning using observational and interventional data. *arXiv preprint arXiv:2106.14421*, 2021.

Luc Steels and Jean-Christophe Baillie. Shared grounding of event descriptions by autonomous robots. *Robotics and autonomous systems*, 43(2-3):163–173, 2003.

Jean-Christophe Baillie, Matthieu Nottale, B Victor, and P France. Talking robots: a fully autonomous implementation of the talking heads. In *IJCAI*, pages 1725–1726, 2005.

Frederic Kaplan and Verena V Hafner. The challenges of joint attention. *Interaction Studies*, 7(2):135–169, 2006.

M. Tomasello, M. Carpenter, J. Call, T. Behne, H. Moll, et al. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and brain sciences*, 28(5):675–690, 2005.

Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Intrinsically motivated goal-conditioned reinforcement learning: a short survey. *arXiv preprint arXiv:2012.09830*, 2020.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in Neurorobotics*, 1, 2007.

P.Y. Oudeyer, F. Kaplan, V.V. Hafner, and A. Whyte. The playground experiment: Task-independent development of a curious robot. In *Proceedings of the AAAI Spring Symposium on Developmental Robotics*, pages 42–47. Stanford, California, 2005.

R. Dunbar. *Approaches to the evolution of language: Social and cognitive bases*. Cambridge University Press, 1998.

Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017.

Alan M Leslie, Ori Friedman, and Tim P German. Core mechanisms in 'theory of mind'. *Trends in cognitive sciences*, 8(12):528–533, 2004.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge, 2022.

Lev Vygotsky. Tool and symbol in child development. *The vygotsky reader*, 1994.

Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artificial Life*, 11(1-2):13–29, 2005.

Malinda Carpenter, Katherine Nagell, Michael Tomasello, George Butterworth, and Chris Moore. Social cognition, joint attention, and communicative competence from 9 to 15 months of age. *Monographs of the society for research in child development*, 1998.

Mary D Sheridan, Ajay Sharma, and Helen Cockerill. *From birth to five years: children's developmental progress*. Routledge, 2007.

Paul Ruvolo and Eric Eaton. Ella: An efficient lifelong learning algorithm. In *International Conference on Machine Learning*, pages 507–515. PMLR, 2013.

Daniel L Silver, Qiang Yang, and Lianghao Li. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*, 2013.

Frank Guerin. Learning like a baby: a survey of artificial intelligence approaches. *The Knowledge Engineering Review*, 26(02):209–236, 2011.

Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

Pedro A Tsividis, Joao Loula, Jake Burga, Nathan Foss, Andres Campero, Thomas Pouncy, Samuel J Gershman, and Joshua B Tenenbaum. Human-level reinforcement learning through theory-based modeling, exploration, and planning. *arXiv preprint arXiv:2107.12544*, 2021.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6): 26–38, 2017.

Sébastien Forestier and Pierre-Yves Oudeyer. A unified model of speech and tool use early development. In *39th Annual Conference of the Cognitive Science Society (CogSci 2017)*, 2017.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Matthew Botvinick, Sam Ritter, Jane X Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow. *Trends in cognitive sciences*, 23 (5):408–422, 2019.

Kalesha Bullard, Franziska Meier, Douwe Kiela, Joelle Pineau, and Jakob Foerster. Exploring zero-shot emergent communication in embodied multi-agent populations. *arXiv preprint arXiv:2010.15896*, 2020.

Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research*, 53:659–697, 2015.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Lazar Valkov, Dipak Chaudhari, Akash Srivastava, Charles Sutton, and Swarat Chaudhuri. Houdini: Lifelong learning as program synthesis. *arXiv preprint arXiv:1804.00218*, 2018.

Alexander L Gaunt, Marc Brockschmidt, Rishabh Singh, Nate Kushman, Pushmeet Kohli, Jonathan Taylor, and Daniel Tarlow. Terpret: A probabilistic programming language for program induction. *arXiv preprint arXiv:1608.04428*, 2016.

Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, pages 1–36, 2020.

Eyal Dechter, Jon Malmaud, Ryan P Adams, and Joshua B Tenenbaum. Bootstrap learning via modular concept discovery. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Kevin Ellis, Eyal Dechter, and Joshua B Tenenbaum. Dimensionality reduction via program induction. In *2015 AAAI Spring Symposium Series*, 2015.

Emilio Parisotto, Abdel-rahman Mohamed, Rishabh Singh, Lihong Li, Dengyong Zhou, and Pushmeet Kohli. Neuro-symbolic program synthesis. *arXiv preprint arXiv:1611.01855*, 2016.

Jens Lehmann, Sebastian Bader, and Pascal Hitzler. Extracting reduced logic programs from artificial neural networks. *Applied intelligence*, 32(3):249–266, 2010.

Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. *arXiv preprint arXiv:1805.10872*, 2018.

Luc Steels. The emergence and evolution of linguistic structure: from lexical to grammatical communication systems. *Connection science*, 17(3-4):213–230, 2005b.

M. Spranger. The evolution of grounded spatial language. *Unpublished doctoral dissertation, Vrije Universiteit Brussels (VUB), Brussels, Belgium*, 2011.

P. Vogt. Language evolution and robotics: Issues on symbol grounding. *Artificial cognition systems*, page 176, 2006.

Paul Barde, Tristan Karch, Derek Nowrouzezahrai, Clément Moulin-Frier, Christopher Joseph Pal, and Pierre-Yves Oudeyer. Learning to guide and to be guided in the architect-builder problem. *ArXiv*, abs/2112.07342, 2021.